

ToolBar

Method Summary

void	addButton (text, method) Add a Button to the toolbar.
void	addButton (text, method, arguments) Add a Button to the toolbar.
void	addButton (text, method, arguments, icon) Add a Button to the toolbar.
void	addButton (text, method, arguments, icon, tooltip) Add a Button to the toolbar.
void	addButton (text, method, arguments, icon, tooltip, enabled) Add a Button to the toolbar.
void	addButton (text, method, arguments, icon, tooltip, enabled, visible) Add a Button to the toolbar.
void	addCheckBox (text, method) Add a CheckBox to the toolbar.
void	addCheckBox (text, method, selected) Add a CheckBox to the toolbar.
void	addCheckBox (text, method, selected, tooltip) Add a CheckBox to the toolbar.
void	addCheckBox (text, method, selected, tooltip, enabled) Add a CheckBox to the toolbar.
void	addCheckBox (text, method, selected, tooltip, enabled, visible) Add a CheckBox to the toolbar.
void	addComboBox (method, index, values) Add a ComboBox to the toolbar.
void	addComboBox (method, index, values, tooltip) Add a ComboBox to the toolbar.
void	addComboBox (method, index, values, tooltip, enabled) Add a ComboBox to the toolbar.
void	addComboBox (method, index, values, tooltip, enabled, visible) Add a ComboBox to the toolbar.
void	addField (method, text) Add a Field to the toolbar.
void	addField (method, text, length) Add a Field to the toolbar.
void	addField (method, text, length, tooltip) Add a Field to the toolbar.
void	addField (method, text, length, tooltip, enabled) Add a Field to the toolbar.
void	addField (method, text, length, tooltip, enabled, visible) Add a Field to the toolbar.
void	addSeparator () Add a Separator to the toolbar.
void	enableItem (index, enabled) Enable/disable the item at the specified index.
void	removeAllItems () Remove all Buttons, Checkboxes etc.
void	removeItem (index) Remove a Button, CheckBox, ComboBox from the toolbar.
void	selectCheckBox (index, selected) Set the CheckBox selection.
void	selectComboBox (index, selection) Select a row of the ComboBox via the index.
void	setFieldText (index, text) Set a (default) text of the field at the given index.
void	visibleItem (index, visible) Make the item at the specified index visible/invisible.

Method Details

addButton

void **addButton** (text, method)
Add a Button to the toolbar.

Parameters

`{String}` text
`{Function}` method

Returns

void

Sample

```
// create a new toolbar
var toolbar = plugins.window.addToolBar('toolbar_0');
// add a button with a text and a method
toolbar.addButton("button", feedback_button);
// add an input array to the button for feedback in the selected method
toolbar.addButton("button", feedback_button, [1, "2", "three"]);
// add an icon to the button
toolbar.addButton("button", feedback_button, [1, "2", "three"], "media:///yourimage.gif");
// add a tooltip to the button
toolbar.addButton("button", feedback_button, [1, "2", "three"], "media:///yourimage.gif", "tooltip");
// show only an icon on the button and disable the button
toolbar.addButton(null, feedback_button, [1, "2", "three"], "media:///yourimage.gif", "tooltip", false);
// make the button non visible
toolbar.addButton(null, feedback_button, [1, "2", "three"], "media:///yourimage.gif", "tooltip", true, false);
```

addButton

void **addButton** (text, method, arguments)

Add a Button to the toolbar.

Parameters

`{String}` text
`{Function}` method
`{Object[]}` arguments

Returns

void

Sample

```
// create a new toolbar
var toolbar = plugins.window.addToolBar('toolbar_0');
// add a button with a text and a method
toolbar.addButton("button", feedback_button);
// add an input array to the button for feedback in the selected method
toolbar.addButton("button", feedback_button, [1, "2", "three"]);
// add an icon to the button
toolbar.addButton("button", feedback_button, [1, "2", "three"], "media:///yourimage.gif");
// add a tooltip to the button
toolbar.addButton("button", feedback_button, [1, "2", "three"], "media:///yourimage.gif", "tooltip");
// show only an icon on the button and disable the button
toolbar.addButton(null, feedback_button, [1, "2", "three"], "media:///yourimage.gif", "tooltip", false);
// make the button non visible
toolbar.addButton(null, feedback_button, [1, "2", "three"], "media:///yourimage.gif", "tooltip", true, false);
```

addButton

void **addButton** (text, method, arguments, icon)

Add a Button to the toolbar.

Parameters

`{String}` text
`{Function}` method
`{Object[]}` arguments
`{Object}` icon

Returns

void

Sample

```
// create a new toolbar
var toolbar = plugins.window.addToolBar('toolbar_0');
// add a button with a text and a method
toolbar.addButton("button", feedback_button);
// add an input array to the button for feedback in the selected method
toolbar.addButton("button", feedback_button, [1, "2", "three"]);
// add an icon to the button
toolbar.addButton("button", feedback_button, [1, "2", "three"], "media:///yourimage.gif");
// add a tooltip to the button
toolbar.addButton("button", feedback_button, [1, "2", "three"], "media:///yourimage.gif", "tooltip");
// show only an icon on the button and disable the button
toolbar.addButton(null, feedback_button, [1, "2", "three"], "media:///yourimage.gif", "tooltip", false);
// make the button non visible
toolbar.addButton(null, feedback_button, [1, "2", "three"], "media:///yourimage.gif", "tooltip", true, false);
```

addButton

void **addButton** (text, method, arguments, icon, tooltip)

Add a Button to the toolbar.

Parameters

```
{String} text
{Function} method
{Object[]} arguments
{Object} icon
{String} tooltip
```

Returns

void

Sample

```
// create a new toolbar
var toolbar = plugins.window.addToolBar('toolbar_0');
// add a button with a text and a method
toolbar.addButton("button", feedback_button);
// add an input array to the button for feedback in the selected method
toolbar.addButton("button", feedback_button, [1, "2", "three"]);
// add an icon to the button
toolbar.addButton("button", feedback_button, [1, "2", "three"], "media:///yourimage.gif");
// add a tooltip to the button
toolbar.addButton("button", feedback_button, [1, "2", "three"], "media:///yourimage.gif", "tooltip");
// show only an icon on the button and disable the button
toolbar.addButton(null, feedback_button, [1, "2", "three"], "media:///yourimage.gif", "tooltip", false);
// make the button non visible
toolbar.addButton(null, feedback_button, [1, "2", "three"], "media:///yourimage.gif", "tooltip", true, false);
```

addButton

void **addButton** (text, method, arguments, icon, tooltip, enabled)

Add a Button to the toolbar.

Parameters

```
{String} text
{Function} method
{Object[]} arguments
{Object} icon
{String} tooltip
{Boolean} enabled
```

Returns

void

Sample

```
// create a new toolbar
var toolbar = plugins.window.addToolBar('toolbar_0');
// add a button with a text and a method
toolbar.addButton("button", feedback_button);
// add an input array to the button for feedback in the selected method
toolbar.addButton("button", feedback_button, [1, "2", "three"]);
// add an icon to the button
toolbar.addButton("button", feedback_button, [1, "2", "three"], "media:///yourimage.gif");
// add a tooltip to the button
toolbar.addButton("button", feedback_button, [1, "2", "three"], "media:///yourimage.gif", "tooltip");
// show only an icon on the button and disable the button
toolbar.addButton(null, feedback_button, [1, "2", "three"], "media:///yourimage.gif", "tooltip", false);
// make the button non visible
toolbar.addButton(null, feedback_button, [1, "2", "three"], "media:///yourimage.gif", "tooltip", true, false);
```

addButton

void **addButton** (text, method, arguments, icon, tooltip, enabled, visible)

Add a Button to the toolbar.

Parameters

```
{String} text
{Function} method
{Object[]} arguments
{Object} icon
{String} tooltip
{Boolean} enabled
{Boolean} visible
```

Returns

void

Sample

```
// create a new toolbar
var toolbar = plugins.window.addToolBar('toolbar_0');
// add a button with a text and a method
toolbar.addButton("button", feedback_button);
// add an input array to the button for feedback in the selected method
toolbar.addButton("button", feedback_button, [1, "2", "three"]);
// add an icon to the button
toolbar.addButton("button", feedback_button, [1, "2", "three"], "media:///yourimage.gif");
// add a tooltip to the button
toolbar.addButton("button", feedback_button, [1, "2", "three"], "media:///yourimage.gif", "tooltip");
// show only an icon on the button and disable the button
toolbar.addButton(null, feedback_button, [1, "2", "three"], "media:///yourimage.gif", "tooltip", false);
// make the button non visible
toolbar.addButton(null, feedback_button, [1, "2", "three"], "media:///yourimage.gif", "tooltip", true, false);
```

addCheckBox

void **addCheckBox** (text, method)

Add a CheckBox to the toolbar.

Parameters

```
{String} text
{Function} method
```

Returns

void

Sample

```
// add a toolbar
var toolbar = plugins.window.addToolBar("toolbar_1");
// add a checkbox with a text and a method
toolbar.addCheckBox("checkbox", feedback_checkbox);
// add an checkbox and set it's state to selected (not selected by default)
toolbar.addCheckBox("checkbox", feedback_checkbox, true);
// add a tooltip to the checkbox
toolbar.addCheckBox("checkbox", feedback_checkbox, false, "tooltip");
// disable the checkbox and select it
toolbar.addCheckBox("checkbox", feedback_checkbox, true, "tooltip", false);
// make the checkbox non visible
toolbar.addCheckBox("checkbox", feedback_checkbox, false, "tooltip", false, false);
```

addCheckBox

void **addCheckBox** (text, method, selected)

Add a CheckBox to the toolbar.

Parameters

{String} text
{Function} method
{Boolean} selected

Returns

void

Sample

```
// add a toolbar
var toolbar = plugins.window.addToolBar("toolbar_1");
// add a checkbox with a text and a method
toolbar.addCheckBox("checkbox", feedback_checkbox);
// add an checkbox and set it's state to selected (not selected by default)
toolbar.addCheckBox("checkbox", feedback_checkbox, true);
// add a tooltip to the checkbox
toolbar.addCheckBox("checkbox", feedback_checkbox, false, "tooltip");
// disable the checkbox and select it
toolbar.addCheckBox("checkbox", feedback_checkbox, true, "tooltip", false);
// make the checkbox non visible
toolbar.addCheckBox("checkbox", feedback_checkbox, false, "tooltip", false, false);
```

addCheckBox

void **addCheckBox** (text, method, selected, tooltip)

Add a CheckBox to the toolbar.

Parameters

{String} text
{Function} method
{Boolean} selected
{String} tooltip

Returns

void

Sample

```
// add a toolbar
var toolbar = plugins.window.addToolBar("toolbar_1");
// add a checkbox with a text and a method
toolbar.addCheckBox("checkbox", feedback_checkbox);
// add an checkbox and set it's state to selected (not selected by default)
toolbar.addCheckBox("checkbox", feedback_checkbox, true);
// add a tooltip to the checkbox
toolbar.addCheckBox("checkbox", feedback_checkbox, false, "tooltip");
// disable the checkbox and select it
toolbar.addCheckBox("checkbox", feedback_checkbox, true, "tooltip", false);
// make the checkbox non visible
toolbar.addCheckBox("checkbox", feedback_checkbox, false, "tooltip", false, false);
```

addCheckBox

void **addCheckBox** (text, method, selected, tooltip, enabled)

Add a CheckBox to the toolbar.

Parameters

`{String}` text
`{Function}` method
`{Boolean}` selected
`{String}` tooltip
`{Boolean}` enabled

Returns

void

Sample

```
// add a toolbar
var toolbar = plugins.window.addToolBar("toolbar_1");
// add a checkbox with a text and a method
toolbar.addCheckBox("checkbox", feedback_checkbox);
// add an checkbox and set it's state to selected (not selected by default)
toolbar.addCheckBox("checkbox", feedback_checkbox, true);
// add a tooltip to the checkbox
toolbar.addCheckBox("checkbox", feedback_checkbox, false, "tooltip");
// disable the checkbox and select it
toolbar.addCheckBox("checkbox", feedback_checkbox, true, "tooltip", false);
// make the checkbox non visible
toolbar.addCheckBox("checkbox", feedback_checkbox, false, "tooltip", false, false);
```

addCheckBox

void **addCheckBox** (text, method, selected, tooltip, enabled, visible)

Add a CheckBox to the toolbar.

Parameters

`{String}` text
`{Function}` method
`{Boolean}` selected
`{String}` tooltip
`{Boolean}` enabled
`{Boolean}` visible

Returns

void

Sample

```
// add a toolbar
var toolbar = plugins.window.addToolBar("toolbar_1");
// add a checkbox with a text and a method
toolbar.addCheckBox("checkbox", feedback_checkbox);
// add an checkbox and set it's state to selected (not selected by default)
toolbar.addCheckBox("checkbox", feedback_checkbox, true);
// add a tooltip to the checkbox
toolbar.addCheckBox("checkbox", feedback_checkbox, false, "tooltip");
// disable the checkbox and select it
toolbar.addCheckBox("checkbox", feedback_checkbox, true, "tooltip", false);
// make the checkbox non visible
toolbar.addCheckBox("checkbox", feedback_checkbox, false, "tooltip", false, false);
```

addComboBox

void **addComboBox** (method, index, values)

Add a ComboBox to the toolbar.

Parameters

`{Function}` method
`{Number}` index
`{String[]}` values

Returns

void

Sample

```
// add a toolbar
var toolbar = plugins.window.addToolBar("toolbar_2");
// add a combobox with the attached method, selected index and input (list) array
toolbar.addComboBox(feedback_combobox, 0, ["input","array","combobox",1]);
// add a tooltip to the combobox
toolbar.addComboBox(feedback_combobox, 1, ["input","array","combobox",2], "tooltip");
// disable the combobox
toolbar.addComboBox(feedback_combobox, 2, ["input","array","combobox",3], "tooltip", false);
// make the combobox non visible
toolbar.addComboBox(feedback_combobox, 3, ["input","array","combobox",4], "tooltip", false, false);
```

addComboBox

void **addComboBox** (method, index, values, tooltip)

Add a ComboBox to the toolbar.

Parameters

{[Function](#)} method
 {[Number](#)} index
 {[String](#)[]} values
 {[String](#)} tooltip

Returns

void

Sample

```
// add a toolbar
var toolbar = plugins.window.addToolBar("toolbar_2");
// add a combobox with the attached method, selected index and input (list) array
toolbar.addComboBox(feedback_combobox, 0, ["input","array","combobox",1]);
// add a tooltip to the combobox
toolbar.addComboBox(feedback_combobox, 1, ["input","array","combobox",2], "tooltip");
// disable the combobox
toolbar.addComboBox(feedback_combobox, 2, ["input","array","combobox",3], "tooltip", false);
// make the combobox non visible
toolbar.addComboBox(feedback_combobox, 3, ["input","array","combobox",4], "tooltip", false, false);
```

addComboBox

void **addComboBox** (method, index, values, tooltip, enabled)

Add a ComboBox to the toolbar.

Parameters

{[Function](#)} method
 {[Number](#)} index
 {[String](#)[]} values
 {[String](#)} tooltip
 {[Boolean](#)} enabled

Returns

void

Sample

```
// add a toolbar
var toolbar = plugins.window.addToolBar("toolbar_2");
// add a combobox with the attached method, selected index and input (list) array
toolbar.addComboBox(feedback_combobox, 0, ["input","array","combobox",1]);
// add a tooltip to the combobox
toolbar.addComboBox(feedback_combobox, 1, ["input","array","combobox",2], "tooltip");
// disable the combobox
toolbar.addComboBox(feedback_combobox, 2, ["input","array","combobox",3], "tooltip", false);
// make the combobox non visible
toolbar.addComboBox(feedback_combobox, 3, ["input","array","combobox",4], "tooltip", false, false);
```

addComboBox

void **addComboBox** (method, index, values, tooltip, enabled, visible)

Add a ComboBox to the toolbar.

Parameters

{Function} method
 {Number} index
 {String[]} values
 {String} tooltip
 {Boolean} enabled
 {Boolean} visible

Returns

void

Sample

```
// add a toolbar
var toolbar = plugins.window.addToolBar("toolbar_2");
// add a combobox with the attached method, selected index and input (list) array
toolbar.addComboBox(feedback_combobox, 0, ["input","array","combobox",1]);
// add a tooltip to the combobox
toolbar.addComboBox(feedback_combobox, 1, ["input","array","combobox",2], "tooltip");
// disable the combobox
toolbar.addComboBox(feedback_combobox, 2, ["input","array","combobox",3], "tooltip", false);
// make the combobox non visible
toolbar.addComboBox(feedback_combobox, 3, ["input","array","combobox",4], "tooltip", false, false);
```

addField

void **addField** (method, text)

Add a Field to the toolbar.

Parameters

{Function} method
 {String} text

Returns

void

Sample

```
// add a toolbar
var toolbar = plugins.window.addToolBar("toolbar_3");
// add a field with the attached method and a default text
toolbar.addField(feedback_field, null);
// set the length of the field.
// default length = 8 when length is not set or set to 0
toolbar.addField(feedback_field, "field", 0, "tooltip");
// add a tooltip to the field
toolbar.addField(feedback_field, null, 10, "tooltip");
// disable the field
toolbar.addField(feedback_field, "field", 5, "tooltip", false);
// make the field non visible
toolbar.addField(feedback_field, "field", 0, "tooltip", false, false);
```

addField

void **addField** (method, text, length)

Add a Field to the toolbar.

Parameters

{Function} method
 {String} text
 {Number} length

Returns

void

Sample

```
// add a toolbar
var toolbar = plugins.window.addToolBar("toolbar_3");
// add a field with the attached method and a default text
toolbar.addField(feedback_field, null);
// set the length of the field.
// default length = 8 when length is not set or set to 0
toolbar.addField(feedback_field, "field", 0, "tooltip");
// add a tooltip to the field
toolbar.addField(feedback_field, null, 10, "tooltip");
// disable the field
toolbar.addField(feedback_field, "field", 5, "tooltip", false);
// make the field non visible
toolbar.addField(feedback_field, "field", 0, "tooltip", false, false);
```

addField

void **addField** (method, text, length, tooltip)

Add a Field to the toolbar.

Parameters

`{Function}` method
`{String}` text
`{Number}` length
`{String}` tooltip

Returns

void

Sample

```
// add a toolbar
var toolbar = plugins.window.addToolBar("toolbar_3");
// add a field with the attached method and a default text
toolbar.addField(feedback_field, null);
// set the length of the field.
// default length = 8 when length is not set or set to 0
toolbar.addField(feedback_field, "field", 0, "tooltip");
// add a tooltip to the field
toolbar.addField(feedback_field, null, 10, "tooltip");
// disable the field
toolbar.addField(feedback_field, "field", 5, "tooltip", false);
// make the field non visible
toolbar.addField(feedback_field, "field", 0, "tooltip", false, false);
```

addField

void **addField** (method, text, length, tooltip, enabled)

Add a Field to the toolbar.

Parameters

`{Function}` method
`{String}` text
`{Number}` length
`{String}` tooltip
`{Boolean}` enabled

Returns

void

Sample

```
// add a toolbar
var toolbar = plugins.window.addToolBar("toolbar_3");
// add a field with the attached method and a default text
toolbar.addField(feedback_field, null);
// set the length of the field.
// default length = 8 when length is not set or set to 0
toolbar.addField(feedback_field, "field", 0, "tooltip");
// add a tooltip to the field
toolbar.addField(feedback_field, null, 10, "tooltip");
// disable the field
toolbar.addField(feedback_field, "field", 5, "tooltip", false);
// make the field non visible
toolbar.addField(feedback_field, "field", 0, "tooltip", false, false);
```

addField

void **addField** (method, text, length, tooltip, enabled, visible)

Add a Field to the toolbar.

Parameters

{Function} method
 {String} text
 {Number} length
 {String} tooltip
 {Boolean} enabled
 {Boolean} visible

Returns

void

Sample

```
// add a toolbar
var toolbar = plugins.window.addToolBar("toolbar_3");
// add a field with the attached method and a default text
toolbar.addField(feedback_field, null);
// set the length of the field.
// default length = 8 when length is not set or set to 0
toolbar.addField(feedback_field, "field", 0, "tooltip");
// add a tooltip to the field
toolbar.addField(feedback_field, null, 10, "tooltip");
// disable the field
toolbar.addField(feedback_field, "field", 5, "tooltip", false);
// make the field non visible
toolbar.addField(feedback_field, "field", 0, "tooltip", false, false);
```

addSeparator

void **addSeparator** ()

Add a Separator to the toolbar.

Returns

void

Sample

```
// add a toolbar
var toolbar = plugins.window.addToolBar("toolbar_0");
// add a button
toolbar.addButton("button", feedback_button);
// add a separator
toolbar.addSeparator();
// add a checkbox
toolbar.addCheckBox("checkbox", feedback_checkbox);
```

enableItem

void **enableItem** (index, enabled)

Enable/disable the item at the specified index.

Parameters

`{Number}` index
`{Boolean}` enabled

Returns

void

Sample

```
// add a toolbar
var toolbar = plugins.window.addToolBar("toolbar_1");
// add a checkbox
toolbar.addCheckBox("checkbox", feedback_checkbox);
// disable the button
// REMARK: the pitfall here is that the indexes start at position 1 here
// position 0 is reserved for the toolbar handle!
toolbar.enableItem(1, false);
```

removeAllItems

void **removeAllItems** ()

Remove all Buttons, Checkboxes etc. from the toolbar.

Returns

void

Sample

```
// add a toolbar
var toolbar = plugins.window.addToolBar("toolbar_1");
// add a button
toolbar.addButton("button", feedback_button);
// add a checkbox
toolbar.addCheckBox("checkbox", feedback_checkbox);
// remove all items from the toolbar
toolbar.removeAllItems();
```

removeItem

void **removeItem** (index)

Remove a Button, CheckBox, ComboBox from the toolbar.

Parameters

`{Number}` index

Returns

void

Sample

```
// add a toolbar
var toolbar = plugins.window.addToolBar("toolbar_1");
// add a button
toolbar.addButton("button", feedback_button);
// add a checkbox
toolbar.addCheckBox("checkbox", feedback_checkbox);
// remove the first item (the button in this case) from the toolbar
// REMARK: the pitfall here is that the indexes start at position 1 here
// position 0 is reserved for the toolbar handle!
toolbar.removeItem(1);
```

selectCheckBox

void **selectCheckBox** (index, selected)

Set the CheckBox selection.

Parameters

`{Number}` index
`{Boolean}` selected

Returns

void

Sample

```
// add a toolbar
var toolbar = plugins.window.addToolBar("toolbar_1");
// add a checkbox
toolbar.addCheckBox("checkbox 1", feedback_checkbox);
// add another checkbox
toolbar.addCheckBox("checkbox 2", feedback_checkbox);
// set the selection of the checkboxes
// REMARK: the pitfall here is that the indexes start at position 1 here
// position 0 is reserved for the toolbar handle!
toolbar.selectCheckBox(1, false);
toolbar.selectCheckBox(2, true);
```

selectComboBox

void **selectComboBox** (index, selection)

Select a row of the ComboBox via the index.

Parameters

{[Number](#)} index
{[Number](#)} selection

Returns

void

Sample

```
// add a toolbar
var toolbar = plugins.window.addToolBar("toolbar_1");
// add a combobox
toolbar.addComboBox(feedback_combobox, 1, ["one", "two", "three"]);
// add another combobox
toolbar.addComboBox(feedback_combobox, 2, [1, 2, 3, 4, 5]);
// set the selection of the comboboxes
// REMARK: the pitfall here is that the indexes start at position 1 here
// position 0 is reserved for the toolbar handle!
toolbar.selectComboBox(1, 0); // entry "one" will be selected in the first combobox
toolbar.selectComboBox(2, 3); // entry 4 will be selected in the second combobox
```

setFieldText

void **setFieldText** (index, text)

Set a (default) text of the field at the given index.

Parameters

{[Number](#)} index
{[String](#)} text

Returns

void

Sample

```
// add a toolbar
var toolbar = plugins.window.addToolBar("toolbar_1");
// add a field
toolbar.addField(feedback_field, "field one");
// add another field
toolbar.addField(feedback_field , "field_two");
// set the text of the fields
// REMARK: the pitfall here is that the indexes start at position 1 here
// position 0 is reserved for the toolbar handle!
toolbar.setFieldText(1, "new text 1");
toolbar.setFieldText(2, "new text 2");
```

visibleItem

void **visibleItem** (index, visible)

Make the item at the specified index visible/invisible.

Parameters

{[Number](#)} index
{[Boolean](#)} visible

Returns

void

Sample

```
// add a toolbar
var toolbar = plugins.window.addToolBar("toolbar_1");
// add a button
toolbar.addButton("button", feedback_button);
// add a checkbox
toolbar.addCheckBox("checkbox", feedback_checkbox);
// make the first item (the button) invisible
// REMARK: the pitfall here is that the indexes start at position 1 here
// position 0 is reserved for the toolbar handle!
toolbar.visibleItem(1, false);
```