COMBOFIX

The COMBOFIX control is a selection control. Depending on its configuration, it is either displayed as a combo box or as a selection list.

The COMBOFIX control allows specifying a defined set of values which can be selected. This set of values is defined as part of the layout definition - it cannot be loaded dynamically from the server.
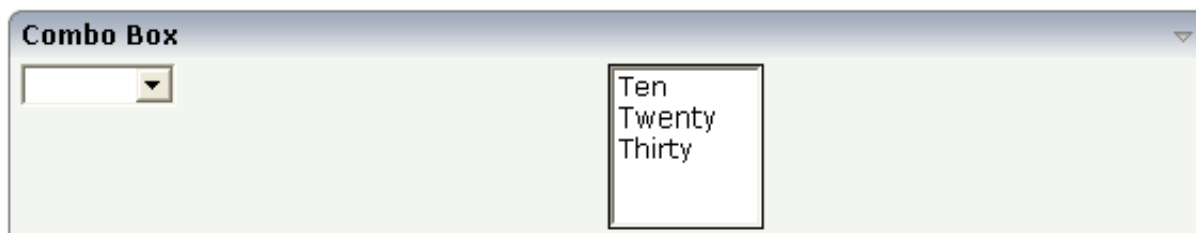
Note:
If you want to use dynamic selection, there are two possibilities. Either use the COMBODYN control which has the same look and feel as the COMBOFIX control, but where the selectable values are not specified as part of the page definition and are derived from an adapter property. Or use the value help pop-up dialogs as described in *Working with Page Navigation*.

The following topics are covered below:

- Example

- Typical Problems with COMBOFIX

- COMBOFIX Properties

- COMBOOPTION Properties

Example

The XML layout code for the example looks as follows:

```
<rowarea name="Combo Box">
    <itr takefullwidth="true">
        <coltable0 width="50%" takefullheight="true">
            <itr>
                <combofix valueprop="factor1" size="1" flush="screen">
                    <combooption name="Ten" value="10">
                    </combooption>
                    <combooption name="Twenty" value="20">
                    </combooption>
                    <combooption name="Thirty" value="30">
                    </combooption>
                </combofix>
            </itr>
        </coltable0>
        <coltable0 width="50%" takefullheight="true">
            <itr takefullwidth="false">
                <combofix valueprop="factor1" size="5" flush="screen">
                    <combooption name="Ten" value="10">
```

```
                    </combooption>
                    <combooption name="Twenty" value="20">
                    </combooption>
                    <combooption name="Thirty" value="30">
                    </combooption>
                </combofix>
            </itr>
        </coltable0>
    </itr>
</rowarea>
```

There is a property `size` inside the COMBOFIX definition. This property specifies the number of lines that are displayed inside the control. If the `size` property is set to "1", a combo box is displayed; if it is set to a value higher than "1", the selection control is displayed.

Inside the COMBOFIX definition, there is a set of COMBOOPTION definitions that represent the selectable values. Each value consists of a display string (`name` property) and a value that is actually set as a property value (`value` property).

The COMBOFIX definition contains a reference to an adapter property (`valueprop` property) into which the value is transferred.

Typical Problems with COMBOFIX

The COMBOFIX control is internally rendered as the HTML control SELECT. Unfortunately, the HTML control has some internal problems and behaves different from normal HTML controls:

- The HTML control SELECT always stays on top of all controls, i.e. it has a "maximum high" z-index and does not allow other controls to overlap.

- The HTML control SELECT does not allow sophisticated style definitions: besides some coloring aspects, you have to accept the look and feel of this control. For example, you cannot turn off its thick borders and you cannot influence the selection image which shows valid values.

Be aware of the consequences of these problems when using the control. Do not place the control into a page are in which a menu might overlap. The COMBOFIX will always be on top of the menu.

COMBOFIX Properties

Basic			
valueprop	Server side property representation of the control.	Obligatory	

width	Width of the control.	Optional	100
	There are three possibilities to define the width:		120
	(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.		140
			160
	(B) Pixel sizing: just input a number value (e.g. "100").		180
	(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.		200
			50%
			100%
comment	Comment without any effect on rendering and behaviour. The comment is shown in the layout editor's tree view.	Optional	
Appearance			
width	(already explained above)		
size	Number of rows that are displayed inside the control. If specified as "1" (default) then the control is rendered as combo box - if ">1" then the control is rendered as multi line selection.	Optional	
displayonly	If set to true, the FIELD will not be accessible for input. It is just used as an output field.	Optional	
align	Horizontal alignment of control in its column.	Optional	left
	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.		center
			right
	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.		
valign	Vertical alignment of control in its column.	Optional	top
	Each control is "packaged" into a column. The column itself is part of a row (e.g. ITR or TR). Sometimtes 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.		middle
			bottom

colspan	Column spanning of control. 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. The property only makes sense in table rows that are snychronized within one container (i.e. TR, STR table rows). It does not make sense in ITR rows, because these rows are explicitly not synched.	Optional	1 2 3 4 5 50 int-value
rowspan	Row spanning of control. 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. The property only makes sense in table rows that are snychronized within one container (i.e. TR, STR table rows). It does not make sense in ITR rows, because these rows are explicitly not synched.	Optional	1 2 3 4 5 50 int-value
combostyle	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	
invisiblemode	If the visibility of the control is determined dynamically by an adapter property then there are two rendering modes if the visibility is "false": (1) "invisible": the control is not visible. (2) "disabled": the control is deactivated: it is "grayed" and does not show any roll over effects any more.	Optional	invisible cleared

tabindex	Index that defines the tab order of the control. Controls are selected in increasing index order and in source order to resolve duplicates.	Optional	-1 0 1 2 5 10 32767
Binding			
valueprop	(already explained above)		
displayprop	Name of adapter property that controls whether the field is displayonly(true) or not (false). By using this property you can dynamically control the "display"-status of the control by your adapter object.	Optional	
statusprop	Name of the adapter property that dynamically passes information how the field should be rendered and how it should act.	Optional	
flush	Flushing behaviour of the input control. 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. 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 synchonization. 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 representaion directly after changing the value.	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	
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	

COMBOOPTION Properties

Basic			
name	Name that is displayed as selectable option. Either use the NAME property to specify the text in a "hard" way or use the TEXTID property to define the text in a language dependent way.	Optional	
textid	Text ID that is used for this option. The text id is passed to the multi language management in order to find a language dependent text.	Optional	
value	Actual value of the option that is passed into the adapter property specified by VALUEPROP inside the COMBOFIX control.	Optional	
comment	Comment without any effect on rendering and behaviour. The comment is shown in the layout editor's tree view.	Optional	