

SUBPAGE

The SUBPAGE control defines an area in which an HTML page is shown. The URL of the page is not statically defined, but is derived from a value of an adapter property.

Due to the browser's capability to embed installed plug-ins, you can use non-HTML objects to be called - and which the browser is able to understand. For example, if you have Microsoft Office installed (or the viewers for Microsoft Office documents) and you pass the name of a Word document as the URL, the Word document will be embedded into the page.

The following topics are covered below:

- Example
 - Typical Problem: Non-Refreshing Subpages
 - Properties
-

Example

```
<rowarea name="Subpage">
  <itr>
    <label name="URL" width="100">
      </label>
      <field valueprop="url" flush="true" length="30">
        </field>
      </itr>
    <vdist>
  </vdist>
  <vdist>
</vdist>
  <itr>
    <subpage valueprop="url" height="300" width="400">
      </subpage>
    </itr>
</rowarea>
```

The above XML layout definition produces a page which looks as follows:



Typical Problem: Non-Refreshing Subpages

Sometimes the SUBPAGE control is used to embed a generated HTML page into an existing page: the adapter program somehow creates an HTML document (e.g. a certain list output) and saves this output to the file system in such a way that it can be reached via a URL. This URL is passed back as a value into the SUBPAGE control.

Pay attention to the fact that the SUBPAGE control only refreshes its content if the URL changes. If the URL stays constant, the SUBPAGE control does not refresh its inner content. Consequence: if you have situations in which the user stays on the same page and creates new subpages multiple times, these subpages must have a URL which changes on every new creation.

A way to force this reloading of a subpage is to append a parameter to the URL which changes every time you want to refresh. Example: if the page's URL is *http://xyz.xyz.html*, then you can write your adapter program in the following way:

```
public void getPageURL()
{
    return m_pageURL;
}

public void onReloadPageURL()
{
    m_pageURL = "http://xyz.xyz.html?DUMMPARAM=" + (new Date()).getTime();
}
```

Check the reload behavior with your web server first.

Properties

Basic			
valueprop	<p>Name of adapter property providing the URL to be displayed inside the SUBPAGE control.</p> <p>The URL may either be an absolute URL ("http://...") or may be a relative URL.</p> <p>Please note: the SUBPAGE control only re-renders its inner content if the URL provided by the property really changes. The SUBPAGE control does not "know" if something changed inside the contained page and that it has to redraw the page. - If you want to refresh the inner page explicitly append some random number to your URL, e.g.: http://...url...?RANDOM=45435. By changing the number the browser will reload the URL.</p>	Obligatory	
width	<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>	Sometimes obligatory	<p>100</p> <p>120</p> <p>140</p> <p>160</p> <p>180</p> <p>200</p> <p>50%</p> <p>100%</p>

height	Height of the control.	Sometimes obligatory	100
	There are three possibilities to define the height:		150
	(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.		200
			250
			300
	(B) Pixel sizing: just input a number value (e.g. "20").		250
	(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.		400
		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)		
height	(already explained above)		
scrolling	Definition of the 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 yes no

<p>pagestyle</p>	<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>	<p>Optional</p>	
<p>colspan</p>	<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>	<p>Optional</p>	<p>1 2 3 4 5 50 int-value</p>
<p>rowspan</p>	<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>	<p>Optional</p>	<p>1 2 3 4 5 50 int-value</p>
<p>Binding</p>			
<p>valueprop</p>	<p>(already explained above)</p>		