

# MenuBar

## Method Summary

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| <a href="#">Menu</a>   | <a href="#">addMenu()</a><br>Add a menu to the menubar.                                |
| <a href="#">Menu</a>   | <a href="#">addMenu(index)</a><br>Add a menu to the menubar.                           |
| <a href="#">Menu</a>   | <a href="#">getMenu(index)</a><br>Get the menu at the selected index (starting at 0).  |
| <a href="#">Number</a> | <a href="#">getMenuCount()</a><br>Get the number of (top level) menu's.                |
| <a href="#">Number</a> | <a href="#">getMenuIndexByText(name)</a><br>Retrieve the index of the item by text.    |
| <a href="#">void</a>   | <a href="#">removeAllMenus()</a><br>Remove all menus from the menubar.                 |
| <a href="#">void</a>   | <a href="#">removeMenu(index)</a><br>Remove the menu(s) at the selected index/indices. |
| <a href="#">void</a>   | <a href="#">reset()</a><br>Reset the menubar to the default.                           |
| <a href="#">void</a>   | <a href="#">setVisible(visible)</a><br>Show/hide the menu bar                          |

## Method Details

### addMenu

[Menu](#) [addMenu\(\)](#)  
Add a menu to the menubar.

#### Returns

[Menu](#)

#### Sample

```
// Note: method addMenu only works in the smart client.

// when you don't define an index the menu will be added at the last
// position of the menubar
var menubar = plugins.window.getMenuBar();
var menu = menubar.addMenu();
// set the text of the menu
menu.text = "add menu";
// set the mnemonic key
menu.setMnemonic("a");
// add another menu at a specific position in the menubar
var another_menu = menubar.addMenu(2);
another_menu.text = "another menu";
another_menu.setMnemonic("t")
// REMARK: normally you would add menu items, checkboxes etc in the same method
// this example will show no menu items for now!
// IMPORTANT: Working with menu's on developer and client can differ
```

### addMenu

[Menu](#) [addMenu\(index\)](#)  
Add a menu to the menubar.

#### Parameters

[{Number}](#) index

#### Returns

[Menu](#)

**Sample**

```
// Note: method addMenu only works in the smart client.

// when you don't define an index the menu will be added at the last
// position of the menubar
var menubar = plugins.window.getMenuBar();
var menu = menubar.addMenu();
// set the text of the menu
menu.text = "add menu";
// set the mnemonic key
menu.setMnemonic("a");
// add another menu at a specific position in the menubar
var another_menu = menubar.addMenu(2);
another_menu.text = "another menu";
another_menu.setMnemonic("t")
// REMARK: normally you would add menu items, checkboxes etc in the same method
// this example will show no menu items for now!
// IMPORTANT: Working with menu's on developer and client can differ
```

**getMenu**[Menu](#) **getMenu** (index)

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

**Parameters**[{Number}](#) index**Returns**[Menu](#)**Sample**

```
// Note: method getMenu only works in the smart client.

var menubar = plugins.window.getMenuBar();
// get the menu at index 2
// indexes start at 0 (zero) so index 2 is in fact position 3
var menu = menubar.getMenu(2);
// set the text of the menu at the chose position
menu.text = "get menu";
// set the mnemonic key
menu.setMnemonic("g");
// disable the menu
menu.setEnabled(false);
// REMARK: we actually changed an original menu! As a result resetting the
// menubar will NOT reset the above changes. We need to reset the menu
// manually the following way:
// get the menu
//var menu = menubar.getMenu(2);
// 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
//menu.text = "i18n:servoy.menuitem.showAll";
//menu.setEnabled(true);
```

**getMenuCount**[Number](#) **getMenuCount** ()

Get the number of (top level) menu's.

**Returns**[Number](#)

---

**Sample**

```
// Note: method getMenuCount only works in the smart client.

// add a new menu before the last menu
var menubar = plugins.window.getMenuBar();
var count = menubar.getMenuCount();
var menu = menubar.addMenu(count-1);
menu.text = 'new menu';
```

**getMenuIndexByText**

**Number** **getMenuIndexByText** (name)

Retrieve the index of the item by text.

**Parameters**

{[String](#)} name

**Returns**

[Number](#)

**Sample**

```
// Note: method getMenuIndexByText only works in the smart client.

var menubar = plugins.window.getMenuBar();
// find the index of the View menu
var idx = menubar.getMenuIndexByText("View");
// add a menu before the View menu
var menu = menubar.addMenu(idx);
menu.text = "new menu";
```

**removeAllMenus**

void **removeAllMenus** ()

Remove all menus from the menubar.

**Returns**

void

**Sample**

```
// Note: method removeAllMenus only works in the smart client.

// Potentially dangerous because all accelerator (short) keys
// will be deleted also (including the quit item)
var menubar = plugins.window.getMenuBar();
menubar.removeAllMenus();
```

**removeMenu**

void **removeMenu** (index)

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

**Parameters**

{[Object](#)[]} index - array of one or more indexes corresponding to menus to remove

**Returns**

void

**Sample**

```
// Note: method removeMenu only works in the smart client.

var menubar = plugins.window.getMenuBar();
// To remove the last menu in the menubar we count the number of menu's in the menubar
// because the index starts at 0 we have to subtract 1 from the counted menu's
// to actually remove the last menu from the menubar
var index = menubar.getMenuCount() - 1;
menubar.removeMenu(index);

// To remove the last 3 (three) menu's from the menubar we
// can do that by adding additional indexes to the method
// and delimit them with a comma.
index = menubar.getMenuCount() - 1;
menubar.removeMenu(index, index-1, index-2);

// For 'security' reasons it is best to ALWAYS remove the menu with the last index
// first to avoid index out of range issues and other issues
// EXAMPLE: when you first remove the menu at index 2 and then the menu at index 4
// you actually remove the menu at index 2 and index 5
// after removing the menu at index 2 all other menu's moved one index to the left
// so the menu at index 4 moved to index 3 and the menu at index 5 moved to index 4 etc.
```

**reset**

void **reset** ()

Reset the menubar to the default.

**Returns**

void

**Sample**

```
// Note: method removeMenu only works in the smart client.

// When the menubar settings are solution specific it is advised to reset
// the bar to its default settings when closing the solution.
// Another reason is that when a client/developer is started first the
// plugin will save the current settings in memory.
// REMARK: Don't manipulate standard Servoy menuitems but remove
// them and create new ones! Due to the way menuitems are managed by java it is not
// possible to reset a menuitem anymore.
var menubar = plugins.window.getMenuBar();
// add a menu
var menu = menubar.addMenu();
menu.text = "new menu";
// reset the menubar, the newly added menu will dissappear
menubar.reset();
```

**setVisible**

void **setVisible** (visible)

Show/hide the menu bar

**Parameters**

{Boolean} visible

**Returns**

void

**Sample**

```
// Note: method setVisible only works in the smart client.

// hide the menu bar
var menubar = plugins.window.getMenuBar();
menubar.setVisible(false);
```

