ToolBar

Method Summery

void #addButton(text, method, [arguments], [icon], [tooltip], [enabled], [visible])
Add a Button to the toolbar.

void #addCheckBox(text, method, [selected], [tooltip], [enabled], [visible])
Add a CheckBox to the toolbar.

void #addComboBox(method, index, input, [tooltip], [enabled], [visible])
Add a ComboBox to the toolbar.

void #addField(method, text, [tooltip], [enabled], [visible])
Add a Field to the toolbar.

void #addSeparator()
Add a Separator to the toolbar.

void #enableItem()

void #removeAllItems()
Remove all Buttons, Checkboxes etc.

void #removeItem(index)
Remove a Button, CheckBox, ComboBox from the toolbar.

void #selectCheckBox(index, boolean)
Set the CheckBox selection.

void #selectComboBox(index, rowindex)
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 #validate()
You need to call this method after adding or removing items to/from the toolbar.

void #visibleItem()

Method Details

addButton

void **addButton**(text, method, [arguments], [icon], [tooltip], [enabled], [visible])
Add a Button to the toolbar.

Parameters
text
method
[arguments]
[icon]
[tooltip]
[enabled]
[visible]

Returns
void

Sample

```
// get the toolbar at the panel by name
var toolbar = plugins.window.getToolBar("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);

// add a separator
toolbar.addSeparator();

// make the button non visible
toolbar.addButton(null, feedback_button, [1, "2", "three"], "media:///yourimage.gif", "tooltip.",true, false);

// and validate the changes
// to make them know to the user interface)
toolbar.validate();
```

addCheckBox

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

Add a CheckBox to the toolbar.

Parameters

text
method
[selected]
[tooltip]
[enabled]
[visible]

Returns

void

Sample

```
// add a toolbar with a name and internal name
var toolbar = plugins.window.addToolBar("toolbar_1", "toolbar_1");

// add a checkbox with a text and a method
toolbar.addCheckBox("checkbox", feedback_button);

// add an checkbox and set it's state to selected (not selected by default)
toolbar.addCheckBox("checkbox", feedback_button, true);

// add a tooltip to the checkbox
toolbar.addCheckBox("checkbox", feedback_button, false, "tooltip");

// add a separator
toolbar.addSeparator();

// disable the checkbox and select it
toolbar.addCheckBox("checkbox", feedback_button, true, "tooltip",false);

// make the button non visible
toolbar.addCheckBox("checkbox", feedback_button, false, "tooltip",false, false);

// and validate the changes
// to make them know to the user interface)
toolbar.validate();
```

addComboBox

void **addComboBox**(method, index, input, [tooltip], [enabled], [visible])

Add a ComboBox to the toolbar.

Parameters

method
index
input
[tooltip]
[enabled]
[visible]

Returns

void

Sample

```
// add a toolbar with a name and internal name at the given row index
var toolbar = plugins.window.addToolBar("toolbar_2", "toolbar_2", 3);

// add a combobox with the attached method, selected index and input (list) array
toolbar.addComboBox(feedback_button, 0, ["input","array","combobox",1]);

// add a tooltip to the combobox
toolbar.addComboBox(feedback_button, 1, ["input","array","combobox",2], "tooltip");

// add a separator
toolbar.addSeparator();

// disable the combobox
toolbar.addComboBox(feedback_button, 2, ["input","array","combobox",3], "tooltip",false);

// make the combobox non visible
toolbar.addComboBox(feedback_button, 3, ["input","array","combobox",4], "tooltip",false, false);

// and validate the changes
// to make them know to the user interface)
toolbar.validate();
```

addField

void **addField**(method, text, [tooltip], [enabled], [visible])

Add a Field to the toolbar.

Parameters
method
text
[tooltip]
[enabled]
[visible]
Returns
void
Sample

```
// add a toolbar with a name and internal name at the given row index
var toolbar = plugins.window.addToolBar("toolbar_3", "toolbar_3", 4);

// add a field with the attached method and a default text
toolbar.addField(feedback_button, null);

// add a separator
toolbar.addSeparator();

// set the length of the field.
// default length = 8 when length is not set or set to 0
toolbar.addField(feedback_button, "field", 0, "tooltip");

// add a separator
toolbar.addSeparator();

// add a tooltip to the field
toolbar.addField(feedback_button, "field", 10, "tooltip");

// add a separator
toolbar.addSeparator();

// disable the field
toolbar.addField(feedback_button, null, 5, "tooltip",false);

// add a separator
toolbar.addSeparator();

// make the field non visible
toolbar.addField(feedback_button, "field", 0, "tooltip",false, false);

// and validate the changes
// to make them know to the user interface)
toolbar.validate();
```

addSeparator

void **addSeparator**()

Add a Separator to the toolbar.

Returns
void

Sample

```
// get the toolbar at the panel by name
var toolbar = plugins.window.getToolBar("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);

// add a separator
toolbar.addSeparator();

// make the button non visible
toolbar.addButton(null, feedback_button, [1, "2", "three"], "media:///yourimage.gif", "tooltip.",true, false);

// and validate the changes
// to make them know to the user interface)
toolbar.validate();
```

enableItem

void **enableItem**()

Replace with description

Returns

void

Sample

```
// get the toolbar at the panel by name
var toolbar = plugins.window.getToolBar("toolbar_1");

// enable/disable the selected item at the index
// 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

```
// get the toolbar at the panel by name
var toolbar = plugins.window.getToolBar("toolbar_0");

// remove all buttons from the toolbar
toolbar.removeAllItems();
```

removeItem

void **removeItem**(index)

Remove a Button, CheckBox, ComboBox from the toolbar.

Parameters

index

Returns

void

Sample

```
// get the toolbar at the panel by name
var toolbar = plugins.window.getToolBar("toolbar_0");

// remove the button, checkbox, combobox, separator or field 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);

// and validate the changes
// to make them know to the user interface)
toolbar.validate();
```

selectCheckBox

void **selectCheckBox**(index, boolean)

Set the CheckBox selection.

Parameters

index
boolean

Returns

void

Sample

```
// get the toolbar at the panel by name
var toolbar = plugins.window.getToolBar("toolbar_1");

// set the selection of the checkbox at the index
// 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, rowindex)

Select a row of the ComboBox via the index.

Parameters

index
rowindex

Returns

void

Sample

```
// get the toolbar at the panel by name
var toolbar = plugins.window.getToolBar("toolbar_2");

// set the selection of the combobox at the index
// 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);
toolbar.selectComboBox(2, 0);
```

setFieldText

void **setFieldText**(index, text)

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

Parameters

index
text

Returns

void

Sample

```
// get the toolbar at the panel by name
var toolbar = plugins.window.getToolBar("toolbar_3");

// set the text of the field at the index
// 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");
```

validate

void **validate**()

You need to call this method after adding or removing items to/from the toolbar.

Returns

void

Sample

```
// get the toolbar at the panel by name
var toolbar = plugins.window.getToolBar("toolbar_0");

// remove the button, checkbox, combobox, separator or field 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);

// and validate the changes
// to make them know to the user interface)
toolbar.validate();
```

visibleItem

void **visibleItem**()

Replace with description

Returns

void

Sample

```
// get the toolbar at the panel by name
var toolbar = plugins.window.getToolBar("toolbar_2");

// make the selected item at the index visible/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);
```