

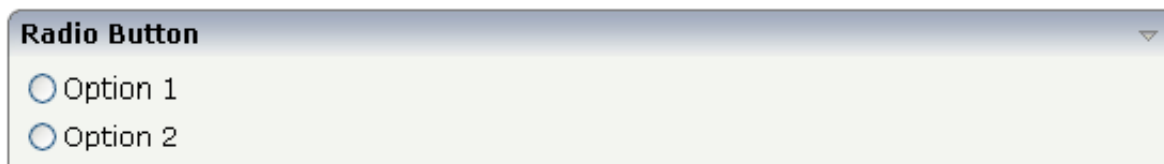
RADIOBUTTON

The RADIOBUTTON control displays the radio button. Radio buttons can be grouped together so that a group of RADIOBUTTON controls manipulates one property of an adapter control. Each RADIOBUTTON instance represents one value provided by the adapter property.

The following topics are covered below:

- Example
- Properties

Example



The XML layout definition is:

```
<rowarea name="Radio Button">
  <itr>
    <radiobutton valueprop="option" value="1">
    </radiobutton>
    <label name="Option 1" asplaintext="true">
    </label>
  </itr>
  <itr>
    <radiobutton valueprop="option" value="2">
    </radiobutton>
    <label name="Option 2" asplaintext="true">
    </label>
  </itr>
</rowarea>
```

The Java code of the adapter is:

```
// property >option<
int m_option;
public int getOption() { return m_option; }
public void setOption(int value) { m_option = value; }
```

In the code example, an integer value is used as a property. You can also use any other kind of data type: string, boolean, float, etc.

Properties

Basic			
valueprop	Server side property representation of the control.	Obligatory	

value	Value that represents this instance of the RADIOBUTTON control. The value is set into the adapter property that is defined by the VALUEPROP property when the user clicks onto the control. - Vice versa: the control is switched to "marked" when the adapter property holds the value defined.	Optional	
comment	Comment without any effect on rendering and behaviour. The comment is shown in the layout editor's tree view.	Optional	
Appearance			
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 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.	Optional	100 120 140 160 180 200 50% 100%
displayonly	If set to true, the FIELD will not be accessible for input. It is just used as an output field.	Optional	true false

align	<p>Horizontal alignment of control in its column.</p> <p>Each control is "packaged" into a column. The column itself is part of a row (e.g. ITR or TR). Sometimes the size of the column is bigger than the size of the control itself. In this case the "align" property specifies the position of the control inside the column. In most cases you do not require the align control to be explicitly defined because the size of the column around the controls exactly is sized in the same way as the contained control.</p> <p>If you want to directly control the alignment of text: in most text based controls there is an explicit property "textalign" in which you align the control's contained text.</p>	Optional	<p>left</p> <p>center</p> <p>right</p>
valign	<p>Vertical alignment of control in its column.</p> <p>Each control is "packaged" into a column. The column itself is part of a row (e.g. ITR or TR). Sometimes the size of the column is bigger than the size of the control. In this case the "align" property specify the position of the control inside the column.</p>	Optional	<p>top</p> <p>middle</p> <p>bottom</p>
colspan	<p>Column spanning of control.</p> <p>If you use TR table rows then you may sometimes want to control the number of columns your control occupies. By default it is "1" - but you may want to define the control to span over more than one columns.</p> <p>The property only makes sense in table rows that are synchronized within one container (i.e. TR, STR table rows). It does not make sense in ITR rows, because these rows are explicitly not synched.</p>	Optional	<p>1</p> <p>2</p> <p>3</p> <p>4</p> <p>5</p> <p>50</p> <p>int-value</p>
rowspan	<p>Row spanning of control.</p> <p>If you use TR table rows then you may sometimes want to control the number of rows your control occupies. By default it is "1" - but you may want to define the control two span over more than one columns.</p> <p>The property only makes sense in table rows that are synchronized within one container (i.e. TR, STR table rows). It does not make sense in ITR rows, because these rows are explicitly not synched.</p>	Optional	<p>1</p> <p>2</p> <p>3</p> <p>4</p> <p>5</p> <p>50</p> <p>int-value</p>

invisiblemode	<p>If the visibility of the control is determined dynamically by an adapter property then there are two rendering modes if the visibility is "false":</p> <p>(1) "invisible": the control is not visible.</p> <p>(2) "disabled": the control is deactivated: it is "grayed" and does not show any roll over effects any more.</p>	Optional	invisible cleared
tabindex	<p>Index that defines the tab order of the control. Controls are selected in increasing index order and in source order to resolve duplicates.</p>	Optional	-1 0 1 2 5 10 32767
Label			
name	<p>Text that is displayed inside the control. Please do not specify the name when using the multi language management - but specify a "textid" instead.</p>	Optional	
textid	<p>Multi language dependent text that is displayed inside the control. The "textid" is translated into a corresponding string at runtime.</p> <p>Do not specify a "name" inside the control if specifying a "textid".</p>	Optional	
hdistpixelwidth	<p>Width of the distance between checkbox and label in pixel.</p>	Optional	

labelstyle	<p>CSS style definition that is directly passed into this control.</p> <p>With the style you can individually influence the rendering of the control. You can specify any style sheet expressions. Examples are:</p> <p>border: 1px solid #FF0000</p> <p>background-color: #808080</p> <p>You can combine expressions by appending and separating them with a semicolon.</p> <p>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.</p>	Optional	<p>background-color: #FF0000</p> <p>color: #0000FF</p> <p>font-weight: bold</p>
Binding			
valueprop	(already explained above)		
displayprop	<p>Name of adapter property that controls whether the field is displayonly(true) or not (false).</p> <p>By using this property you can dynamically control the "display"-status of the control by your adapter object.</p>	Optional	
statusprop	Name of the adapter property that dynamically passes information how the field should be rendered and how it should act.	Optional	

flush	<p>Flushing behaviour of the input control.</p> <p>By default an input into the control is registered within the browser client - and communicated to the server adapter object when a user e.g. presses a button. By using the FLUSH property you can change this behaviour.</p> <p>Setting FLUSH to "server" means that directly after changing the input a synchronization with the server adapter is triggered. As consequence you directly can react inside your adapter logic onto the change of the corresponding value. - Please be aware of that during the synchronization always all changed properties - also the ones that were changed before - are transferred to the adapter object, not only the one that triggered the synchronization.</p> <p>Setting FLUSH to "screen" means that the changed value is populated inside the page. You use this option if you have redundant usage of the same property inside one page and if you want to pass one changed value to all its representation directly after changing the value.</p>	Optional	screen server
flushmethod	When the data synchronization of the control is set to FLUSH="server" then you can specify an explicit method to be called when the user updates the content of the control. By doing so you can distinguish on the server side from which control the flush of data was triggered.	Optional	
Online Help			
helpid	Help id that is passed to the online help management in case the user presses F1 on the control.	Optional	
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	Text ID that is passed to the multi language management - representing the tooltip text that is used for the control.	Optional	
Miscellaneous			
testtoolid	Use this attribute to assign a fixed control identifier that can be later on used within your test tool in order to do the object identification	Optional	

The RADIOBUTTON control is typically followed by a label explaining its meaning.