ROWTABLEAREA2 - The Flexible Control Grid

The ROWTABLEAREA2 is a container control that allows other controls to be arranged inside its grid management.

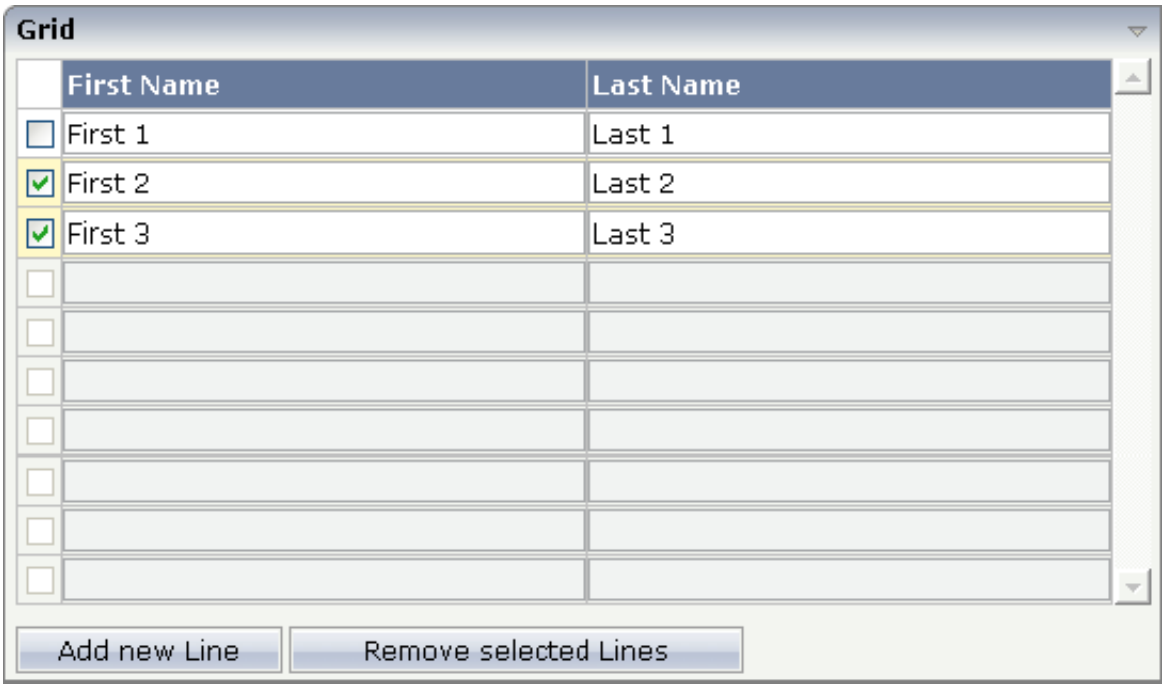
The ROWTABLEAREA2 control supports server-side scrolling and sorting. This concept is explained in *Server-Side Scrolling and Sorting*. An example for the usage of server-side scrolling and sorting with the ROWTABLEAREA2 control is contained in the example library `SYSEXNJX`.

This chapter covers the following topics:

- Example
- Adapter Interface
- Built-in Events
- Making Grids Look like Grids
- ROWTABLEAREA2 Properties
- STR Properties

Example

There is a grid that contains a header row and 10 lines. Each line contains one check box and two fields. Some of the lines are highlighted.



The XML layout definition is:

```
<rowarea name="Grid">
    <rowtablearea2 griddataprop="lines" rowcount="10" width="100%" withborder="true">
        <tr>
            <hdist>
            </hdist>
            <label name="First Name" asheadline="true">
            </label>
            <label name="Last Name" asheadline="true">
            </label>
        </tr>
        <repeat>
            <str valueprop="selected">
                <checkbox valueprop="selected" flush="screen" width="30">
                </checkbox>
                <field valueprop="firstname" width="50%">
                </field>
                <field valueprop="lastname" width="50%">
                </field>
            </str>
        </repeat>
    </rowtablearea2>
    <vdist height="10">
    </vdist>
    <itr>
        <button name="Add new Line" method="onAddLine">
        </button>
        <hdist>
        </hdist>
        <button name="Remove selected Lines" method="onRemoveLines">
        </button>
    </itr>
</rowarea>
```

Note the following:

- There is a ROWTABLEAREA2 definition with the property `griddataprop="lines"`. There is a `rowcount` definition of "10". This is the same as for the text grid processing: the grid container is bound to a server-side collection. Similar to the TEXTGRIDSSS2 definition, there is a row count that defines the number of lines.
- Inside the ROWTABLEAREA2 definition, there is first the definition of a normal table row (TR) in which a distance and two labels are defined. The labels are rendered with `asheadline="true"`.
- Inside the REPEAT definition, there is a special table row definition "STR" (selectable table row) that itself contains one CHECKBOX and two FIELD definitions. CHECKBOX and FIELDs are bound to properties themselves.
- After the ROWTABLEAREA2 definition, there is a vertical distance and a row that contains two buttons with which a user can manipulate the grid.

The content of the REPEAT block is repeated as many times as defined inside the `rowcount` definition of ROWTABLEAREA2. The content holds a table row (STR) - therefore the result is a grid.

Adapter Interface

In the parameter data area of the adapter, the grid data is represented by the following data structure:

```
DEFINE DATA PARAMETER
1 LINES (1:*)
2 FIRSTNAME (U) DYNAMIC
2 LASTNAME (U) DYNAMIC
2 SELECTED (L)
END-DEFINE
```

If the grid has been configured for server-side scrolling and sorting, the data structure contains additional fields that control server-side scrolling and sorting (see below). In order to use server-side scrolling and sorting, set the property `natsss` in NATPAGE to "true".

```
DEFINE DATA PARAMETER
1 LINES (1:*)
2 FIRSTNAME (U) DYNAMIC
2 LASTNAME (U) DYNAMIC
2 SELECTED (L)
1 LINESINFO
2 ROWCOUNT (I4)
2 SIZE (I4)
2 SORTPROPS (1:*)
3 ASCENDING (L)
3 PROPNAME (U) DYNAMIC
2 TOPINDEX (I4)
END-DEFINE
```

Built-in Events

value-of-griddataprop.onCtrlSelect
value-of-griddataprop.onSelect
value-of-griddataprop.onShiftSelect
value-of-griddataprop.onSort
value-of-griddataprop.onTopindexChanged

Making Grids Look like Grids

Fields typically contain a high number of FIELD controls. Typically, a FIELD control has a certain rendering that renders a field with a border and with a certain background color.

Be aware that inside the FIELD definition, there are two important properties:

- `noborder` - if set to "true", no border will be drawn
- `transparentbackground` - if set to "true", the field will always take over the background of the controls in which it is positioned (e.g. STR row).

Have a look at the difference between the following screens. One screen uses the properties, the other screen does not use them.

This is a grid:

	Article	Price
☐	Article 1	0.99
☐	Article 2	1.98
☐	Article 3	2.97
☐	Article 4	3.96
☐	Article 5	4.96
☐	Article 6	5.94
☐	Article 7	6.93
☐	Article 8	7.92
☐	Article 9	8.92
☐	Article 10	9.91

This is collection of fields:

	Article	Price
☐	Article 1	0.99
☐	Article 2	1.98
☐	Article 3	2.97
☐	Article 4	3.96
☐	Article 5	4.96
☐	Article 6	5.94
☐	Article 7	6.93
☐	Article 8	7.92
☐	Article 9	8.92
☐	Article 10	9.91

ROWTABLEAREA2 Properties

Basic			
griddataprop	Name of the adapter parameter that represents the control in the adapter.	Obligatory	

rowcount	Number of rows that is renderes inside the control. There are two ways of using this property - dependent on whether you in addition define the HEIGHT property: 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. If a HEIGHT value is defined an 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.	Optional	

height	Height of the control. There are three possibilities to define the height: (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. (B) Pixel sizing: just input a number value (e.g. "20"). (C) Percentage sizing: input a percantage 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.	Optional	100 150 200 250 300 250 400 50% 100%

width	Width of the control. There are three possibilities to define the width: (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. (B) Pixel sizing: just input a number value (e.g. "100"). (C) Percentage sizing: input a percantage 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.	Sometimes obligatory	100 120 140 160 180 200 50% 100%
firstrowcolwidths	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. Default is "false", i.e. the grid is sized according to its "whole content". 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.	Sometimes obligatory	true false

onloadbehaviour	Loading behaviour of the items into the client. "block" (=default) means that the client always requests the currently visible items from the server (=Server Side Scrolling). "collection" means that the client requests all items at the beginning from the server. The client itself implements the scrolling in the JavaScript/SWT.	Optional	block collection
comment	Comment without any effect on rendering and behaviour. The comment is shown in the layout editor’s tree view.	Optional	
Appearance			
withborder	If set to "false" then no thin border is drawn around the controls that are contained in the grid. Default is "true".	Optional	true false
hscroll	Definition of the horizontal scrollbar’s appearance. 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"). Default is "auto".	Optional	auto scroll hidden
vscroll	Definition of the vertical scrollbar’s appearance. 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"). Default is "auto".	Optional	auto scroll hidden
firstrowcolwidths	(already explained above)		

clipboardaccess	If switched to true then the content of the grid can be selected and exported into the client’s clipboard.	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
darkbackground	Normally the background is in light colour but the CIS style sheets also have a dark(er) grey colour to be used. If DARKBACKGROUND is set to true then the darker background colour is chosen. This property typically is used to integrate light coloured controls into darker container areas.	Optional	true false
invisiblemodeincompletelastrow	If set to "invisible" an incomplete last row is not shown.	Optional	invisible visible
withsliderfreeze	Setting this to "true" prevents unwisched slider jumps while scrolling up/down in a grid with a huge number of lines (for example 20000).	Optional	true false
Binding			
oncontextmenumethod	Name of the event that is sent to the adapter when the user presses the right mouse button in the grid, but not on an existing row, but in an empty area of the grid.	Optional	

fwdtabkeymethod	Name of the event that is sent to the adapter when 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 the event that is sent to the adapter when 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	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 Example: 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. Use the popup help within the Layout Painter to input hot keys.	Optional	
Natural			

njx:natname	If a Natural variable with a name not valid for Application Designer (for instance #FIELD1) shall be bound to the control, a different name (for instance HFIELD1) can be bound instead. If the original name (in this case #FIELD1) is then specified in this attribute, the original name is generated into the parameter data area of the Natural adapter and a mapping between the two names is generated into the PROCESS PAGE statement of the Natural adapter.	Optional	
njx:natcomment	The value of this attribute is generated as comment line into the parameter data area of the Natural adapter, before the field name. The Map Converter, for instance, uses this attributes to indicate for a generated statusprop variable to which field the statusprop belongs.	Optional	

STR Properties

STR (selectable table row) is a normal table row (TR) that highlights its background depending on an adapter property.

Basic			
valueprop	Name of the adapter parameter that defines if the row is selected or not.	Obligatory	
withalterbackground	Flag that indicates if the grid line shows alternating background color (like rows within a textgrids). Default is false. Please note: controls inside the row must have transparent background. In case of the FIELD control simply set property TRANSPARENTBACKGROUND to true.	Optional	true false
showifempty	Flag that indicates if an unused row is visible. Example: if set to false a grid with rowcount ten and a server side collection size of seven will hide the three remaining rows. Default is false.	Optional	true false
comment	Comment without any effect on rendering and behaviour. The comment is shown in the layout editor's tree view.	Optional	
Binding			
valueprop	(already explained above)		
onclickmethod	Name of the event that is sent to the adapter when the user clicks a line.	Optional	
ondblclickmethod	Name of the event that is sent to the adapter when the user double clicks a line.	Optional	
contextmenumethod	Name of the event that is sent to the adapter when the user presses the right mouse button in an empty area.	Optional	
proprefprop	Name of the adapter parameter that is filled when the user clicks a FIELD control. The VALUEPROP of the clicked field control will passed.	Optional	
backgroundcolorprop	Name of the adapter parameter that dynamically provides the background color for this control.	Optional	