

BlendSplitter

3.0.0

Generated by Doxygen 1.8.11

Contents

1	Hierarchical Index	1
1.1	Class Hierarchy	1
2	Class Index	3
2.1	Class List	3
3	Class Documentation	5
3.1	BlendSplitter Class Reference	5
3.1.1	Detailed Description	7
3.1.2	Constructor & Destructor Documentation	7
3.1.2.1	BlendSplitter(QWidget *(*defaultWidget)())=[]() ->QWidget *{return new SwitchingWidget{}};, Qt::Orientation orientation=Qt::Horizontal)	7
3.1.3	Member Function Documentation	7
3.1.3.1	addSplitter(BlendSplitter *splitter)	7
3.1.3.2	addWidget()	7
3.1.3.3	addWidget(QWidget *widget)	8
3.1.3.4	insertSplitter(int index, BlendSplitter *splitter)	8
3.1.3.5	insertWidget(int index)	8
3.1.3.6	insertWidget(int index, QWidget *widget)	8
3.1.4	Member Data Documentation	9
3.1.4.1	expanderImage	9
3.1.4.2	expanderSize	9
3.1.4.3	switchingBarHeight	9
3.2	RegistryItem Class Reference	9
3.2.1	Detailed Description	10

3.2.2	Constructor & Destructor Documentation	10
3.2.2.1	RegistryItem(QString name=""Default"", QWidget *(*widget)()=[]) ->QWidget *{return new QLabel{""Default Widget""};}, void(*populateBar)(SwitchingBar *, QWidget *)=[](SwitchingBar *, QWidget *) ->void{}	10
3.2.3	Member Data Documentation	10
3.2.3.1	name	10
3.2.3.2	populateBar	11
3.2.3.3	widget	11
3.3	SwitchingBar Class Reference	11
3.3.1	Detailed Description	13
3.3.2	Member Function Documentation	13
3.3.2.1	addMenu(QMenu *menu)	13
3.3.2.2	addWidget(QWidget *widget)	13
3.4	SwitchingWidget Class Reference	13
3.4.1	Detailed Description	14
3.4.2	Constructor & Destructor Documentation	15
3.4.2.1	SwitchingWidget(QWidget *parent=nullptr)	15
3.5	WidgetRegistry Class Reference	15
3.5.1	Detailed Description	18
3.5.2	Member Function Documentation	18
3.5.2.1	addItem(RegistryItem *item)	18
3.5.2.2	addItem(QString name=""Default"", QWidget *(*widget)()=[]) ->QWidget *{return new QLabel{""Default widget""};}, void(*populateBar)(SwitchingBar *, QWidget *)=[](SwitchingBar *, QWidget *) ->void{}	18
3.5.2.3	getDefault()	19
3.5.2.4	getRegistry()	19
3.5.2.5	indexOf(RegistryItem *item) const	19
3.5.2.6	insertItem(int index, RegistryItem *item)	20
3.5.2.7	insertItem(int index, QString name=""Default"", QWidget *(*widget)()=[]) - >QWidget *{return new QLabel{""Default widget""};}, void(*populateBar)(← SwitchingBar *, QWidget *)=[](SwitchingBar *, QWidget *) ->void{}	20
3.5.2.8	item(int i) const	20
3.5.2.9	registryChanged	20
3.5.2.10	removeItem(RegistryItem *item)	21
3.5.2.11	removeItem(int index)	21
3.5.2.12	setDefault(RegistryItem *item)	21
3.5.2.13	setDefault(int index=0)	21
3.5.2.14	size() const	22

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

QObject	
WidgetRegistry	15
QSplitter	
BlendSplitter	5
SwitchingWidget	13
QWidget	
SwitchingBar	11
RegistryItem	9

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

BlendSplitter		
	A user-defined Splitter	5
RegistryItem		
	An item intended to be put into WidgetRegistry	9
SwitchingBar		
	A menu bar which is always found on the bottom of SwitchingWidget	11
SwitchingWidget		
	A widget whose actual content can be selected from a combo box	13
WidgetRegistry		
	A registry of all widgets that can be displayed in a SwitchingWidget	15

Chapter 3

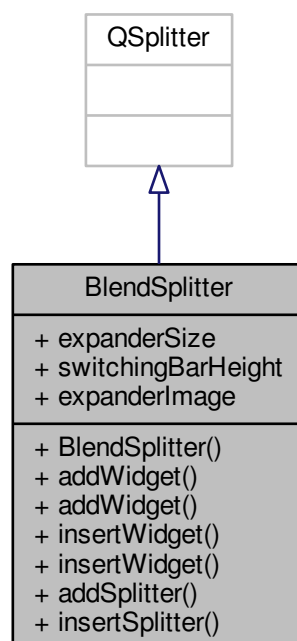
Class Documentation

3.1 BlendSplitter Class Reference

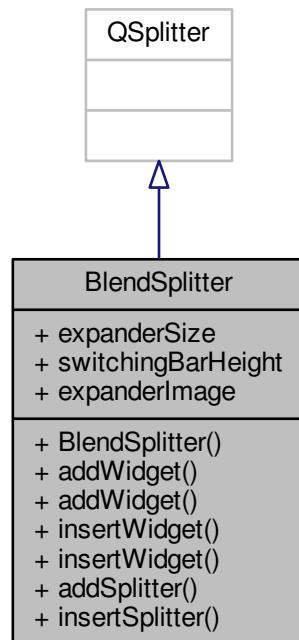
A user-defined Splitter.

```
#include <BlendSplitter.hpp>
```

Inheritance diagram for BlendSplitter:



Collaboration diagram for BlendSplitter:



Public Member Functions

- `BlendSplitter` (`QWidget *(*defaultWidget)()=[]()` -> `QWidget *`{return new `SwitchingWidget`{}, Qt::Orientation orientation=Qt::Horizontal)
BlendSplitter class constructor.
- void `addWidget` ()
Add a widget to the BlendSplitter.
- void `addWidget` (`QWidget *widget`)
Add a widget to the BlendSplitter.
- void `insertWidget` (int index)
Insert a widget into the BlendSplitter.
- void `insertWidget` (int index, `QWidget *widget`)
Insert a widget into the BlendSplitter.
- void `addSplitter` (`BlendSplitter *splitter`)
Add another BlendSplitter to this BlendSplitter.
- void `insertSplitter` (int index, `BlendSplitter *splitter`)
Insert another BlendSplitter into this BlendSplitter.

Static Public Attributes

- static int `expanderSize`
- static int `switchingBarHeight`
- static QString `expanderImage`

3.1.1 Detailed Description

A user-defined Splitter.

This widget implements the functionality of Blender (Open-source 3D modelling software) widget management. This widget displays a splitter similar to QSplitter. However, each widget in [BlendSplitter](#) has a pair of Expanders (one in top right and one in bottom left corner). By dragging from these Expanders inwards a new widget is created in the direction of the drag. If the direction is different to that of the [BlendSplitter](#), a new [BlendSplitter](#) with parallel direction is created in place of the widget with the widget and the new widget in it. By dragging from these expanders outwards, a neighbouring widget (or a collection of widgets) can be closed. While the mouse is held, the widgets to be closed are marked with black overlay. When the mouse is released, they are closed. [BlendSplitter](#) can be used like any other QWidget, although setting one as the central widget is recommended. A [BlendSplitter](#) can contain any QWidget, but to achieve best results, use it together with [SwitchingWidget](#).

Definition at line 25 of file BlendSplitter.hpp.

3.1.2 Constructor & Destructor Documentation

3.1.2.1 [BlendSplitter::BlendSplitter](#) ([QWidget](#) *defaultWidget* =
 [] () ->QWidget *{return new SwitchingWidget{}};*, [Qt::Orientation](#)
orientation = Qt::Horizontal)

[BlendSplitter](#) class constructor.

Parameters

<i>defaultWidget</i>	A pointer to function constructing the default widget. This function is called when a new widget is added to BlendSplitter .
<i>orientation</i>	Orientation of the main BlendSplitter

3.1.3 Member Function Documentation

3.1.3.1 [void BlendSplitter::addSplitter](#) ([BlendSplitter](#) * *splitter*)

Add another [BlendSplitter](#) to this [BlendSplitter](#).

Adds a [BlendSplitter](#) (usually with parallel orientation) to the [BlendSplitter](#)

Parameters

<i>splitter</i>	A pointer to the BlendSplitter to be added
-----------------	--

3.1.3.2 [void BlendSplitter::addWidget](#) ()

Add a widget to the [BlendSplitter](#).

Adds the default widget to the very bottom/right of the [BlendSplitter](#).

3.1.3.3 void BlendSplitter::addWidget (QWidget * *widget*)

Add a widget to the [BlendSplitter](#).

Adds the specified widget to the very bottom/right of the [BlendSplitter](#)

Parameters

<i>widget</i>	A pointer to the widget to be added
---------------	-------------------------------------

3.1.3.4 void BlendSplitter::insertSplitter (int *index*, BlendSplitter * *splitter*)

Insert another [BlendSplitter](#) into this [BlendSplitter](#).

Inserts a [BlendSplitter](#) (usually with parallel orientation) into the [BlendSplitter](#) at the given position counting from top/left (counting starts at 0).

Parameters

<i>index</i>	The desired position
<i>splitter</i>	A pointer to the BlendSplitter to be inserted

3.1.3.5 void BlendSplitter::insertWidget (int *index*)

Insert a widget into the [BlendSplitter](#).

Inserts the default widget into the [BlendSplitter](#) at the given position counting from top/left (counting starts at 0). This function should NOT be called with a [BlendSplitter](#) as a parameter.

Parameters

<i>index</i>	The desired position
--------------	----------------------

3.1.3.6 void BlendSplitter::insertWidget (int *index*, QWidget * *widget*)

Insert a widget into the [BlendSplitter](#).

Inserts the specified widget into the [BlendSplitter](#) at the given position counting from top/left (counting starts at 0). This function should NOT be called with a [BlendSplitter](#) as a parameter.

Parameters

<i>index</i>	The desired position
<i>widget</i>	A pointer to the widget to be inserted

3.1.4 Member Data Documentation

3.1.4.1 QString BlendSplitter::expanderImage [static]

The image to be used for the top left expander. The bottom right one will rotate this by pi (180 degrees).

Definition at line 32 of file BlendSplitter.hpp.

3.1.4.2 int BlendSplitter::expanderSize [static]

Size of the expanders in the corners

Definition at line 30 of file BlendSplitter.hpp.

3.1.4.3 int BlendSplitter::switchingBarHeight [static]

Height of the [SwitchingBar](#). A static property.

Definition at line 31 of file BlendSplitter.hpp.

The documentation for this class was generated from the following file:

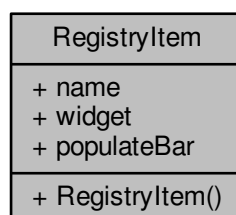
- include/BlendSplitter.hpp

3.2 RegistryItem Class Reference

An item intended to be put into [WidgetRegistry](#).

```
#include <RegistryItem.hpp>
```

Collaboration diagram for RegistryItem:



Public Member Functions

- [RegistryItem](#) (QString [name](#)="Default", QWidget [*\(*widget\)\(\)](#)=[]) ->QWidget [*{return new QLabel{"Default Widget"};};](#), void([*populateBar](#))([SwitchingBar](#) *, QWidget [*](#))=[])([SwitchingBar](#) *, QWidget [*](#)) ->void{}

A constructor setting all the internal values.

Public Attributes

- QString [name](#)
- QWidget [*\(* widget\)\(\)](#)
A function constructing the widget.
- void([* populateBar](#))([SwitchingBar](#) *, QWidget [*](#))
A function populating the [SwitchingBar](#).

3.2.1 Detailed Description

An item intended to be put into [WidgetRegistry](#).

Each [RegistryItem](#) corresponds to one widget that can be displayed in a [BlendSplitter](#). It describes how this widget should be constructed, what is its name and what items should be in the [SwitchingBar](#) when this widget is selected.

Definition at line 17 of file RegistryItem.hpp.

3.2.2 Constructor & Destructor Documentation

3.2.2.1 `RegistryItem::RegistryItem (QString name = "Default", QWidget *\(*\(\)\) widget = []() ->QWidget *{return new QLabel{"Default Widget"};};, void(*populateBar = [])(SwitchingBar *, QWidget *) ->void{ })`

A constructor setting all the internal values.

This constructor takes 3 parameters corresponding to the 3 members of the [RegistryItem](#) class. See their description for more details.

Parameters

<i>name</i>	The name of the widget, used in the SwitchingBar combo box
<i>widget</i>	A pointer to a function constructing the widget
<i>populateBar</i>	A pointer to a function populating the SwitchingBar

3.2.3 Member Data Documentation

3.2.3.1 QString RegistryItem::name

The name of the widget, used in the [SwitchingBar](#) combo box.

Definition at line 20 of file RegistryItem.hpp.

3.2.3.2 void(* RegistryItem::populateBar) (SwitchingBar *, QWidget *)

A function populating the [SwitchingBar](#).

A pointer to a function populating the [SwitchingBar](#). This function is called each time this widget is selected in any [SwitchingWidget](#). Usually this function makes use of the interface provided by [SwitchingBar](#) to populate it.

Parameters

A	pointer to the SwitchingBar to be populated
A	pointer to the newly-created widget in the SwitchingWidget

Definition at line 33 of file RegistryItem.hpp.

3.2.3.3 QWidget*(* RegistryItem::widget) ()

A function constructing the widget.

A pointer to a function returning QWidget*. This function is called to construct the widget each time it is selected in any [SwitchingWidget](#). Usually in this function the widget is dynamically created using `new` operator and the pointer is returned.

Returns

A pointer to the newly-created QWidget

Definition at line 26 of file RegistryItem.hpp.

The documentation for this class was generated from the following file:

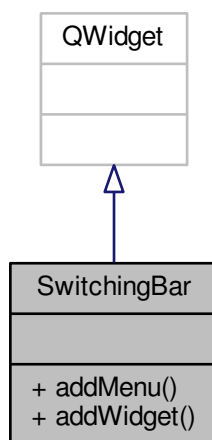
- include/BS/RegistryItem.hpp

3.3 SwitchingBar Class Reference

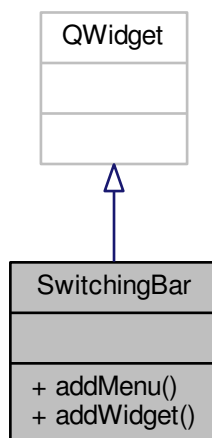
A menu bar which is always found on the bottom of [SwitchingWidget](#).

```
#include <SwitchingBar.hpp>
```

Inheritance diagram for SwitchingBar:



Collaboration diagram for SwitchingBar:



Public Member Functions

- void [addMenu](#) (QMenu *menu)
Add a QMenu.
- void [addWidget](#) (QWidget *widget)
Add a QWidget.

3.3.1 Detailed Description

A menu bar which is always found on the bottom of [SwitchingWidget](#).

This menu bar is similar to the built-in `QMenuBar`, but can also contain plain `QWidgets`. The first item on the left is always a combo box for selecting which widget should be displayed in the [SwitchingWidget](#).

Definition at line 20 of file `SwitchingBar.hpp`.

3.3.2 Member Function Documentation

3.3.2.1 `void SwitchingBar::addMenu (QMenu * menu)`

Add a `QMenu`.

This function adds a `QMenu` to the very right of the [SwitchingBar](#). The menu is wrapped in an invisible `QMenuBar`.

Parameters

<i>menu</i>	A pointer to the <code>QMenu</code> to be added
-------------	---

3.3.2.2 `void SwitchingBar::addWidget (QWidget * widget)`

Add a `QWidget`.

This function adds a `QWidget` to the very right of the [SwitchingBar](#). The widget is placed in a `QHBoxLayout`.

Parameters

<i>widget</i>	A pointer to the <code>QWidget</code> to be added
---------------	---

The documentation for this class was generated from the following file:

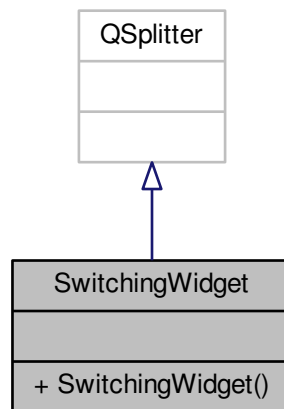
- `include/BS/SwitchingBar.hpp`

3.4 SwitchingWidget Class Reference

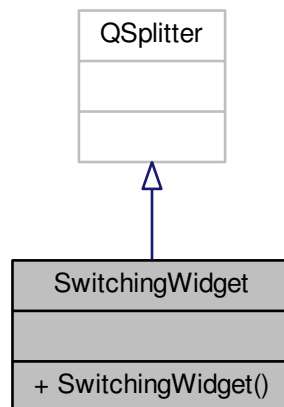
A widget whose actual content can be selected from a combo box.

```
#include <SwitchingWidget.hpp>
```

Inheritance diagram for SwitchingWidget:



Collaboration diagram for SwitchingWidget:



Public Member Functions

- [SwitchingWidget](#) (`QWidget *parent=nullptr`)
A default constructor similar to that of `QWidget`.

3.4.1 Detailed Description

A widget whose actual content can be selected from a combo box.

This widget displays a Widget with a [SwitchingBar](#) on the bottom. The widget displayed is one from [WidgetRegistry](#) and it can be selected using a combo box in the [SwitchingBar](#).

Definition at line 17 of file SwitchingWidget.hpp.

3.4.2 Constructor & Destructor Documentation

3.4.2.1 SwitchingWidget::SwitchingWidget (QWidget * *parent* = nullptr) [explicit]

A default constructor similar to that of QWidget.

Creates a [SwitchingWidget](#) containing the default widget specified in [WidgetRegistry](#)

Parameters

<i>parent</i>	A parent widget
---------------	-----------------

The documentation for this class was generated from the following file:

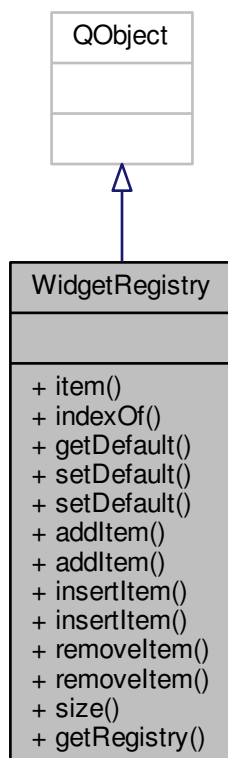
- include/BS/SwitchingWidget.hpp

3.5 WidgetRegistry Class Reference

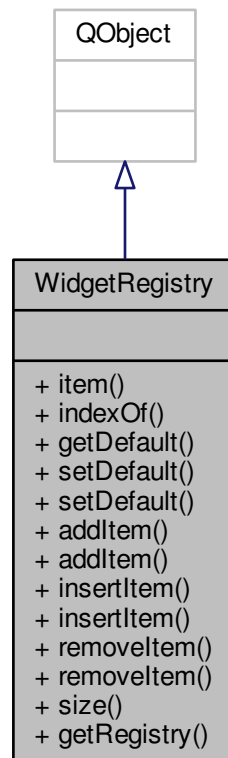
A registry of all widgets that can be displayed in a [SwitchingWidget](#).

```
#include <WidgetRegistry.hpp>
```

Inheritance diagram for WidgetRegistry:



Collaboration diagram for WidgetRegistry:



Signals

- void `registryChanged` ()
Signal emitted when `WidgetRegistry` changes its contents.

Public Member Functions

- `RegistryItem * item` (int i) const
Get the item at position *i*.
- int `indexOf` (RegistryItem *item) const
Get the position of an item in `WidgetRegistry`.
- `RegistryItem * getDefault` ()
Get the default `RegistryItem`.
- void `setDefault` (RegistryItem *item)
Set the default `RegistryItem`.
- void `setDefault` (int index=0)
Set the default `RegistryItem`.
- void `addItem` (RegistryItem *item)
Add an item to `WidgetRegistry`.

- void `addItem` (QString name="Default", QWidget *(*widget)()=[]) ->QWidget *{return new QLabel{"Default widget"};}, void(*populateBar)(SwitchingBar *, QWidget *)=[](SwitchingBar *, QWidget *) ->void{}
Add an item to [WidgetRegistry](#).
- void `insertItem` (int index, RegistryItem *item)
Insert an item into [WidgetRegistry](#).
- void `insertItem` (int index, QString name="Default", QWidget *(*widget)()=[]) ->QWidget *{return new QLabel{"Default widget"};}, void(*populateBar)(SwitchingBar *, QWidget *)=[](SwitchingBar *, QWidget *) ->void{}
Insert an item into [WidgetRegistry](#).
- void `removeItem` (RegistryItem *item)
Remove a [RegistryItem](#) from [WidgetRegistry](#).
- void `removeItem` (int index)
Remove a [RegistryItem](#) from [WidgetRegistry](#).
- int `size` () const
Get the size of [WidgetRegistry](#).

Static Public Member Functions

- static [WidgetRegistry](#) * `getRegistry` ()
Registry getter.

3.5.1 Detailed Description

A registry of all widgets that can be displayed in a [SwitchingWidget](#).

This singleton-class acts as a registry of widgets that can be displayed in a [SwitchingWidget](#) by selecting from a combo box in the [SwitchingBar](#). Each item is represented as one [RegistryItem](#). The Registry also contains a pointer to the default [RegistryItem](#), which is shown when a new [SwitchingWidget](#) is created.

Definition at line 18 of file `WidgetRegistry.hpp`.

3.5.2 Member Function Documentation

3.5.2.1 void WidgetRegistry::addItem (RegistryItem * item)

Add an item to [WidgetRegistry](#).

Adds a given [RegistryItem](#) at the end of [WidgetRegistry](#).

Parameters

<i>item</i>	A pointer to the RegistryItem to be added
-------------	---

```
3.5.2.2 void WidgetRegistry::addItem ( QString name = "Default", QWidget *(*()) widget =
[]() ->QWidget *{return new QLabel{"Default widget"};}, void*)(SwitchingBar
*, QWidget *) populateBar = [] (SwitchingBar *, QWidget *) ->void{} )
```

Add an item to [WidgetRegistry](#).

Adds a [RegistryItem](#) constructed with the given parameters at the end of [WidgetRegistry](#). This is equal to calling `addItem(new RegistryItem{name, widget, populateBar})`.

Parameters

<i>name</i>	The name of the widget, used in the SwitchingBar combo box
<i>widget</i>	A pointer to a function constructing the widget
<i>populateBar</i>	A pointer to a function populating the SwitchingBar

3.5.2.3 RegistryItem* WidgetRegistry::getDefault ()

Get the default [RegistryItem](#).

This function gives you the default [RegistryItem](#). Note that if no item was set as default, the currently first item is set as default by this function. If the registry is empty, a [RegistryItem](#) is added to the registry (using the default constructor) and set as default.

Returns

A pointer to the default [RegistryItem](#)

3.5.2.4 static WidgetRegistry* WidgetRegistry::getRegistry () [static]

Registry getter.

This is a singleton class, i. e. you can't construct any object yourself. To get the one-and-only instance of this class, you need to call [WidgetRegistry::getRegistry\(\)](#). The function will create the object if necessary (= when called for the first time) and return a pointer to it.

Returns

A pointer to the one-and-only instance of [WidgetRegistry](#)

3.5.2.5 int WidgetRegistry::indexOf (RegistryItem * item) const

Get the position of an item in [WidgetRegistry](#).

Get the index (counting starts at 0) of an item. Often used together with [item\(int i\) const](#).

Parameters

<i>item</i>	A pointer to the item whose index is to be returned
-------------	---

Returns

Index of the item

3.5.2.6 void WidgetRegistry::insertItem (int index, RegistryItem * item)

Insert an item into [WidgetRegistry](#).

Inserts a given [RegistryItem](#) into [WidgetRegistry](#) at a given index.

Parameters

<i>index</i>	The desired index of the inserted RegistryItem (counting starts at 0)
<i>item</i>	A pointer to the RegistryItem to be added

```
3.5.2.7 void WidgetRegistry::insertItem ( int index, QString name = "Default ", QWidget *(*()) widget =
[] () ->QWidget *{return new QLabel{"Default widget"}; }, void(*) (SwitchingBar
*, QWidget *) populateBar = [] (SwitchingBar *, QWidget *) ->void{} )
```

Insert an item into widgetRegistry.

Inserts a [RegistryItem](#) constructed with the given parameters into [WidgetRegistry](#) at a given index. This is equal to calling insertItem(index, new [RegistryItem](#)(name, widget, populateBar)).

Parameters

<i>index</i>	The desired index of the inserted RegistryItem (counting starts at 0)
<i>name</i>	The name of the widget, used in the SwitchingBar combo box
<i>widget</i>	A pointer to a function constructing the widget
<i>populateBar</i>	A pointer to a function populating the SwitchingBar

3.5.2.8 RegistryItem* WidgetRegistry::item (int i) const

Get the item at position i.

This function gives you the item at position i (counting starts at 0).

Parameters

<i>i</i>	Index of the item to be returned
----------	----------------------------------

Returns

A pointer to the [RegistryItem](#) at position i

3.5.2.9 void WidgetRegistry::registryChanged () [signal]

Signal emitted when [WidgetRegistry](#) changes its contents.

This signal is emitted when a [RegistryItem](#) is added, inserted or removed from [WidgetRegistry](#). It is NOT emitted when the default item is changed unless this change requires adding a [RegistryItem](#).

3.5.2.10 void WidgetRegistry::removeItem (RegistryItem * item)

Remove a [RegistryItem](#) from [WidgetRegistry](#).

Removes a given [RegistryItem](#) from [WidgetRegistry](#). If this is also the default [RegistryItem](#), the first [RegistryItem](#) is set as default if it exists. This is equal to calling `removeItem(indexOf(item))`.

Parameters

<i>item</i>	
-------------	--

3.5.2.11 void WidgetRegistry::removeItem (int index)

Remove a [RegistryItem](#) from [WidgetRegistry](#).

Removes the [RegistryItem](#) at position index (counting starts at 0) from [WidgetRegistry](#). If this is also the default [RegistryItem](#), the first [RegistryItem](#) is set as default if it exists.

Parameters

<i>index</i>	Index of the RegistryItem to be removed
--------------	---

3.5.2.12 void WidgetRegistry::setDefault (RegistryItem * item)

Set the default [RegistryItem](#).

This function sets the default [RegistryItem](#). Note that if the item is not in [WidgetRegistry](#), it is added as the last entry. The default item is used when a new [SwitchingWidget](#) is created as the displayed widget.

Parameters

<i>item</i>	A pointer to the RegistryItem to be set as default
-------------	--

3.5.2.13 void WidgetRegistry::setDefault (int index = 0)

Set the default [RegistryItem](#).

This function sets the default [RegistryItem](#) to be the [RegistryItem](#) at given index. This is equal to calling `setDefault(item(index))`.

Parameters

<i>index</i>	Index of the RegistryItem to be set as default (counting starts at 0)
--------------	---

3.5.2.14 int WidgetRegistry::size () const

Get the size of [WidgetRegistry](#).

This function returns the number of RegistryItems currently in [WidgetRegistry](#). Note that these are indexed as 0, ..., ([size\(\)](#) - 1).

Returns

Number of RegistryItems in [WidgetRegistry](#)

The documentation for this class was generated from the following file:

- include/BS/WidgetRegistry.hpp

Index

- addItem
 - WidgetRegistry, 18
- addMenu
 - SwitchingBar, 13
- addSplitter
 - BlendSplitter, 7
- addWidget
 - BlendSplitter, 7
 - SwitchingBar, 13
- BlendSplitter, 5
 - addSplitter, 7
 - addWidget, 7
 - BlendSplitter, 7
 - expanderImage, 9
 - expanderSize, 9
 - insertSplitter, 8
 - insertWidget, 8
 - switchingBarHeight, 9
- expanderImage
 - BlendSplitter, 9
- expanderSize
 - BlendSplitter, 9
- getDefault
 - WidgetRegistry, 19
- getRegistry
 - WidgetRegistry, 19
- indexOf
 - WidgetRegistry, 19
- insertItem
 - WidgetRegistry, 19, 20
- insertSplitter
 - BlendSplitter, 8
- insertWidget
 - BlendSplitter, 8
- item
 - WidgetRegistry, 20
- name
 - RegistryItem, 10
- populateBar
 - RegistryItem, 10
- registryChanged
 - WidgetRegistry, 20
- RegistryItem, 9
 - name, 10
 - populateBar, 10
 - RegistryItem, 10
 - widget, 11
- removeItem
 - WidgetRegistry, 20, 21
- setDefault
 - WidgetRegistry, 21
- size
 - WidgetRegistry, 21
- SwitchingBar, 11
 - addMenu, 13
 - addWidget, 13
- switchingBarHeight
 - BlendSplitter, 9
- SwitchingWidget, 13
 - SwitchingWidget, 15
- widget
 - RegistryItem, 11
- WidgetRegistry, 15
 - addItem, 18
 - getDefault, 19
 - getRegistry, 19
 - indexOf, 19
 - insertItem, 19, 20
 - item, 20
 - registryChanged, 20
 - removeItem, 20, 21
 - setDefault, 21
 - size, 21