


# RuntimeTextArea

 Apr 03, 2024 10:48

Supported Clients

SmartClient WebClient NGClient

Extends

RuntimeComponent

Property Summary		
String	bgcolor	Gets or sets the background color of a field.
String	border	Gets or sets the border attribute(s) of a specified element.
Number	caretPosition	Gets or sets the number value (position) of the text caret (text "I" bar) in a field.
Boolean	editable	Gets or sets the editable/read-only state of a field; true - editable; false - read-only.
Boolean	enabled	Gets or sets the enabled state of a specified field, also known as "grayed".
String	fgcolor	Gets or sets the foreground color of a field.
String	font	Gets or sets the font name, style, and size of an element.
String	placeholderText	Gets or sets the placeholder text in a field.
Boolean	readOnly	Gets or sets the editable/read-only state of a field; true - read-only; false - editable; ! - the editable/read-only state is inverted (the opposite).
String	titleText	Gets or sets the title text.
String	toolTipText	Gets or sets the tool tip text of an element; text displays when the mouse cursor hovers over an element.
Boolean	transparent	Gets or sets the transparency of an element; true - transparent; false - not transparent.
Boolean	visible	Gets or sets the visibility of an element; true - visible; false - not visible; ! - the visibility state is inverted (the opposite).

Methods Summary		
void	addStyleClass(styleName)	Adds a style to the styleClass property.
Number	getAbsoluteFormLocationY()	Returns the absolute form (designed) Y location.
Object	getClientProperty(key)	Gets the specified client property for the element based on a key.
String	getDataProviderID()	Get the data provider this UI element (display) is showing.
Object	getDesignProperties()	Get the design-time properties of an element.
Object	getDesignTimeProperty(key)	Get a design-time property of an element.
String	getElementType()	Returns the type of a specified element.
String	getFormName()	Returns the name of the form.
Number	getHeight()	Returns the height of the current element.
Array	getLabelForElementNames()	Returns an Array of label element names that has this field filled in as the labelFor.
Number	getLocationX()	Returns the x location of the current element.
Number	getLocationY()	Returns the y location of the current element.
String	getName()	Returns the name of an element.
Number	getScrollX()	Returns the x scroll location of specified element - only for an element where height of element is less than the height of element content.
Number	getScrollY()	Returns the y scroll location of specified element - only for an element where height of element is less than the height of element content.
String	getSelectedText()	Returns the currently selected text in the specified text field or text area.
Number	getWidth()	Returns the width of the current element.
void	hasStyleClass(styleName)	Check if an element already have a style from the styleClass property.
void	putClientProperty(key, value)	Sets the value for the specified element client property key.
void	removeStyleClass(styleName)	Removes a style from the styleClass property.
void	replaceSelectedText(s)	Replaces the selected text; if no text has been selected, the replaced value will be inserted at the last cursor position.
void	requestFocus()	Request the focus in this element.
void	requestFocus (mustExecuteOnFocusGainedMethod)	Request the focus in this element.
void	selectAll()	Selects all the contents of a field.
void	setLocation(x, y)	Sets the location of an element.
void	setScroll(x, y)	Sets the scroll location of an element.
void	setSize(width, height)	Sets the size of an element.

Property Details

bgcolor

Gets or sets the background color of a field. The color has to be set using the hexadecimal RGB value as used in HTML.

It only returns it's correct value if it was explicitly set.

#### Returns

[String](#)

#### Supported Clients

SmartClient,WebClient,NGClient

#### Sample

```
//sets the background color of the field
%%elementName%%.bgcolor = "#FFFFFF";
//gets the background color of the field
var c = %%elementName%%.bgcolor;
```

#### border

Gets or sets the border attribute(s) of a specified element.

The border attributes:

borderType - EmptyBorder, EtchedBorder, BevelBorder, LineBorder, TitleBorder, MatteBorder, SpecialMatteBorder.  
 size - (numeric value) for: bottom, left, right, top.  
 color - (hexadecimal value) for: bottom, left, right, top.  
 dash pattern - (numeric value) for selected side(s).  
 rounding radius - (numeric value) for selected side(s).

It only returns it's correct value if it was explicitly set.

NOTE: Use the same value(s) and order of attribute(s) from the element design time property "borderType".

#### Returns

[String](#)

#### Supported Clients

SmartClient,WebClient,NGClient

#### Sample

```
//sets the border type to "LineBorder"
//sets a 1 px line width for the bottom and left side of the border
//sets the hexadecimal color of the border to "#ccffcc"
%%elementName%%.border = 'LineBorder,1,#ccffcc';
```

#### caretPosition

Gets or sets the number value (position) of the text caret (text "I" bar) in a field.

NOTE: This does not work in Web Client.

#### Returns

[Number](#)

#### Supported Clients

SmartClient,WebClient,NGClient

#### Sample

```
//get the current caretposition
var caretPos = %%elementName%%.caretPosition;
//add one and set it
%%elementName%%.caretPosition = caretPos+1;
```

#### editable

Gets or sets the editable/read-only state of a field; true - editable; false - read-only.

NOTE the "!" operator can be used to invert the editable state.

#### Returns

[Boolean](#)

#### Supported Clients

SmartClient,WebClient,NGClient

**Sample**

```
var currentState = %%elementName%%.editable;
%%elementName%%.editable = !currentState;
```

**enabled**

Gets or sets the enabled state of a specified field, also known as "grayed".  
true - enabled; false - not enabled; ! - the enabled state is inverted (the opposite).

NOTE: A disabled element cannot be selected by clicking the element (or by pressing the TAB key even if this option is supported by the operating system).

NOTE: A label or button element will not disable if the "displayType" design time property for a field is set to HTML\_AREA.

NOTE: The disabled "grayed" color is dependent on the LAF set in the Servoy Client Application Preferences. For more information see Preferences: Look And Feel in the Servoy Developer User's Guide.

**Returns**

[Boolean](#)

**Supported Clients**

SmartClient,WebClient,NGClient

**Sample**

```
//gets the enabled state of the field
var currState = %%elementName%%.enabled;

//sets the enabled state of the field
%%elementName%%.enabled = !currentState;
```

**fgcolor**

Gets or sets the foreground color of a field. The color has to be set using the hexadecimal RGB value as used in HTML.

It only returns it's correct value if it was explicitly set.

**Returns**

[String](#)

**Supported Clients**

SmartClient,WebClient,NGClient

**Sample**

```
//sets the foreground color of the field
%%elementName%%.fgcolor = "#000000";

//gets the foreground color of the field
var c = %%elementName%%.fgcolor;
```

**font**

Gets or sets the font name, style, and size of an element.

font name - the name of the font family.

style - the type of the font. (plain = 0; bold = 1; italic = 2; bold-italic = 3).

size - the size of the font (in points).

It only returns it's correct value if it was explicitly set.

**Returns**

[String](#)

**Supported Clients**

SmartClient,WebClient,NGClient

**Sample**

```
%%elementName%%.font = 'Tahoma,1,11';
```

**placeholderText**

---

Gets or sets the placeholder text in a field.

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

SmartClient, WebClient, NGClient

**Sample**

```
//get the placeholder text
var placeholderText = %%elementName%%.placeholderText;
//set another placeholder
%%elementName%%.placeholderText = 'Search..';
```

**readOnly**

Gets or sets the editable/read-only state of a field; true - read-only; false - editable; ! - the editable/read-only state is inverted (the opposite).

NOTE: A field set as read-only can be selected by clicking (or pressing the TAB key if this option is supported by the operating system) and the field data can be copied.

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

SmartClient, WebClient, NGClient

**Sample**

```
//gets the editable/read-only state of the field
var currentState = %%elementName%%.readOnly;

//sets the editable/read-only state of the field
%%elementName%%.readOnly = !currentState;
```

**titleText**

Gets or sets the title text.

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

SmartClient, WebClient, NGClient

**Sample**

```
var titleText = %%elementName%%.titleText;
```

**toolTipText**

Gets or sets the tool tip text of an element; text displays when the mouse cursor hovers over an element.

NOTE: HTML should be used for multi-line tooltips; you can also use any valid HTML tags to format tooltip text.

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

SmartClient, WebClient, NGClient

**Sample**

```
//gets the tooltip text of the element
var toolTip = %%elementName%%.toolTipText;

//sets the tooltip text of the element
%%elementName%%.toolTipText = "New tip";
%%elementName%%.toolTipText = "<html>This includes <b>bolded text</b> and <font color='blue'>BLUE</font> text as well.";
```

## transparent

Gets or sets the transparency of an element; true - transparent; false - not transparent.

NOTE: transparency can be inverted using ! operator: `elements.elementName.transparent = !elements.elementName.transparent;`

NOTE: transparency will be mostly used for background color, a transparent element will receive the background of the element "beneath" it, a non transparent one will use its own background color

### Returns

[Boolean](#)

### Supported Clients

SmartClient,WebClient,NGClient

### Sample

```
//gets the transparency of the element
var currentState = %%elementName%%.transparent;

//sets the transparency of the element
%%elementName%%.transparent = !currentState;
```

## visible

Gets or sets the visibility of an element; true - visible; false - not visible; ! - the visibility state is inverted (the opposite).

NOTE: The visibility of an element is not persistent; the state of visibility only applies to the current user in his/her current session.

### Returns

[Boolean](#)

### Supported Clients

SmartClient,WebClient,NGClient

### Sample

```
//sets the element as visible
forms.company.elements.faxBtn.visible = true;

//gets the visibility of the element
var currentState = forms.company.elements.faxBtn.visible;

//sets the element as not visible when the current state is visible
forms.company.elements.faxBtn.visible = !currentState;
```

## Methods Details

### addStyleClass(styleName)

Adds a style to the styleClass property. This works only for NGClient where multiple styles are supported.

### Parameters

[String](#) styleName the name of the style class to add

### Supported Clients

NGClient

### Sample

```
%%elementName%%.addStyleClass('redbg');
```

### getAbsoluteFormLocationY()

Returns the absolute form (designed) Y location.

### Returns

[Number](#) The y location of the form in pixels.

### Supported Clients

SmartClient,WebClient,NGClient

---

**Sample**

```
var absolute_y = %%elementName%%.getAbsoluteFormLocationY();
```

**getClientProperty(key)**

Gets the specified client property for the element based on a key.

NOTE: Depending on the operating system, a user interface property name may be available.

**Parameters**

[Object](#) key user interface key (depends on operating system)

**Returns**

[Object](#) The value of the property for specified key.

**Supported Clients**

SmartClient,WebClient,NGClient

**Sample**

```
var property = %%elementName%%.getClientProperty('ToolTipText');
```

**getDataProviderID()**

Get the data provider this UI element (display) is showing.

**Returns**

[String](#) The data provider as String.

**Supported Clients**

SmartClient,WebClient,NGClient

**Sample**

```
%%elementName%%.getDataProviderID();
```

**getDesignProperties()**

Get the design-time properties of an element.

**Returns**

[Object](#)

**Supported Clients**

SmartClient,WebClient,NGClient

**Sample**

```
var prop = forms.orders.elements.mylabel.getDesignProperties()
```

**getDesignTimeProperty(key)**

Get a design-time property of an element.

**Parameters**

[String](#) key the name of the property

**Returns**

[Object](#)

**Supported Clients**

SmartClient,WebClient,NGClient

**Sample**

```
var prop = forms.orders.elements.mylabel.getDesignTimeProperty('myprop')
```

**getElementType()**

Returns the type of a specified element.

---

**Returns**

[String](#) The display type of the element as String.

**Supported Clients**

SmartClient, WebClient, NGClient

**Sample**

```
var et = %%elementName%%.getElementType();
```

**getFormName()**

Returns the name of the form. (may be empty string as well)

**Returns**

[String](#) The name of the form.

**Supported Clients**

SmartClient, WebClient, NGClient

**Sample**

```
var name = %%elementName%%.getFormName();
```

**getHeight()**

Returns the height of the current element.

NOTE: getHeight() can be used with getWidth() to set the size of an element using the setSize function. For example:

```
//returns the width (w) and height (h)
var w = forms.company.elements.faxBtn.getWidth();
var h = forms.company.elements.faxBtn.getHeight();

//sets the new size
forms.company.elements.faxBtn.setSize(w,h);

//sets the new size and adds 1 px to both the width and height
forms.company.elements.faxBtn.setSize(w+1,h+1);
```

**Returns**

[Number](#) The height of the element in pixels.

**Supported Clients**

SmartClient, WebClient, NGClient

**Sample**

```
var ht = %%elementName%%.getHeight();
```

**getLabelForElementNames()**

Returns an Array of label element names that has this field filled in as the labelFor.

**Returns**

[Array](#) An array with element names.

**Supported Clients**

SmartClient, WebClient, NGClient

**Sample**

```
var array = elements.name_first.getLabelForElementNames();
for (var i = 0; i<array.length; i++)
{
    elements[array[i]].fgcolor = "#ff00ff";
}
```

**getLocationX()**

Returns the x location of the current element.

NOTE: getLocationX() can be used with getLocationY() to set the location of an element using the setLocation function. For Example:

```
//returns the X and Y coordinates
var x = forms.company.elements.faxBtn.getLocationX();
var y = forms.company.elements.faxBtn.getLocationY();

//sets the new location 10 px to the right; 10 px down from the current location
forms.company.elements.faxBtn.setLocation(x+10,y+10);
```

#### Returns

[Number](#) The x location of the element in pixels.

#### Supported Clients

SmartClient,WebClient,NGClient

#### Sample

```
var x = %%elementName%%.getLocationX();
```

### getLocationY()

Returns the y location of the current element. The method can only be used in Record view.

NOTE: getLocationY() can be used with getLocationX() to set the location of an element using the setLocation function. For Example:

```
//returns the X and Y coordinates
var x = forms.company.elements.faxBtn.getLocationX();
var y = forms.company.elements.faxBtn.getLocationY();

//sets the new location 10 px to the right; 10 px down from the current location
forms.company.elements.faxBtn.setLocation(x+10,y+10);
```

#### Returns

[Number](#) The y location of the element in pixels.

#### Supported Clients

SmartClient,WebClient,NGClient

#### Sample

```
var y = %%elementName%%.getLocationY();
```

### getName()

Returns the name of an element. (may be null as well)

#### Returns

[String](#) The name of the element.

#### Supported Clients

SmartClient,WebClient,NGClient

#### Sample

```
var name = %%elementName%%.getName();
```

### getScrollX()

Returns the x scroll location of specified element - only for an element where height of element is less than the height of element content.

NOTE: getScrollX() can be used with getScrollY() to set the scroll location of an element using the setScroll function. For Example:

```
//returns the X and Y scroll coordinates
var x = forms.company.elements.portal50.getScrollX();
var y = forms.company.elements.portal50.getScrollY();

//sets the new scroll location
forms.company.elements.portal50.setScroll(x+10,y+10);
```

#### Returns

[Number](#) The x scroll location in pixels.



---

**Supported Clients**

SmartClient,WebClient,NGClient

**Sample**

```
var x = %%elementName%%.getScrollX();
```

**getScrollY()**

Returns the y scroll location of specified element - only for an element where height of element is less than the height of element content.

NOTE: getScrollY() can be used with getScrollX() to set the scroll location of an element using the setScroll function. For Example:

```
//returns the X and Y scroll coordinates
var x = forms.company.elements.portal50.getScrollX();
var y = forms.company.elements.portal50.getScrollY();

//sets the new scroll location
forms.company.elements.portal50.setScroll(x+10,y+10);
```

**Returns**

[Number](#) The y scroll location in pixels.

**Supported Clients**

SmartClient,WebClient,NGClient

**Sample**

```
var y = %%elementName%%.getScrollY();
```

**getSelectedText()**

Returns the currently selected text in the specified text field or text area.

NOTE: This does not work for text fields in the Web Client.

**Returns**

[String](#) The selected text from the component.

**Supported Clients**

SmartClient,WebClient,NGClient

**Sample**

```
var my_text = %%elementName%%.getSelectedText();
```

**getWidth()**

Returns the width of the current element.

NOTE: getWidth() can be used with getHeight() to set the size of an element using the setSize function. For Example:

```
//returns the width (w) and height (h)
var w = forms.company.elements.faxBtn.getWidth();
var h = forms.company.elements.faxBtn.getHeight();

//sets the new size
forms.company.elements.faxBtn.setSize(w,h);

//sets the new size and adds 1 px to both the width and height
forms.company.elements.faxBtn.setSize(w+1,h+1);
```

**Returns**

[Number](#) The width of the element in pixels.

**Supported Clients**

SmartClient,WebClient,NGClient

**Sample**

```
var w = %%elementName%%.getWidth();
```

**hasStyleClass(styleName)**

Check if an element already have a style from the styleClass property.

**Parameters**

[String](#) styleName the name of the style class to be checked

**Supported Clients**

NGClient

**Sample**

```
%%elementName%%.hasStyleClass('redbg');
```

**putClientProperty(key, value)**

Sets the value for the specified element client property key.

NOTE: Depending on the operating system, a user interface property name may be available.

**Parameters**

[Object](#) key user interface key (depends on operating system)

[Object](#) value a predefined value for the key

**Supported Clients**

SmartClient,WebClient,NGClient

**Sample**

```
%%elementName%%.putClientProperty('ToolTipText','some text');
```

**removeStyleClass(styleName)**

Removes a style from the styleClass property. This works only for NGClient where multiple styles are supported.

**Parameters**

[String](#) styleName the name of the style class to remove

**Supported Clients**

NGClient

**Sample**

```
%%elementName%%.removeStyleClass('redbg');
```

**replaceSelectedText(s)**

Replaces the selected text; if no text has been selected, the replaced value will be inserted at the last cursor position.

NOTE: replaceSelectedText applies to text fields and all XXX\_AREA displayType text - RTF\_AREA, HTML\_AREA, or TEXT\_AREA.

**Parameters**

[String](#) s The replacement text.

**Supported Clients**

SmartClient,WebClient,NGClient

**Sample**

```
//returns the current selected text
var my_text = %%elementName%%.getSelectedText();

//replaces the current selected text
%%elementName%%.replaceSelectedText('John');
```

**requestFocus()**

Request the focus in this element. (Focus is also a text cursor on text components).

**Supported Clients**

SmartClient,WebClient,NGClient

**Sample**

```
//request the focus in this %%elementName%% (focus is also a text cursor on text components)
%%elementName%%.requestFocus();
```

**requestFocus(mustExecuteOnFocusGainedMethod)**

Request the focus in this element. (Focus is also a text cursor on text components).

**Parameters**

**Boolean** mustExecuteOnFocusGainedMethod If true will execute onFocusGained method, else will not; default value is true.

**Supported Clients**

SmartClient,WebClient,NGClient

**Sample**

```
//request the focus in this %%elementName%% (focus is also a text cursor on text components), skip on focus
gained method call
%%elementName%%.requestFocus(false);
```

**selectAll()**

Selects all the contents of a field.

**Supported Clients**

SmartClient,WebClient,NGClient

**Sample**

```
%%elementName%%.selectAll();
```

**setLocation(x, y)**

Sets the location of an element. It takes as input the X (horizontal) and Y (vertical) coordinates - starting from the TOP LEFT side of the screen. Please note that location should not be altered at runtime when an element is anchored. Use the solutionModel in such a situation.

NOTE: getLocationX() can be used with getLocationY() to return the current location of an element; then use the X and Y coordinates with the setLocation function to set a new location. For Example:

```
//returns the X and Y coordinates
var x = forms.company.elements.faxBtn.getLocationX();
var y = forms.company.elements.faxBtn.getLocationY();

//sets the new location 10 px to the right; 10 px down from the current location
forms.company.elements.faxBtn.setLocation(x+10,y+10);
```

**Parameters**

**Number** x the X coordinate of the element in pixels.

**Number** y the Y coordinate of the element in pixels.

**Supported Clients**

SmartClient,WebClient,NGClient

**Sample**

```
%%elementName%%.setLocation(200,200);
```

**setScroll(x, y)**

Sets the scroll location of an element. It takes as input the X (horizontal) and Y (vertical) coordinates - starting from the TOP LEFT side of the screen - only for an element where the height of the element is greater than the height of element content

NOTE: getScrollX() can be used with getScrollY() to return the current scroll location of an element; then use the X and Y coordinates with the setScroll function to set a new scroll location. For Example:

```
//returns the X and Y coordinates
var x = forms.company.elements.portal50.getScrollX();
var y = forms.company.elements.portal50.getScrollY();

//sets the new location
forms.company.elements.portal50.setScroll(x+10,y+10);
```

---

**Parameters**

**Number** x the X coordinate of the portal scroll location in pixels

**Number** y the Y coordinate of the portal scroll location in pixels

**Supported Clients**

SmartClient,WebClient,NGClient

**Sample**

```
%%elementName%%.setScroll(200,200);
```

**setSize(width, height)**

Sets the size of an element. It takes as input the width and the height.

Please note that size should not be altered at runtime when an element is anchored. Use the solutionModel in such a situation.

NOTE: getWidth() can be used with getHeight() to set the size of an element using the setSize function. For Example:

```
//returns the width (w) and height (h)
var w = forms.company.elements.faxBtn.getWidth();
var h = forms.company.elements.faxBtn.getHeight();

//sets the new size
forms.company.elements.faxBtn.setSize(w,h);

//sets the new size and adds 1 px to both the width and height
forms.company.elements.faxBtn.setSize(w+1,h+1);
```

**Parameters**

**Number** width the width of the element in pixels.

**Number** height the height of the element in pixels.

**Supported Clients**

SmartClient,WebClient,NGClient

**Sample**

```
%%elementName%%.setSize(20,30);
```