

# ROWAREA and COLAREA

The ROWAREA or COLAREA container represents an area surrounded by a border and which may have a title text. By clicking on the title of such a control, the inner content is hidden (the ROWAREA or COLAREA is "folded").

The following topics are covered below:

- ROWAREA Properties
- COLAREA Properties

---

## ROWAREA Properties

Basic			
name	Text that is displayed inside the control. Please do not specify the name when using the multi language management - but specify a "textid" instead.	Sometimes obligatory	
textid	Multi language dependent text that is displayed inside the control. The "textid" is translated into a corresponding string at runtime.  Do not specify a "name" inside the control if specifying a "textid".	Sometimes obligatory	
nameprop	Name of adapter property that provides as value the text that is shown inside the control.	Optional	

height	<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>	Optional	<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>
comment	Comment without any effect on rendering and behaviour. The comment is shown in the layout editor's tree view.	Optional	
Visibility			
foldable	The "folding"-function that is available by clicking on the title of the area can be switched off ("false"). "True" is the default.	Optional	<p>true</p> <p>false</p>
foldableprop	<p>Name of adapter property that dynamically controls whether clicking on the title of the area will fold/unfoald this area.</p> <p>Valid values provided by the adapter property are TRUE (=foldable) and FALSE(=not foldable).</p>	Optional	
foldedprop	<p>Name of adapter property that controls whether the content of the ROWAREA is folded (true) or displayed (false).</p> <p>By using this property you can dynamically control the "folded"-status of the control by your adapter object.</p>	Optional	

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	
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 representaion directly after changing the value.</p>	Optional	screen server
flushmethod	<p>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.</p>	Optional	
Appearance			

image	<p>URL of image that is displayed inside the control. Any image type (.gif, .jpg, ...) that your browser does understand is valid.</p> <p>Use the following options to specify the URL:</p> <p>(A) Define the URL relative to your page. Your page is generated directly into your project's folder. Specifying "images/xyz.gif" will point into a directory parallel to your page. Specifying "../HTMLBasedGUI/images/new.gif" will point to an image of a neighbour project.</p> <p>(B) Define a complete URL, like "http://www.softwareag.com/images/logo.gif".</p>	Optional	
imageprop	<p>Name of adapter property that provides as value the URL of the image that is shown inside the control.</p> <p>The URL must either be an absolute URL or a relative URL.</p>	Optional	
withtoppadding	<p>The control by default renders some blank vertical space (2 pixels) on top of its area. Reason: if you vertically arrange one ROW/COLAREA after the other then automatically some distance is put between.</p> <p>By specifying "false" you can avoid this behaviour. "</p>	Optional	true false
withleftborder	<p>The control normally renders a black border around its area. With the properties WITHLEFTBORDER, WITHRIGHTBORDER and WITHBOTTOMBORDER you can avoid this.</p> <p>Reason being: sometimes you want a ROWAREA/COLAREA to be used as "neighbour" of other ROWAERA/COLAREA controls. In this case one of the "neighbours" has to avoid the rendering of border lines - otherwise two border lines will be rendered.</p>	Optional	true false
withtopborder	See description of WITHLEFTBORDER property.	Optional	true false
withrightborder	See description of WITHLEFTBORDER property.	Optional	true false

withbottomborder	See description of WITHLEFTBORDER property.	Optional	true false
paddingleft	Number of pixels between the left border and the area's content. Default is 5 pixels.	Optional	1 2 3 int-value
paddingright	Number of pixels between the right border and the area's content. Default is 5 pixels.	Optional	1 2 3 int-value
areastyle	<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	background-color: #FF0000  color: #0000FF  font-weight: bold
contenttablestyle	CSS style definition that is applied to the content part of the ROWAREA control.	Optional	background-color: #FF0000  color: #0000FF  font-weight: bold

notabstop	<p>The title of the area by default can be used by the user to hide/show the area's content. In order to also reach this title with the tab-key is is part of the normal tab-sequence of a page.</p> <p>Set this property to "true" if you do not want to make the title reachable by tab-key. As consequence hiding/showing will only be available by mouse-clicking on the title.</p>	Optional	true false
fixlayout	<p>The fixlayout property is important for saving rendering performance inside your browser. To become effective it requires to have specified the height and the width (if available as property) of the control.</p> <p>If setting fixlayout to "true" then the control's area is defined as area which is not sized dependent on its content (as normally done with table rendering). Instead the size is predefined from outside without letting the browser "look" into the content of the area. If the content is not fitting into the area then it is cut.</p> <p>You typically use this control if the content of the control's area is flexibly sizable. E.g. if the content (e.g. a TEXTGRID control) is following the size of the container.</p> <p>When using vertical percentage based sizing you should pay attention to set the fixlayout-property to "true" as often as possible. - The browser as consequence will be much faster in doing its rendering because a screen consists out of "building blocks" with simple to calculate sizes.</p>	Optional	true false
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

withcontenttoppadding	<p>The control by default renders some blank vertical space (3 pixels) on bottom of the content area.</p> <p>By specifying "false" you can avoid this behaviour.</p>	Optional	true false
withcontentbottompadding	<p>The control by default renders some blank vertical space (3 pixels) on bottom of the content area.</p> <p>By specifying "false" you can avoid this behaviour.</p>	Optional	true false
withfadedtoggling	<p>The animation of the controls can be switched off! Please take a look in your cisconfig.xml file. Set animatecontrols="true" (default) if you generally want to animate all of your controls.</p> <p>The rowarea control has a seperate switch (withfadedtoggling = true/false) to (de)activate the 'FadedToggling' animation especially for this single rowarea control.</p> <p>Notice: Entering true or false into the withfadedtoggling attribute overwrites the general animatecontrols setting !</p>	Optional	true false
stylevariant	<p>Some controls offer the possibility to define style variants. By this style variant you can address different styles inside your style sheet definition file (.css). If not defined "normal" styles are chosen, if defined (e.g. "VAR1") then other style definitions (xxxVAR1xxx) are chosen.</p> <p>Purpose: you can set up style variants in the style sheet defintion and use them multiple times by addressing them via the "stylevariant" property. CIS currently offerst two variants "VAR1" and "VAR2" but does not predefine any semantics behind - this is up to you!</p>	Optional	
titlerowontop	<p>Default value is 'true'. If set to 'false' the titlerow is rendered at the bottom of this area.</p>	Optional	true false
toggleimgtitle	<p>A text that is displayed as tooltip of the toggle image.</p>	Optional	

toggleimgtitletextid	Multi language dependent text that is displayed as tooltip of the toggle image.  Do not specify a "toggleimagetitle" inside the control if specifying a "toggleimagetextid".	Optional	
Online Help			
title	Text that is shown as tooltip for the control.  Either specify the text "hard" by using this TITLE property - or use the TITLETEXTID in order to define a language dependent literal.	Optional	
titletextid	Text ID that is passed to the multi lanaguage management - representing the tooltip text that is used for the control.	Optional	

## COLAREA Properties

The properties of COLAREA are very similar to those of ROWAREA.

Basic			
name	Text that is displayed inside the control. Please do not specify the name when using the multi language management - but specify a "textid" instead.	Sometimes obligatory	
textid	Multi language dependent text that is displayed inside the control. The "textid" is translated into a corresponding string at runtime.  Do not specify a "name" inside the control if specifying a "textid".	Sometimes obligatory	
nameprop	Name of adapter property that provides as value the text that is shown inside the control.	Optional	



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>
comment	Comment without any effect on rendering and behaviour. The comment is shown in the layout editor's tree view.	Optional	
Appearance			
takefullheight	<p>Indicates if the content of the control's area gets the full available height.</p> <p>If you use percentage sizing inside the control's area then this property must be switched to 'true'. If you use no explicit vertical sizing at all - or you use vertical pixel sizing for your controls - the property must be switched to 'false'.</p> <p>Background information: container control's internally open up a table in which you place rows (ITR/TR) which then hold controls (e.g. LABEL/FIELD). The table that is opened up normally has no explicit height and grows with its content as consequence. By specifying "takefullheight=true" the table itself is sized to fill the maximum height of the available area.</p>	Optional	<p>true</p> <p>false</p>

image	<p>URL of image that is displayed inside the control. Any image type (.gif, .jpg, ...) that your browser does understand is valid.</p> <p>Use the following options to specify the URL:</p> <p>(A) Define the URL relative to your page. Your page is generated directly into your project's folder. Specifying "images/xyz.gif" will point into a directory parallel to your page. Specifying "../HTMLBasedGUI/images/new.gif" will point to an image of a neighbour project.</p> <p>(B) Define a complete URL, like "http://www.softwareag.com/images/logo.gif".</p>	Optional	
imageprop	<p>Name of adapter property that provides as value the URL of the image that is shown inside the control.</p> <p>The URL must either be an absolute URL or a relative URL.</p>	Optional	
fixlayout	<p>The fixlayout property is important for saving rendering performance inside your browser. To become effective it requires to have specified the height and the width (if available as property) of the control.</p> <p>If setting fixlayout to "true" then the control's area is defined as area which is not sized dependent on its content (as normally done with table rendering). Instead the size is predefined from outside without letting the browser "look" into the content of the area. If the content is not fitting into the area then it is cut.</p> <p>You typically use this control if the content of the control's area is flexibly sizable. E.g. if the content (e.g. a TEXTGRID control) is following the size of the container.</p> <p>When using vertical percentage based sizing you should pay attention to set the fixlayout-property to "true" as often as possible. - The browser as consequence will be much faster in doing its rendering because a screen consists out of "building blocks" with simple to calculate sizes.</p>	Optional	true false

withleftborder	<p>The control normally renders a black border around its area. With the properties WITHLEFTBORDER, WITHRIGHTBORDER and WITHBOTTOMBORDER you can avoid this.</p> <p>Reason behing: somtimes you want a ROWAREA/COLAREA to be used as "neighbour" of other ROWAERA/COLAREA controls. In this case one of the "neighbours" has to avoid the rendering of border lines - otherwise two border lines will be rendered.</p>	Optional	true false
withtopborder	See description of WITHLEFTBORDER property.	Optional	true false
withrightborder	See description of WITHLEFTBORDER property.	Optional	true false
withbottomborder	See description of WITHLEFTBORDER property.	Optional	true false
paddingleft	Number of pixels between the left border and the area's content. Default is 5 pixels.	Optional	1 2 3 int-value
paddingright	Number of pixels between the right border and the area's content. Default is 5 pixels.	Optional	1 2 3 int-value

areastyle	<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>
contenttablestyle	<p>CSS style that is applied to the content area of the COLAREA control.</p>	Optional	<p>background-color: #FF0000</p> <p>color: #0000FF</p> <p>font-weight: bold</p>
withcontenttoppadding	<p>The control by default renders some blank vertical space (3 pixels) on bottom of the content area.</p> <p>By specifying "false" you can avoid this behaviour.</p>	Optional	<p>true</p> <p>false</p>
withcontentbottompadding	<p>The control by default renders some blank vertical space (3 pixels) on bottom of the content area.</p> <p>By specifying "false" you can avoid this behaviour.</p>	Optional	<p>true</p> <p>false</p>
titlerowontop	<p>Default value is 'true'. If set to 'false' the titlerow is rendered at the bottom of this area.</p>	Optional	<p>true</p> <p>false</p>

stylevariant	<p>Some controls offer the possibility to define style variants. By this style variant you can address different styles inside your style sheet definition file (.css). If not defined "normal" styles are chosen, if defined (e.g. "VAR1") then other style definitions (xxxVAR1xxx) are chosen.</p> <p>Purpose: you can set up style variants in the style sheet definition and use them multiple times by addressing them via the "stylevariant" property. CIS currently offerst two variants "VAR1" and "VAR2" but does not predefine any semantics behind - this is up to you!</p>	Optional	
withtoppadding	<p>The control by default renders some blank vertical space (2 pixels) on top of its area. Reason: if you vertically arrange one ROW/COLAREA after the other then automatically some distance is put between.</p> <p>By specifying "false" you can avoid this behaviour. "</p>	Optional	true false
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	