

MGDGRID - Managing the Grid

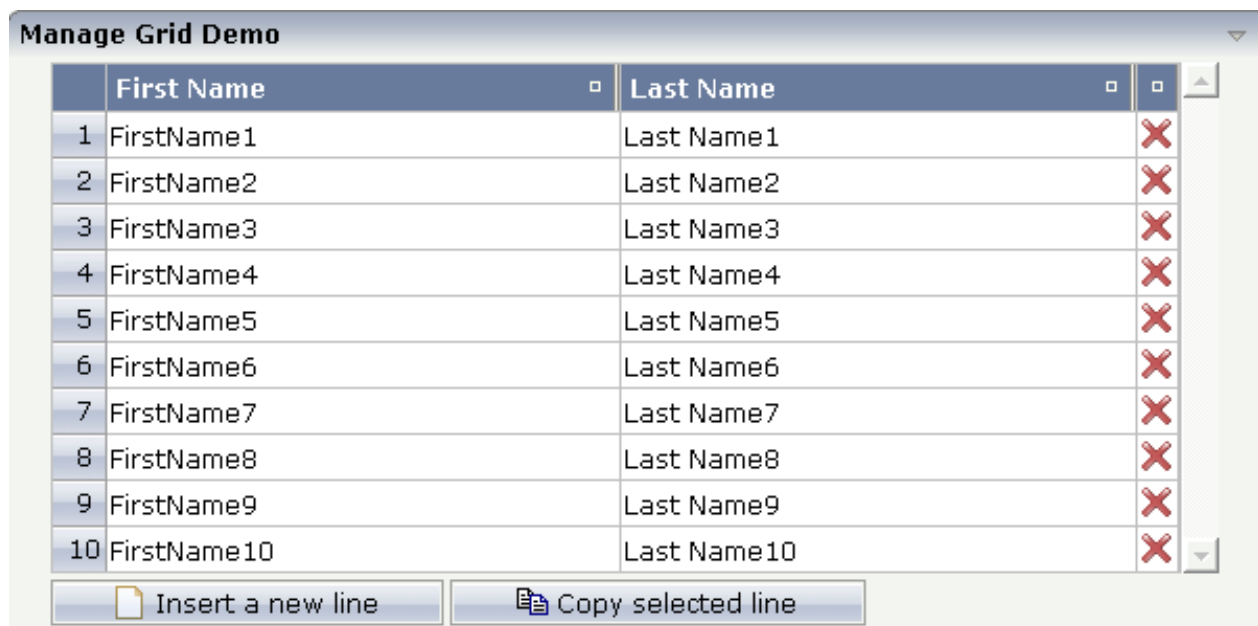
The MGDGRID control is an extension of the ROWTABLEAREA2 control. It allows to insert, copy and delete rows of the grid without the need of any corresponding Java coding.

This chapter covers the following topics:

- Example
- MGDGRID Properties
- ROWINSERT Properties
- ROWCOPY Properties
- ROWDELETE Properties

See also *STR Properties* which are described with the ROWTABLEAREA2 control.

Example



There is a grid that contains a header row and 10 lines. Each line contains two fields and a "delete row" control.

Each of the function controls (insert, copy, delete) can be added at the top of the MGDGRID, below the MGDGRID or within the lines of the MGDGRID.

Look at the corresponding layout definition:

```

<rowarea name="Manage Grid Demo">
  <mgdgrid griddataprop="mglines" rowcount="10" width="100%" firstrowcolwidths="true">
    <tr>
      <label name=" " width="25" asheadline="true">
      </label>
      <gridcolheader name="First Name" width="50%">
      </gridcolheader>
      <gridcolheader name="Last Name" width="50%" >
      </gridcolheader>
      <gridcolheader width="20">
      </gridcolheader>
      <hdist></hdist>
    </tr>
    <repeat>
      <str valueprop="selected" showifempty="true">
        <selector valueprop="selected" singleselect="true">
        </selector>
        <field valueprop="fname" width="100%">
        </field>
        <field valueprop="lname" width="100%">
        </field>
        <rowdelete>
        </rowdelete>
      </str>
    </repeat>
    <mgdfunctions>
      <rowinsert title="Insert a new line">
      </rowinsert>
      <rowcopy title="Copy selected line">
      </rowcopy>
    </mgdfunctions>
  </mgdgrid>
</rowarea>

```

This is the corresponding adapter code:

```

public class ManageGridDemoAdapter
  extends Adapter
{
  // property >mglines<
  MGDGRIDCollection m_mglines = new MGDGRIDCollection(MGDItem.class);
  public GRIDCollection getMglines() { return m_mglines; }
}

```

The constructor of the MGDGRIDCollection class needs an "item class". This "item class" cannot be a sub/inner class this time; therefore two separate Java files are required for the MGDGRID.

This is the Java code of the corresponding MGDItem class:

```

public class MGDItem
{
  // property >fname<
  String m_fname;
  public String getFname() { return m_fname; }
  public void setFname(String value) { m_fname = value; }

  // property >lname<
  String m_lname;
  public String getLname() { return m_lname; }
  public void setLname(String value) { m_lname = value; }
}

```

```

// property >selected<
boolean m_selected;
public boolean getSelected() { return m_selected; }
public void setSelected(boolean value) { m_selected = value; }
}
    
```

With the MGDGRID control, there is no need of further Java coding. It handles insert, copy and delete events itself, but the developer has to take care of the corresponding "item class".

The MGDGRID control is an extension to the ROWTABLEAREA2 control. See the description of the ROWTABLEAREA2 control for further information.

MGDGRID Properties

Basic			
griddataprop	<p>Name of adapter property representing the grid on server side.</p> <p>Must be of type "GRIDCollection". The whole grid is represented by the GRIDCollection-object, each individual row of the grid is represented by one item inside the collection.</p> <p>If using the control for building trees (TREENODE-control inside the grid's items) then use "TREECollection" on server side.</p>	Obligatory	
rowcount	<p>Number of rows that is rendered inside the control.</p> <p>There are two ways of using this property - dependent on whether you in addition define the HEIGHT property:</p> <p>If you do NOT define the HEIGHT property then the control is rendered with exactly the number of rows that is defined as ROWCOUNT value.</p> <p>If a HEIGHT value is defined in addition (e.g. as percentage value "100%") then the number of rows depends on the actual height of the control. The ROWCOUNT value in this case indicates the maximum number of rows that is picked from the server. You should define this value in a way that it is not too low - otherwise your grid will not be fully filled. On the other hand it should not be defined too high ("100") because this causes more communication traffic and more rendering effort inside the browser.</p>	Optional	

<p>height</p>	<p>Height of the control.</p> <p>There are three possibilities to define the height:</p> <p>(A) You do not define a height at all. As consequence the control will be rendered with its default height. If the control is a container control (containing) other controls then the height of the control will follow the height of its content.</p> <p>(B) Pixel sizing: just input a number value (e.g. "20").</p> <p>(C) Percentage sizing: input a percentage value (e.g. "50%"). Pay attention: percentage sizing will only bring up correct results if the parent element of the control properly defines a height this control can reference. If you specify this control to have a height of 50% then the parent element (e.g. an ITR-row) may itself define a height of "100%". If the parent element does not specify a width then the rendering result may not represent what you expect.</p>	<p>Optional</p>	<p>100</p> <p>150</p> <p>200</p> <p>250</p> <p>300</p> <p>250</p> <p>400</p> <p>50%</p> <p>100%</p>
<p>width</p>	<p>Width of the control.</p> <p>There are three possibilities to define the width:</p> <p>(A) You do not define a width at all. In this case the width of the control will either be a default width or - in case of container controls - it will follow the width that is occupied by its content.</p> <p>(B) Pixel sizing: just input a number value (e.g. "100").</p> <p>(C) Percentage sizing: input a percentage value (e.g. "50%"). Pay attention: percentage sizing will only bring up correct results if the parent element of the control properly defines a width this control can reference. If you specify this control to have a width of 50% then the parent element (e.g. an ITR-row) may itself define a width of "100%". If the parent element does not specify a width then the rendering result may not represent what you expect.</p>	<p>Sometimes obligatory</p>	<p>100</p> <p>120</p> <p>140</p> <p>160</p> <p>180</p> <p>200</p> <p>50%</p> <p>100%</p>

firstrowcolwidths	<p>If set to "true" then the grid is sized according to its first row. This first row typically is a header-TR-row in which GRIDCOLHEADER controls are used as column headers for the subsequent rows.</p> <p>Default is "false", i.e. the grid is sized according to its "whole content".</p> <p>Please note: when using the GRIDCOLHEADER control within the header-TR-row this property must be set to "true" - otherwise column resizing (by drag and drop) does not work correctly.</p>	Sometimes obligatory	true false
comment	<p>Comment without any effect on rendering and behaviour. The comment is shown in the layout editor's tree view.</p>	Optional	
Appearance			
withborder	<p>If set to "false" then no thin border is drawn around the controls that are contained in the grid.</p> <p>Default is "true".</p>	Optional	true false
hscroll	<p>Definition of the horizontal scrollbar's appearance.</p> <p>You can define that the scrollbars only are shown if the content is exceeding the control's area ("auto"). Or scrollbars can be shown always ("scroll"). Or scrollbars are never shown - and the content is cut ("hidden").</p> <p>Default is "auto".</p>	Optional	auto scroll hidden
vscroll	<p>Definition of the vertical scrollbar's appearance.</p> <p>You can define that scrollbars only are shown if the content is exceeding the control's area ("auto"). Or scrollbars can be shown always ("scroll"). Or scrollbars are never shown - and the content is cut ("hidden").</p> <p>Default is "auto".</p>	Optional	auto scroll hidden
firstrowcolwidths	(already explained above)		
clipboardaccess	<p>If switched to true then the content of the grid can be selected and exported into the client's clipboard.</p>	Optional	true false

withblockscrolling	If switched to "true" then the grid will show small scroll icons by which the user can scroll the grid's content. Scrolling typically is done by using the grid's scrollbar - the scroll icons that are switched on by this property are an additional possibility to scroll.	Optional	true false
touchpadinput	If set to "true" then touch screen icons for scrolling are displayed in addition. Default is "false".	Optional	true false
requiredheight	Minimum height of the control in pixels. Use this property to ensure a minimum height if the overall control's height is a percentage of the available space - i.e. if value of property HEIGHT is a percentage (e.g. 100%). Please note: You must not use FIXLAYOUT at the surrounding row container (ITR and ROWAREA). Otherwise: if the available space is less than the required height the end of the control is just cut off.	Optional	1 2 3 int-value
tablestyle	CSS style definition that is directly passed into this control. With the style you can individually influence the rendering of the control. You can specify any style sheet expressions. Examples are: border: 1px solid #FF0000 background-color: #808080 You can combine expressions by appending and separating them with a semicolon. Sometimes it is useful to have a look into the generated HTML code in order to know where direct style definitions are applied. Press right mouse-button in your browser and select the "View source" or "View frame's source" function.	Optional	background-color: #FF0000 color: #0000FF font-weight: bold
Binding			
oncontextmenumethod	Name of adapter method that is called when the user presses right mouse button into the grid - but not on an existing row (then the row item object is responsible for handling the right mouse button) but on "empty area" of the grid.	Optional	

fwdtabkeymethod	Name of an adapter method that is called if the user presses the TAB key within the very last cell of the grid (last cell within the last line). Use property FWDTABKEYFILTER to associate this call with a grid column.	Optional	
fwdtabkeyfilter	By default the FWDTABKEYMETHOD is called if the user presses the TAB key within the veryfirst cell of the grid. Input the name of a cell's VALUEPROP to associate the method call with any other column.	Optional	
bwdtabkeymethod	Name of an adapter method that is called if the user presses SHIFT and TAB keys within the first cell of a grid line. Use property BWDTABKEYFILTER to associate this call with a cell of choice.	Optional	
bwdtabkeyfilter	By default the BWDTABKEYMETHOD is called if the user presses the SHIFT and TAB keys within the very first cell of the grid. Input the name of a cell's VALUEPROP to associate the method call with any other column.	Optional	
Hot Keys			
hotkeys	<p>Comma separated list of hot keys. A hotkey consists of a list of keys and a method name. Separate the keys by "-" and the method name again with a comma</p> <p>Example:</p> <p>ctrl-alt-65;onCtrlAltA;13;onEnter ...defines two hot keys. Method onCtrlAltA is invoked if the user presses Ctrl-Alt-A. Method "onEnter" is called if the user presses the ENTER key.</p> <p>Use the popup help within the Layout Painter to input hot keys.</p>	Optional	

ROWINSERT Properties

Basic			
image	<p>URL that points to the image that is shown as icon.</p> <p>The URL either is an absolute URL or a relative URL. If using a relative URL then be aware of that the generated page is located directly inside your project's directory.</p> <p>Example: "images/icon.gif" points to an icon in an images-folder that is parallel to the page itself. "../HTMLBasedGUI/images/new.gif" point to a URL that is located inside a different project.</p>	Obligatory	
comment	<p>Comment without any effect on rendering and behaviour. The comment is shown in the layout editor's tree view.</p>	Optional	
Binding			
visibleprop	<p>Name of an adapter property that provides the information if this control is displayed or not. As consequence you can control the visibility of the control dynamically.</p> <p>The server side property needs to be of type "boolean".</p>	Optional	
Online Help			
title	<p>Text that is shown as tooltip for the control.</p> <p>Either specify the text "hard" by using this TITLE property - or use the TITLETEXTID in order to define a language dependent literal.</p>	Optional	
titletextid	<p>Text ID that is passed to the multi lanaguage management - representing the tooltip text that is used for the control.</p>	Optional	

ROWCOPY Properties

Basic			
image	<p>URL that points to the image that is shown as icon.</p> <p>The URL either is an absolute URL or a relative URL. If using a relative URL then be aware of that the generated page is located directly inside your project's directory.</p> <p>Example: "images/icon.gif" points to an icon in an images-folder that is parallel to the page itself. "../HTMLBasedGUI/images/new.gif" point to a URL that is located inside a different project.</p>	Obligatory	
comment	<p>Comment without any effect on rendering and behaviour. The comment is shown in the layout editor's tree view.</p>	Optional	
Binding			
visibleprop	<p>Name of an adapter property that provides the information if this control is displayed or not. As consequence you can control the visibility of the control dynamically.</p> <p>The server side property needs to be of type "boolean".</p>	Optional	
Online Help			
title	<p>Text that is shown as tooltip for the control.</p> <p>Either specify the text "hard" by using this TITLE property - or use the TITLETEXTID in order to define a language dependent literal.</p>	Optional	
titletextid	<p>Text ID that is passed to the multi lanaguage management - representing the tooltip text that is used for the control.</p>	Optional	

ROWDELETE Properties

Basic			
image	<p>URL that points to the image that is shown as icon.</p> <p>The URL either is an absolute URL or a relative URL. If using a relative URL then be aware of that the generated page is located directly inside your project's directory.</p> <p>Example: "images/icon.gif" points to an icon in an images-folder that is parallel to the page itself. "../HTMLBasedGUI/images/new.gif" point to a URL that is located inside a different project.</p>	Obligatory	
comment	<p>Comment without any effect on rendering and behaviour. The comment is shown in the layout editor's tree view.</p>	Optional	
Binding			
visibleprop	<p>Name of an adapter property that provides the information if this control is displayed or not. As consequence you can control the visibility of the control dynamically.</p> <p>The server side property needs to be of type "boolean".</p>	Optional	
Online Help			
title	<p>Text that is shown as tooltip for the control.</p> <p>Either specify the text "hard" by using this TITLE property - or use the TITLETEXTID in order to define a language dependent literal.</p>	Optional	
titletextid	<p>Text ID that is passed to the multi lanaguage management - representing the tooltip text that is used for the control.</p>	Optional	