


# Label

 Apr 05, 2024 20:40

Supported Clients

SmartClient WebClient NGClient MobileClient

Property Summary		
Number	anchors	Enables a component to stick to a specific side of form and/or to grow or shrink when a window is resized.
String	background	The background color of the component.
String	borderType	The type, color and style of border of the component.
String	comment	
CSSPosition	cssPosition	Thecss position of the component.
String	dataProvider	The dataprovider of the component.
Object	designTimeProperties	Property to get and add design-time properties for a component.
Boolean	displaysTags	Flag that enables or disables merging of data inside components using tags (placeholders).
Boolean	enabled	The enable state of the component, default true.
String	fontType	The font type of the component.
String	foreground	The foreground color of the component.
Number	formIndex	Form index returns the z-index of the form element.
String	format	The format that should be applied when displaying data(using dataProviderID) in the label /button.
Number	horizontalAlignment	Horizontal alignment of the text inside the component.
Number	imageMedia	The image Media object that should be displayed inside the component.
String	labelFor	Some components can be set to be labels of other components.
Number	labelSize	Label (header) size property.
String	location	The x and y position of the component, in pixels, separated by a comma.
String	margin	The margins of the component.
Number	mediaOptions	Options to scale the image Media object that is displayed inside the component.
String	mnemonic	The keyboard shortcut that activates this component.
String	name	The name of the component.
Number	printSliding	Enables an element to resize based on its content and/or move when printing.
Boolean	printable	Flag that tells if the component should be printed or not when the form is printed.
Number	rolloverCursor	The cursor that is shown as the mouse is rolled over the component.
Number	rolloverImageMedia	The roll over image Media object used.
Boolean	showClick	When set, the element will show the clicked state when selected.
Boolean	showFocus	When set the text of an element will showfocus when selected.
String	size	The width and height (in pixels), separated by a comma.
String	styleClass	The name of the style class that should be applied to this component.
Number	tabSeq	An index that specifies the position of the component in the tab sequence.
String	text	The text that is displayed inside the component.
Number	textRotation	The rotation of the element.
Boolean	titleDisplaysTags	Flag for header text to field component that enables or disables merging of data inside components using tags (placeholders).
Boolean	titleVisible	
String	toolTipText	The text displayed when hovering over the component with a mouse cursor.
Boolean	transparent	Flag that tells if the component is transparent or not.
Number	verticalAlignment	The vertical alignment of the text inside the component.
Boolean	visible	The visible property of the component, default true.

Event Summary		
void	onAction()	
void	onDoubleClick(event)	The method that is executed when the component is double clicked.
void	onRender()	The method that is executed when the component is rendered.
void	onRightClick(event)	The method that is executed when the component is right clicked.

Property Details

anchors

---

Enables a component to stick to a specific side of form and/or to grow or shrink when a window is resized.

If opposite anchors are activated then the component with grow or shrink with the window. For example if Top and Bottom are activated, then the component will grow/shrink when the window is vertically resized. If Left and Right are activated then the component will grow/shrink when the window is horizontally resized.

If opposite anchors are not activated, then the component will keep a constant distance from the sides of the window which correspond to the activated anchors.

**Returns**

[Number](#)

**Supported Clients**

SmartClient,WebClient,NGClient

**Sample****background**

The background color of the component.

**Returns**

[String](#)

**Supported Clients**

SmartClient,WebClient,NGClient

**Sample****borderType**

The type, color and style of border of the component.

**Returns**

[String](#)

**Supported Clients**

SmartClient,WebClient,NGClient

**Sample****comment****Returns**

[String](#)

**Supported Clients**

SmartClient,WebClient,NGClient

**Sample****cssPosition**

Thecss position of the component.

**Returns**

[CSSPosition](#)

**Supported Clients**

SmartClient,WebClient,NGClient

**Sample****dataProvider**

The dataprovider of the component.

**Returns**

[String](#)

---

**Supported Clients**

SmartClient,WebClient,NGClient,MobileClient

**Sample****designTimeProperties**

Property to get and add design-time properties for a component.

**Returns**

[Object](#) map of the design-time properties

**Supported Clients**

SmartClient,WebClient,NGClient

**Sample****displaysTags**

Flag that enables or disables merging of data inside components using tags (placeholders). Tags (or placeholders) are words surrounded by %% on each side. There are data tags and standard tags. Data tags consist in names of dataproviders surrounded by %%. Standard tags are a set of predefined tags that are made available by the system.

See the "Merging data" section for more details about tags.

The default value of this flag is "false", that is merging of data is disabled by default.

**Returns**

[Boolean](#)

**Supported Clients**

SmartClient,WebClient,NGClient,MobileClient

**Sample****enabled**

The enable state of the component, default true.

**Returns**

[Boolean](#) enabled state

**Supported Clients**

SmartClient,WebClient,NGClient,MobileClient

**Sample****fontType**

The font type of the component.

**Returns**

[String](#)

**Supported Clients**

SmartClient,WebClient,NGClient

**Sample****foreground**

The foreground color of the component.

**Returns**

[String](#)

**Supported Clients**

SmartClient,WebClient,NGClient

---

**Sample****formIndex**

Form index returns the z-index of the form element. If two elements overlap, then the element with higher Z index is displayed above the form element with lower Z index. In case of form inheritance, form index is evaluated in the context (form) where property is defined. Elements which have child form as context will always be on top of elements which have parent form. If element is overridden in child form but form index property is still inherited its context is still parent form.

**Returns**

[Number](#)

**Supported Clients**

SmartClient,WebClient,NGClient

**Sample****format**

The format that should be applied when displaying data(using dataProviderID) in the label/button. Some examples are "%", "dd-MM-yyyy", "MM-dd-yyyy", etc.

**Returns**

[String](#)

**Supported Clients**

SmartClient,WebClient,NGClient,MobileClient

**Sample****horizontalAlignment**

Horizontal alignment of the text inside the component. Can be one of LEFT, CENTER or RIGHT.

Note that this property does not refer to the horizontal alignment of the component inside the form.

**Returns**

[Number](#)

**Supported Clients**

SmartClient,WebClient,NGClient

**Sample****imageMedia**

The image Media object that should be displayed inside the component.

**Returns**

[Number](#)

**Supported Clients**

SmartClient,WebClient,NGClient

**Sample****labelFor**

Some components can be set to be labels of other components. This is useful in two situations. In table view mode it is used for constructing the header of the table. In record view mode, by setting mnemonics on the label, keyboard shortcuts can be used to set the focus to fields.

**Returns**

[String](#)

**Supported Clients**

SmartClient,WebClient,NGClient

---

**Sample****labelSize**

Label (header) size property.  
Sets the header size on a label. Valid values are between 'h1' to 'h6',  
default value is 'h4'.  
The numbers match the heading tags h1 to h6 as used in html.

**Returns**

[Number](#)

**Supported Clients**

NGClient, MobileClient

**Sample****location**

The x and y position of the component, in pixels, separated by a comma.

**Returns**

[String](#)

**Supported Clients**

SmartClient, WebClient, NGClient, MobileClient

**Sample****margin**

The margins of the component. They are specified in this order,  
separated by commas: top, left, bottom, right.

**Returns**

[String](#)

**Supported Clients**

SmartClient, WebClient, NGClient

**Sample****mediaOptions**

Options to scale the image Media object that is displayed inside the component.  
Can be set to one or a combination of CROP, REDUCE, ENLARGE and KEEPASPECT.

REDUCE will scale down the image if the component is smaller than the image.  
REDUCE combined with KEEPASPECT will reduce the image, but keep its aspect ratio.  
This is useful when the component has other proportions than the image.

ENLARGE will scale up the image if the component is larger than the image.  
ENLARGE combined with KEEPASPECT will scale up the image while keeping its aspect ratio.

CROP will leave the image at its original size. If the component is smaller than  
the image this will result in only a part of the image showing up.

**Returns**

[Number](#)

**Supported Clients**

SmartClient, WebClient, NGClient

**Sample****mnemonic**

---

The keyboard shortcut that activates this component. A letter must be specified, and the actual shortcut will be combination of ALT + the specified letter.

This property can be used in two ways. Normally the keyboard shortcut activates the onClick event of the component. But if the "labelFor" property is set for the component, then the keyboard shortcut will move the focus to the component whose label this component is.

**Returns**

[String](#)

**Supported Clients**

SmartClient,WebClient,NGClient

**Sample****name**

The name of the component. Through this name it can also accessed in methods. Must be a valid javascript name. (no - in the name or start with number)

**Returns**

[String](#)

**Supported Clients**

SmartClient,WebClient,NGClient,MobileClient

**Sample****printSliding**

Enables an element to resize based on its content and/or move when printing. The component can move horizontally or vertically and can grow or shrink in height and width, based on its content and the content of neighboring components.

**Returns**

[Number](#)

**Supported Clients**

SmartClient,WebClient

**Sample****printable**

Flag that tells if the component should be printed or not when the form is printed.

By default components are printable.

**Returns**

[Boolean](#)

**Supported Clients**

SmartClient,WebClient

**Sample****rolloverCursor**

The cursor that is shown as the mouse is rolled over the component. Possible options are DEFAULT and HAND. Note that roll over cursor is not supported in Smart client for list view and tableview forms.

**Returns**

[Number](#)

**Supported Clients**

SmartClient,WebClient,NGClient

**Sample****rolloverImageMedia**

---

The roll over image Media object used. It will only work if a property image is also used. When the mouse is moved over the component, this image Media will be displayed. When the mouse is moved out of the component, whatever text or image was being initially displayed will be restored. Note that roll over image is not supported in Smart client for list view and tableview forms.

**Returns**[Number](#)**Supported Clients**

SmartClient,WebClient,NGClient

**Sample****showClick**

When set, the element will show the clicked state when selected.  
Applies to labels and buttons and images only.

**Returns**[Boolean](#)**Supported Clients**

SmartClient,WebClient,NGClient

**Sample****showFocus**

When set the text of an element will showfocus when selected.  
Applies to labels and buttons only.  
The text property for the element MUST be filled in first.

NOTE: The TAB key may also be used to select the element, depending on the operating system being used and the selected LAF.

**Returns**[Boolean](#)**Supported Clients**

SmartClient,WebClient,NGClient

**Sample****size**

The width and height (in pixels), separated by a comma.

**Returns**[String](#)**Supported Clients**

SmartClient,WebClient,NGClient

**Sample****styleClass**

The name of the style class that should be applied to this component.

When defining style classes for specific component types, their names must be prefixed according to the type of the component. For example in order to define a class names 'fancy' for fields, in the style definition the class must be named 'field.fancy'. If it would be intended for labels, then it would be named 'label.fancy'. When specifying the class name for a component, the prefix is dropped however. Thus the field or the label will have its styleClass property set to 'fancy' only.

**Returns**[String](#)**Supported Clients**

SmartClient,WebClient,NGClient

**Sample**

**tabSeq**

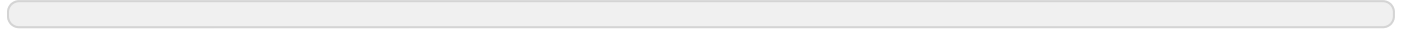
An index that specifies the position of the component in the tab sequence. The components are put into the tab sequence in increasing order of this property. A value of 0 means to use the default mechanism of building the tab sequence (based on their location on the form). A value of -2 means to remove the component from the tab sequence.

**Returns**

[Number](#)

**Supported Clients**

SmartClient,WebClient,NGClient

**Sample**

**text**

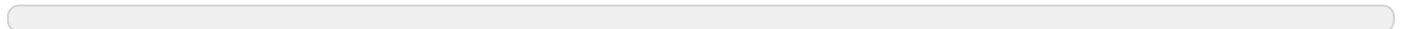
The text that is displayed inside the component.

**Returns**

[String](#)

**Supported Clients**

SmartClient,WebClient,NGClient,MobileClient

**Sample**

**textRotation**

The rotation of the element. You can choose 0, 90, 180, or 270 and the label is rotated accordingly. This property also applies to buttons and images.

**Returns**

[Number](#)

**Supported Clients**

SmartClient,WebClient,NGClient

**Sample**

**titleDisplaysTags**

Flag for header text to field component that enables or disables merging of data inside components using tags (placeholders).

Tags (or placeholders) are words surrounded by %% on each side. There are data tags and standard tags. Data tags consist in names of dataproviders surrounded by %%. Standard tags are a set of predefined tags that are made available by the system.

See the "Merging data" section for more details about tags.

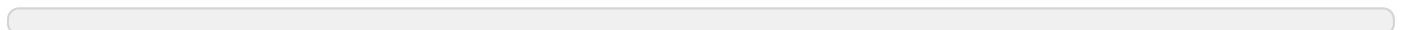
The default value of this flag is "false", that is merging of data is disabled by default.

**Returns**

[Boolean](#)

**Supported Clients**

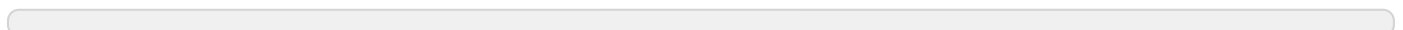
SmartClient,WebClient

**Sample**

**titleVisible****Returns**

[Boolean](#)

**Supported Clients**

NGClient,MobileClient

**Sample**




---

**toolTipText**

The text displayed when hovering over the component with a mouse cursor.

**NOTE:**

HTML should be used for multi-line tooltips; you can also use any valid HTML tags to format tooltip text. For example:  
<html>This includes<b>bolded text</b> and  
<font color='blue'>BLUE</font> text as well.</html>

**Returns**

[String](#)

**Supported Clients**

SmartClient,WebClient,NGClient

**Sample****transparent**

Flag that tells if the component is transparent or not.

The default value is "false", that is the components are not transparent.

**Returns**

[Boolean](#)

**Supported Clients**

SmartClient,WebClient,NGClient

**Sample****verticalAlignment**

The vertical alignment of the text inside the component. Can be one of TOP, CENTER or BOTTOM.

Note that this property does not refer to the vertical alignment of the component inside the form.

**Returns**

[Number](#)

**Supported Clients**

SmartClient,WebClient,NGClient

**Sample****visible**

The visible property of the component, default true.

**Returns**

[Boolean](#) visible property

**Supported Clients**

SmartClient,WebClient,NGClient,MobileClient

**Sample**

---

**Event Details****onAction()****Supported Clients**

SmartClient,WebClient,NGClient

**Sample****onDoubleClick(event)**

---

The method that is executed when the component is double clicked.

**Parameters**

[JSEvent](#) event the event that triggered the action

**Supported Clients**

SmartClient,WebClient,NGClient

**Sample****onRender()**

The method that is executed when the component is rendered.

**Supported Clients**

SmartClient,WebClient

**Sample****onRightClick(event)**

The method that is executed when the component is right clicked.

**Parameters**

[JSEvent](#) event the event that triggered the action

**Supported Clients**

SmartClient,WebClient,NGClient

**Sample**