Menu

Property Summery

String #text

Method Summery

CheckBox	#addCheckBox([name], [method], [icon], [mnemonic], [enabled], [align]) Add the Checkbox at the selected index (starting at 0) or add it at the end (empty).
Menu	#addMenu([name], [menu], [icon], [mnemonic], [enabled], [align]) Add the submenu at the selected index (starting at 0) or add it at the end (empty).
MenuItem	#addMenuItem([name], [method], [icon], [mnemonic], [enabled], [align]) Add the item at the selected index (starting at 0) or add it at the end (empty).
RadioButton	#addRadioButton([name], [method], [icon], [mnemonic], [enabled], [align]) Add the Radiobutton at the selected index (starting at 0) or add it at the end (empty).
void	#addRadioGroup() Add a Radiogroup for the Radiobuttons.
void	#addSeparator() Add the separator at the selected index (starting at 0) or add it at the end (empty).
void	#doClick(click) Script the selection (emulate a mouse click) of the menu.
CheckBox	#getCheckBox(index) Get the Checkbox at the selected index (starting at 0).
MenuItem	#getItem(index) Get the item at the selected index (starting at 0).
Number	#getItemCount() Get the number of items in the menu.
Number	#getItemIndexByText(name) Retrieve the index of the item by text.
Menu	#getMenu(index) Get the submenu at the selected index (starting at 0).
RadioButton	#getRadioButton(index) Get the Radiobutton at the selected index (starting at 0).
void	#removeAllItems() Remove all items from the menu.
void	#removeItem(index 1, [index 2-n]) Remove the item(s) at the selected index/indices.
void	#setEnabled(enabled) Set the the selected menu enabled or disabled.
void	#setIcon(icon) Set the icon of the menu.
void	#setMnemonic(mnemonic) Set the mnemonic of the selected menu.

Property Details

text

Replace with description

Returns

String

Method Details

addCheckBox

CheckBox **addCheckBox**([name], [method], [icon], [mnemonic], [enabled], [align])

Add the Checkbox at the selected index (starting at 0) or add it at the end (empty).

Parameters

[name]
[method]
[icon]
[mnemonic]
[enabled]
[align]

Returns

CheckBox

Sample

```
// Add the Checkbox at the selected index (starting at 0) or add it at the end (empty).
// get the menu at the last index
// indexes start at 0 (zero) so index 2 is in fact position 3
var menu = plugins.window.getMenu(plugins.window.getMenuCount() - 1);

// when you don't define an index the checkbox will be added at the last position
// this is what you usually do to build a new menu
// minimum settings are the text and method properties
// the method can be a global or form method
// be sure to enter the method WITHOUT '()' at the end
var checkbox = menu.addCheckBox("checkbox with feedback",feedback_checkbox);

var checkbox = menu.addCheckBox("checkbox selected",feedback_checkbox);
// set the checkbox to selected
checkbox.setSelected(true);

var checkbox = menu.addCheckBox("checkbox with input");

// add an 'input' array. the array will be concatenated to the end of the arguments
// array which can be read out in the selected method
var input = [1,"is","the","added","input",false];

checkbox.setMethod(feedback_checkbox, input);

// create a checkbox with an icon
var checkbox = menu.addCheckBox("checkbox with icon",feedback_checkbox,"media:///yourimage.gif");

var checkbox = menu.addCheckBox("checkbox with accelerator",feedback_checkbox,"media:///yourimage.gif");
// add an accelerator key ('alt shift a' in the below example)
// REMARK: always test the accelerator key. sometimes they will not work because
// these keys already have an 'action' assigned to them via the operating system.
checkbox.setAccelerator("alt shift a");

var checkbox = menu.addCheckBox("checkbox with mnemonic",feedback_checkbox,false,input,"media:///yourimage.
gif");
// add a mnemonic key  ('i' in our example) which is the underlined shortkey on windows
// REMARK: setting the mnemonic key is platform dependent
checkbox.setMnemonic("i");

var checkbox = menu.addCheckBox("checkbox disabled",feedback_checkbox);
// disable the menu item
checkbox.setEnabled(false);

var checkbox = menu.addCheckBox("checkbox invisible",feedback_checkbox);
// set the menu item disabled and NOT visible
checkbox.setVisible(false);

// add a separator at the last position or at a given index
menu.addSeparator();
```

addMenu

Menu **addMenu**([name], [menu], [icon], [mnemonic], [enabled], [align])

Add the submenu at the selected index (starting at 0) or add it at the end (empty).

Parameters

[name]
[menu]
[icon]
[mnemonic]
[enabled]
[align]

Returns

Menu

Sample

```
// Add the submenu at the selected index (starting at 0) or add it at the end (empty).
// get the menu at the last index
// indexes start at 0 (zero) so index 2 is in fact position 3
var menu = plugins.window.getMenu(plugins.window.getMenuCount() - 1);

// add a (first) submenu
var submenu1 = menu.addMenu("submenu 1");
submenu1.addMenuItem("sub item 1",globals.feedback_item);

// add a (second) submenu
var submenu2 = submenu1.addMenu("submenu 2");
submenu2.addMenuItem("sub item 2",globals.feedback_item);

// add a (third) submenu
var submenu3 = submenu1.addMenu("submenu 3");
submenu3.addMenuItem("sub item 3",globals.feedback_item);

// add a (first) submenu to the (third) submenu
var submenu4 = submenu3.addMenu("submenu 4");
submenu4.addMenuItem("sub item 4",globals.feedback_item);

// add a (first) submenu to the (first) submenu of the (third) submenu
var submenu5 = submenu4.addMenu("submenu 5");
submenu5.addMenuItem("sub item 5",globals.feedback_item);
```

addMenuItem

MenuItem **addMenuItem**([name], [method], [icon], [mnemonic], [enabled], [align])

Add the item at the selected index (starting at 0) or add it at the end (empty).

Parameters

[name]
[method]
[icon]
[mnemonic]
[enabled]
[align]

Returns

MenuItem

Sample

```
// Add the item at the selected index (starting at 0) or add it at the end (empty).
// get the menu at the last index
// indexes start at 0 (zero) so index 2 is in fact position 3
var menu = plugins.window.getMenu(plugins.window.getMenuCount() - 1);

// when you don't define an index the item will be added at the last position
// this is what you usually do to build a new menu
// create the settings for the specified menu item
// minimum settings are the text and method properties
// the method can be a global or form method
// be sure to enter the method WITHOUT '()' at the end
var item = menu.addMenuItem("item with feedback",globals.feedback_item);

var item = menu.addMenuItem();

// add an 'input' array. the array will be concatenated to the end of the arguments
// array which can be read out in the selected method
var input = [1,"is","the","added","input",false];

item.text = "item with input";
item.setMethod(globals.feedback_item,input);

var item = menu.addMenuItem();
// add an icon to the item
item.text = "item with icon";
item.setMethod(globals.feedback_item, input);
item.setIcon("media:///yourimage.gif");

var item = menu.addMenuItem();
// add an accelerator key ('alt shift 2' in the below example)
// REMARK: always test the accelerator key. sometimes they will not work because
// these keys already have an 'action' assigned to them via the operating system.
item.text = "item with accelerator";
item.setMethod(globals.feedback_item, input);
item.setIcon("media:///yourimage.gif");
item.setAccelerator("alt shift 2");

var item = menu.addMenuItem();
// add a mnemonic key  ('i' in our example) which is the underlined shortkey on windows
// REMARK: setting the mnemonic key is platform dependent
// the accelerator key will not work in this and the next example
item.text = "item with mnemonic";
item.setMethod(globals.feedback_item, input);
item.setIcon("media:///yourimage.gif");
item.setAccelerator("pressed COMMA");
item.setMnemonic("i");

// create a disabled menu item
var item = menu.addMenuItem("item disabled",globals.feedback_item,"media:///yourimage.gif","t",false);
// set the method args
item.setMethodArguments(input);

var item = menu.addMenuItem("item visible",globals.feedback_item,"media:///yourimage.gif","e");
// this accelerator key will work
item.setAccelerator("shift meta PAGE_DOWN");

var item = menu.addMenuItem("item invisible",globals.feedback_item,"media:///yourimage.gif");
// now the item is enabled and NOT visible
item.setVisible(false);

// add a separator at the last position or at a given index
menu.addSeparator();
return;
```

addRadioButton

RadioButton **addRadioButton**([name], [method], [icon], [mnemonic], [enabled], [align])

Add the Radiobutton at the selected index (starting at 0) or add it at the end (empty).

Parameters
[name]
[method]
[icon]
[mnemonic]
[enabled]
[align]

Returns
RadioButton

Sample

```
// Add the Radiobutton at the selected index (starting at 0) or add it at the end (empty).
// get the menu at the last index
// indexes start at 0 (zero) so index 2 is in fact position 3
var menu = plugins.window.getMenu(plugins.window.getMenuCount() - 1);

// add a new Radiobutton group
// a group will 'bind' all added radiobuttons after the group together
// as a result checking one item will uncheck the other
menu.addRadioGroup();

// when you don't define an index the radiobutton will be added at the last position
// this is what you usually do to build a new menu

// create the settings for the specified menu item
// minimum settings are the text and method properties
// the method can be a global or form method
// be sure to enter the method WITHOUT '()' at the end
var radiobutton = menu.addRadioButton("radiobutton with feedback",feedback_radiobutton);

var radiobutton = menu.addRadioButton("radiobutton selected",feedback_radiobutton);
// set the radiobutton to selected
radiobutton.setSelected(true);

var radiobutton = menu.addRadioButton("radiobutton with input");

// add an 'input' array. the array will be concatenated to the end of the arguments
// array which can be read out in the selected method
var input = [1,"is","the","added","input",false];

radiobutton.setMethod(feedback_radiobutton,input);

// create an item with an icon
var radiobutton = menu.addRadioButton("radiobutton with icon",feedback_radiobutton,"media:///yourimage.gif");

var radiobutton = menu.addRadioButton("radiobutton with accelerator",feedback_radiobutton);
// add an accelerator key ('alt shift 3' in the below example)
// REMARK: always test the accelerator key. sometimes they will not work because
// these keys already have an 'action' assigned to them via the operating system.
radiobutton.setAccelerator("alt shift 3");

// add a separator at the last position or at a given index
menu.addSeparator();

// add a new Radiobutton group
menu.addRadioGroup();

// add a mnemonic key  ('i' in our example) which is the underlined shortkey on windows
// REMARK: setting the mnemonic key is platform dependent
var radiobutton = menu.addRadioButton("radiobutton with mnemonic",feedback_radiobutton,"media:///yourimage.
gif","i");

var radiobutton = menu.addRadioButton("radiobutton disabled",feedback_radiobutton);
// disable the menu item
radiobutton.setEnabled(false);

var radiobutton = menu.addRadioButton("radiobutton invisible",feedback_radiobutton);
// now the item is enabled and NOT visible
radiobutton.setVisible(false);

// add a separator at the last position or at a given index
menu.addSeparator();
```

addRadioGroup

void **addRadioGroup**()

Add a Radiogroup for the Radiobuttons.

Returns

void

Sample

```
// Add a Radiogroup for the Radiobuttons.
// get the menu at the last index
// indexes start at 0 (zero) so index 2 is in fact position 3
var menu = plugins.window.getMenu(plugins.window.getMenuCount() - 1);

// add a new Radiobutton group
// a group will 'bind' all added radiobuttons after the group together
// as a result checking one item will uncheck the other
menu.addRadioGroup();

// when you don't define an index the radiobutton will be added at the last position
// this is what you usually do to build a new menu

// create the settings for the specified menu item
// minimum settings are the text and method properties
// the method can be a global or form method
// be sure to enter the method WITHOUT '()' at the end
var radiobutton = menu.addRadioButton("radiobutton with feedback",feedback_radiobutton);

var radiobutton = menu.addRadioButton("radiobutton selected",feedback_radiobutton);
// set the radiobutton to selected
radiobutton.setSelected(true);

var radiobutton = menu.addRadioButton("radiobutton with input");

// add an 'input' array. the array will be concatenated to the end of the arguments
// array which can be read out in the selected method
var input = [1,"is","the","added","input",false];

radiobutton.setMethod(feedback_radiobutton,input);

// create an item with an icon
var radiobutton = menu.addRadioButton("radiobutton with icon",feedback_radiobutton,"media:///yourimage.gif");

var radiobutton = menu.addRadioButton("radiobutton with accelerator",feedback_radiobutton);
// add an accelerator key ('alt shift 3' in the below example)
// REMARK: always test the accelerator key. sometimes they will not work because
// these keys already have an 'action' assigned to them via the operating system.
radiobutton.setAccelerator("alt shift 3");

// add a separator at the last position or at a given index
menu.addSeparator();

// add a new Radiobutton group
menu.addRadioGroup();

// add a mnemonic key  ('i' in our example) which is the underlined shortkey on windows
// REMARK: setting the mnemonic key is platform dependent
var radiobutton = menu.addRadioButton("radiobutton with mnemonic",feedback_radiobutton,"media:///yourimage.
gif","i");

var radiobutton = menu.addRadioButton("radiobutton disabled",feedback_radiobutton);
// disable the menu item
radiobutton.setEnabled(false);

var radiobutton = menu.addRadioButton("radiobutton invisible",feedback_radiobutton);
// now the item is enabled and NOT visible
radiobutton.setVisible(false);

// add a separator at the last position or at a given index
menu.addSeparator();
```

addSeparator

void **addSeparator**()

Add the separator at the selected index (starting at 0) or add it at the end (empty).

Returns

void

Sample

```
// Add the separator at the selected index (starting at 0) or add it at the end (empty).
plugins.window.getMenu(0).addSeparator();
```

doClick

void **doClick**(click)

Script the selection (emulate a mouse click) of the menu.

Parameters

click

Returns

void

Sample

```
// Script the selection (emulate a mouse click) of the menu.
plugins.window.getMenu(0).doClick();
```

getCheckBox

CheckBox **getCheckBox**(index)

Get the Checkbox at the selected index (starting at 0).

Parameters

index

Returns

CheckBox

Sample

```
// Get the Checkbox at the selected index (starting at 0).
// get the menu at the last position
// indexes start at 0 (zero) so index 2 is in fact position 3
var menu = plugins.window.getMenu(plugins.window.getMenuCount() - 1);

var checkbox = menu.getCheckBox(0);

checkbox.setText("Changed menu item");

// REMARK: we actually changed an original menu (item)! As a result resetting the
// menubar will NOT reset the above changes. We need to reset the menu (item)
// manually the following way:

// get the menu
// var menu = plugins.window.getMenu(2);

// get the item
// var item = menu.getItem(0);

// reset the values to default
// notice we use an i18n message here the same way you would use it with
// standard Servoy methods and plugins
// item.setText("i18n:servoy.menuitem.viewAsRecord");
```

getItem

MenuItem **getItem**(index)

Get the item at the selected index (starting at 0).

Parameters

index

Returns

MenuItem

Sample

```
// Get the item at the selected index (starting at 0).
// get the menu at the last position
// indexes start at 0 (zero) so index 2 is in fact position 3
var menu = plugins.window.getMenu(plugins.window.getMenuCount() - 1);

var item = menu.getItem(0);

item.setText("Changed menu item");

// REMARK: we actually changed an original menu (item)! As a result resetting the
// menubar will NOT reset the above changes. We need to reset the menu (item)
// manually the following way:

// get the menu
// var menu = plugins.window.getMenu(2);

// get the item
// var item = menu.getItem(0);

// reset the values to default
// notice we use an i18n message here the same way you would use it with
// standard Servoy methods and plugins
// item.setText("i18n:servoy.menuitem.viewAsRecord");
```

getItemCount

Number **getItemCount**()

Get the number of items in the menu.

Returns

Number

Sample

```
// Get the number of items in the menu.
// REMARK: indexes start at 0, disabled items, non visible items and seperators are counted also
// REMARK: this is especially important when getting items by the index
application.output(plugins.window.getMenu(0).getItemCount());
```

getItemIndexByText

Number **getItemIndexByText**(name)

Retrieve the index of the item by text.

Parameters

name

Returns

Number

Sample

getMenu

Menu **getMenu**(index)

Get the submenu at the selected index (starting at 0).

Parameters

index

Returns

Menu

Sample

```
// Get the submenu at the selected index (starting at 0).
// get the menu at the last position
// indexes start at 0 (zero) so index 2 is in fact position 3
var menu = plugins.window.getMenu(plugins.window.getMenuCount() - 1);

var checkbox = menu.getMenu(0);

checkbox.setText("Changed menu item");
```

getRadioButton

RadioButton **getRadioButton**(index)

Get the Radiobutton at the selected index (starting at 0).

Parameters

index

Returns

RadioButton

Sample

```
// Get the Radiobutton at the selected index (starting at 0).
// get the menu at the last position
// indexes start at 0 (zero) so index 2 is in fact position 3
var menu = plugins.window.getMenu(plugins.window.getMenuCount() - 1);

var checkbox = menu.getItem(0);

checkbox.setText("Changed menu item");

// REMARK: we actually changed an original menu (item)! As a result resetting the
// menubar will NOT reset the above changes. We need to reset the menu (item)
// manually the following way:

// get the menu
// var menu = plugins.window.getMenu(2);

// get the item
// var item = menu.getItem(0);

// reset the values to default
// notice we use an i18n message here the same way you would use it with
// standard Servoy methods and plugins
// item.setText("i18n:servoy.menuitem.viewAsRecord");
```

removeAllItems

void **removeAllItems**()

Remove all items from the menu.

Returns

void

Sample

```
// Remove all items from the menu.
// get the menu at the last index
// indexes start at 0 (zero) so index 2 is in fact position 3
var menu = plugins.window.getMenu(plugins.window.getMenuCount() - 1);

// remove all menu items from the selected menu
menu.removeAllItems();
```

removeItem

void **removeItem**(index 1, [index 2-n])

Remove the item(s) at the selected index/indices.

Parameters

index 1

[index 2-n]

Returns
void
Sample

```
// Remove the item(s) at the selected index/indices.
// get the menu at the last index
// indexes start at 0 (zero) so index 2 is in fact position 3
var menu = plugins.window.getMenu(plugins.window.getMenuCount() - 1);

// remove only one item at the selected index
// from the selected menu
// menu.removeItem(0);

// remove more than one item at the selected indices
// from the selected menu
menu.removeItem(1,2);
```

setEnabled

void **setEnabled**(enabled)
Set the the selected menu enabled or disabled.
Parameters
enabled
Returns
void
Sample

```
// Set the the selected menu enabled or disabled.
var menu = plugins.window.getMenu(0);
menu.setText("Hello");
menu.setMnemonic("H");
menu.setEnabled(false);
```

setIcon

void **setIcon**(icon)
Set the icon of the menu.
Parameters
icon
Returns
void
Sample

```

```

setMnemonic

void **setMnemonic**(mnemonic)
Set the mnemonic of the selected menu.
Parameters
mnemonic
Returns
void
Sample

```
// Set the mnemonic of the selected menu.
var menu = plugins.window.getMenu(0);
menu.setText("Hello");
menu.setMnemonic("H");
menu.setEnabled(false);
```