

# SFML Gui

0.1

Generated by Doxygen 1.5.6

Fri Jan 2 12:07:59 2009



# Contents

<b>1</b>	<b>Namespace Index</b>	<b>1</b>
1.1	Namespace List . . . . .	1
<b>2</b>	<b>Class Index</b>	<b>3</b>
2.1	Class Hierarchy . . . . .	3
<b>3</b>	<b>Class Index</b>	<b>5</b>
3.1	Class List . . . . .	5
<b>4</b>	<b>File Index</b>	<b>7</b>
4.1	File List . . . . .	7
<b>5</b>	<b>Namespace Documentation</b>	<b>9</b>
5.1	sfgui Namespace Reference . . . . .	9
5.1.1	Detailed Description . . . . .	9
5.1.2	Enumeration Type Documentation . . . . .	10
5.1.2.1	"@0 . . . . .	10
<b>6</b>	<b>Class Documentation</b>	<b>11</b>
6.1	sfgui::Button Class Reference . . . . .	11
6.1.1	Detailed Description . . . . .	12
6.1.2	Constructor & Destructor Documentation . . . . .	12
6.1.2.1	Button . . . . .	12
6.1.2.2	~Button . . . . .	12
6.1.3	Member Function Documentation . . . . .	12
6.1.3.1	updateTextPos . . . . .	12
6.1.3.2	getText . . . . .	12
6.1.3.3	SetText . . . . .	13
6.1.3.4	GetText . . . . .	13
6.1.3.5	SetTextColor . . . . .	13

6.1.3.6	GetTextColor	13
6.1.3.7	SetTextSize	13
6.1.3.8	GetTextSize	13
6.1.3.9	SetFont	13
6.1.3.10	GetFont	13
6.1.3.11	SetTextAlignment	13
6.1.3.12	SetTextMargin	13
6.1.3.13	SetTextLeftMargin	13
6.1.3.14	SetTextRightMargin	13
6.1.3.15	SetTextTopMargin	13
6.1.3.16	SetTextBottomMargin	13
6.1.3.17	SetPosition	14
6.1.3.18	Move	14
6.1.3.19	Show	14
6.1.4	Member Data Documentation	14
6.1.4.1	m_text	14
6.1.4.2	m_textAlignment	14
6.1.4.3	m_margin	14
6.2	sfgui::Margin Struct Reference	15
6.2.1	Detailed Description	15
6.2.2	Member Function Documentation	15
6.2.2.1	SetMargin	15
6.2.3	Member Data Documentation	15
6.2.3.1	Left	15
6.2.3.2	Right	15
6.2.3.3	Top	15
6.2.3.4	Bottom	16
6.3	sfgui::Object Class Reference	17
6.3.1	Detailed Description	18
6.3.2	Constructor & Destructor Documentation	18
6.3.2.1	Object	18
6.3.2.2	~Object	18
6.3.3	Member Function Documentation	18
6.3.3.1	SetBackground	18
6.3.3.2	SetBackground	18
6.3.3.3	SetBackground	18

6.3.3.4	SetBackground	18
6.3.3.5	CheckEvent	18
6.3.3.6	SetClickCallback	18
6.3.3.7	SetMouseHoverCallback	18
6.3.3.8	ManageMouse	19
6.3.3.9	Clicked	19
6.3.3.10	MouseHover	19
6.3.4	Member Data Documentation	19
6.3.4.1	m_clickCallback	19
6.3.4.2	m_mouseHoverCallback	19
6.3.4.3	m_parentRenderWindow	19
6.3.4.4	m_BackgroundImg	19
6.3.4.5	m_Event	19
6.4	sfgui::TextEdit Class Reference	20
6.4.1	Detailed Description	21
6.4.2	Constructor & Destructor Documentation	21
6.4.2.1	TextEdit	21
6.4.3	Member Function Documentation	21
6.4.3.1	textChanged	21
6.4.3.2	deactivated	21
6.4.3.3	activated	21
6.4.3.4	SetText	21
6.4.3.5	AddChar	21
6.4.3.6	DelChar	22
6.4.3.7	Activate	22
6.4.3.8	Deactivate	22
6.4.3.9	CheckEvent	22
6.4.3.10	SetTextChangedCallback	22
6.4.3.11	SetReturnPressedCallback	22
6.4.3.12	SetActivatedCallback	22
6.4.3.13	SetDeactivatedCallback	22
6.4.3.14	Show	22
6.4.4	Member Data Documentation	23
6.4.4.1	m_stdText	23
6.4.4.2	m_nbCharToShow	23
6.4.4.3	m_itemActive	23

6.4.4.4	<a href="#">m_textChangedCallback</a> . . . . .	23
6.4.4.5	<a href="#">m_deactivatedCallback</a> . . . . .	23
6.4.4.6	<a href="#">m_activatedCallback</a> . . . . .	23
6.4.4.7	<a href="#">m_returnPressedCallback</a> . . . . .	23
<b>7</b>	<b>File Documentation</b>	<b>25</b>
7.1	<a href="#">button.cpp</a> File Reference . . . . .	25
7.2	<a href="#">button.hpp</a> File Reference . . . . .	26
7.3	<a href="#">constantes.hpp</a> File Reference . . . . .	27
7.3.1	Detailed Description . . . . .	27
7.4	<a href="#">main.cpp</a> File Reference . . . . .	28
7.4.1	Function Documentation . . . . .	28
7.4.1.1	<a href="#">App</a> . . . . .	28
7.4.1.2	<a href="#">clickedCallBack</a> . . . . .	28
7.4.1.3	<a href="#">main</a> . . . . .	28
7.4.1.4	<a href="#">textChangeCallback</a> . . . . .	28
7.4.2	Variable Documentation . . . . .	28
7.4.2.1	<a href="#">App</a> . . . . .	28
7.5	<a href="#">margin.cpp</a> File Reference . . . . .	29
7.6	<a href="#">margin.hpp</a> File Reference . . . . .	30
7.6.1	Detailed Description . . . . .	30
7.7	<a href="#">object.cpp</a> File Reference . . . . .	31
7.7.1	Function Documentation . . . . .	31
7.7.1.1	<a href="#">SetBackground</a> . . . . .	31
7.8	<a href="#">object.hpp</a> File Reference . . . . .	32
7.9	<a href="#">textedit.cpp</a> File Reference . . . . .	33
7.10	<a href="#">textedit.hpp</a> File Reference . . . . .	34

# Chapter 1

## Namespace Index

### 1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

<a href="#">sfgui</a> (All graphics widgets ) . . . . .	9
---	---





# Chapter 2

## Class Index

### 2.1 Class Hierarchy

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

sfgui::Margin . . . . .	15
sfgui::Object . . . . .	17
sfgui::Button . . . . .	11
sfgui::TextEdit . . . . .	20



# Chapter 3

## Class Index

### 3.1 Class List

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

<a href="#">sfgui::Button</a> (A push button ) . . . . .	<a href="#">11</a>
<a href="#">sfgui::Margin</a> (Struct which contains margin values ) . . . . .	<a href="#">15</a>
<a href="#">sfgui::Object</a> (A simple graphic item ) . . . . .	<a href="#">17</a>
<a href="#">sfgui::TextEdit</a> (A single line text entry ) . . . . .	<a href="#">20</a>



# Chapter 4

## File Index

### 4.1 File List

Here is a list of all files with brief descriptions:

<a href="#">button.cpp</a>	25
<a href="#">button.hpp</a>	26
<a href="#">constantes.hpp</a> (Description not set )	27
<a href="#">main.cpp</a>	28
<a href="#">margin.cpp</a>	29
<a href="#">margin.hpp</a> (A simple struct for margin )	30
<a href="#">object.cpp</a>	31
<a href="#">object.hpp</a>	32
<a href="#">textedit.cpp</a>	33
<a href="#">textedit.hpp</a>	34



## Chapter 5

# Namespace Documentation

### 5.1 sfgui Namespace Reference

All graphics widgets.

#### Classes

- class [Button](#)  
*A push button.*
- struct [Margin](#)  
*Struct which contains margin values.*
- class [Object](#)  
*A simple graphic item.*
- class [TextEdit](#)  
*A single line text entry.*

#### Enumerations

- enum { [Left](#), [Right](#), [Center](#) }

#### 5.1.1 Detailed Description

All graphics widgets.

This namespace contains all the graphics widgets

## 5.1.2 Enumeration Type Documentation

### 5.1.2.1 anonymous enum

This enumeration defines the position of an item (such as text). It is used by widget to define autoposition (like [Button](#) which has `SetTextPosition...`)

**Enumerator:**

*Left* [Object](#) is on the left.

*Right* [Object](#) is on the right.

*Center* [Object](#) is centered.



## Chapter 6

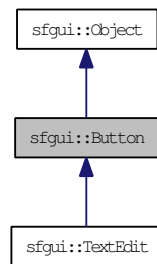
# Class Documentation

### 6.1 sfgui::Button Class Reference

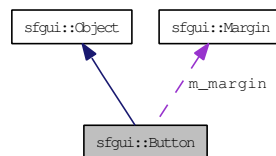
A push button.

```
#include <button.hpp>
```

Inheritance diagram for sfgui::Button:



Collaboration diagram for sfgui::Button:



### Public Member Functions

- [Button](#) (sf::RenderWindow \*parentWindow)
- [~Button](#) ()
- std::string [getText](#) ()
- void [SetText](#) (std::string text)
- std::string [GetText](#) ()
- void [SetTextColor](#) (sf::Color &)

- sf::Color [GetTextColor](#) ()
- void [SetTextSize](#) (float)
- float [GetTextSize](#) ()
- void [SetFont](#) (sf::Font &)
- sf::Font [GetFont](#) ()
- void [SetTextAlignment](#) (int)
- void [SetTextMargin](#) (float)
- void [SetTextLeftMargin](#) (float)
- void [SetTextRightMargin](#) (float)
- void [SetTextTopMargin](#) (float)
- void [SetTextBottomMargin](#) (float)
- void [SetPosition](#) (float x, float y)
- void [Move](#) (float x, float y)
- void [Show](#) ()

## Protected Member Functions

- void [updateTextPos](#) ()

## Protected Attributes

- sf::String [m\\_text](#)
- int [m\\_textAlignment](#)
- sfgui::Margin [m\\_margin](#)

## 6.1.1 Detailed Description

A push button.

This class represents a push button graphic item. It provides some signals (call some useful callbacks) like clicked, mouseOver...

## 6.1.2 Constructor & Destructor Documentation

### 6.1.2.1 sfgui::Button::Button (sf::RenderWindow \* *parentWindow*)

Create a [Button](#) on the parent window.

### 6.1.2.2 sfgui::Button::~~Button ()

## 6.1.3 Member Function Documentation

### 6.1.3.1 void sfgui::Button::updateTextPos () [protected]

### 6.1.3.2 std::string sfgui::Button::getText () [inline]

Get the button text

**6.1.3.3 void sfgui::Button::SetText (std::string *text*)**

Set the button text.

**6.1.3.4 std::string sfgui::Button::GetText ()****6.1.3.5 void sfgui::Button::SetTextColor (sf::Color & *color*)****6.1.3.6 sf::Color sfgui::Button::GetTextColor ()****6.1.3.7 void sfgui::Button::SetTextSize (float *size*)**

Set size of the text

**6.1.3.8 float sfgui::Button::GetTextSize ()****6.1.3.9 void sfgui::Button::SetTextFont (sf::Font & *font*)****6.1.3.10 sf::Font sfgui::Button::GetTextFont ()****6.1.3.11 void sfgui::Button::SetTextAlignment (int *al*)**

This set the position of the text on the button. You should use one of the following constant value sfgui::LEFT, sfgui::RIGHT, sfgui::CENTER

**6.1.3.12 void sfgui::Button::SetTextMargin (float *margin*)**

Set the global margin. The text will be spaced from the button each border by a number of pixels.

**6.1.3.13 void sfgui::Button::SetTextLeftMargin (float *margin*)**

Set the left margin. Text will be spaced from the left border of the button

**6.1.3.14 void sfgui::Button::SetTextRightMargin (float *margin*)**

Set the right margin. Text will be spaced form the right border of the button.

**6.1.3.15 void sfgui::Button::SetTextTopMargin (float *margin*)**

Set the top margin. Text will be spaced from the top border of the button.

**6.1.3.16 void sfgui::Button::SetTextBottomMargin (float *margin*)**

set the bottom margin. Text will be spaced from the bottom border of the button

**6.1.3.17 void sfgui::Button::SetPosition (float *x*, float *y*)**

Set the button position. Adjust the text position to keep the text in the choosen position (center, left, right...)

**6.1.3.18 void sfgui::Button::Move (float *x*, float *y*)**

Move the button and adjust the text position

**6.1.3.19 void sfgui::Button::Show ()**

Display the button on the parent window

Reimplemented in [sfgui::TextEdit](#).

**6.1.4 Member Data Documentation****6.1.4.1 sf::String sfgui::Button::m\_text** [protected]**6.1.4.2 int sfgui::Button::m\_textAlignment** [protected]**6.1.4.3 sfgui::Margin sfgui::Button::m\_margin** [protected]

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

- [button.hpp](#)
- [button.cpp](#)

## 6.2 sfgui::Margin Struct Reference

Struct which contains margin values.

```
#include <margin.hpp>
```

### Public Member Functions

- void [SetMargin](#) (float)

### Public Attributes

- float [Left](#)  
*Left margin value.*
- float [Right](#)  
*Right margin value.*
- float [Top](#)  
*Top margin value.*
- float [Bottom](#)  
*Bottom margin value.*

### 6.2.1 Detailed Description

Struct which contains margin values.

This struc contains the pixel spacing from each border of an object. The function SetMargin is provided for convenience as it set all the margin with the same value.

### 6.2.2 Member Function Documentation

#### 6.2.2.1 void sfgui::Margin::SetMargin (float *margin*)

### 6.2.3 Member Data Documentation

#### 6.2.3.1 float sfgui::Margin::Left

Left margin value.

#### 6.2.3.2 float sfgui::Margin::Right

Right margin value.

#### 6.2.3.3 float sfgui::Margin::Top

Top margin value.

#### 6.2.3.4 float sfgui::Margin::Bottom

Bottom margin value.

The documentation for this struct was generated from the following files:

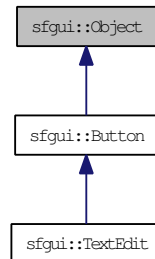
- [margin.hpp](#)
- [margin.cpp](#)

## 6.3 sfgui::Object Class Reference

A simple graphic item.

```
#include <object.hpp>
```

Inheritance diagram for sfgui::Object:



### Public Member Functions

- [Object](#) (sf::RenderWindow \*parentWindow)
- [~Object](#) ()
- void [SetBackground](#) (std::string)
- void [SetBackground](#) (sf::Image &)
- void [SetBackground](#) (sf::Color &)
- void [SetBackground](#) (sf::Color &, sf::Color &)
- void [CheckEvent](#) (sf::Event Event)
- void [SetClickCallback](#) (void(\*)())
- void [SetMouseHoverCallback](#) (void(\*)())
- void [ManageMouse](#) ()
- void [Clicked](#) ()
- void [MouseHover](#) ()

### Protected Attributes

- sf::RenderWindow \* [m\\_parentRenderWindow](#)  
*Pointer to the parent sf::RenderWindow.*
- sf::Image \* [m\\_BackgroundImg](#)  
*Curent background image.*
- sf::Event [m\\_Event](#)  
*Copy of the current sfml event.*

### Private Attributes

- void(\* [m\\_clickCallback](#) )()  
*Pointer to the click callback function.*
- void(\* [m\\_mouseHoverCallback](#) )()

### 6.3.1 Detailed Description

A simple graphic item.

This is a very simple object, which is rather useless. It's mostly present to be used as parent by other graphics items classes. This class create a `sf::Sprite` which represent a graphic item. There are some basic functions to set the general appearance like `setBackground`. This class can handle some common events which are possibly used by all widgets like clicks, mouse over...

### 6.3.2 Constructor & Destructor Documentation

#### 6.3.2.1 `sfgui::Object::Object (sf::RenderWindow * parentWindow)`

Test de classe

Construct a simple graphic object

#### 6.3.2.2 `sfgui::Object::~~Object ()`

### 6.3.3 Member Function Documentation

#### 6.3.3.1 `void sfgui::Object::SetBackground (std::string path)`

Set a background from an image path

#### 6.3.3.2 `void sfgui::Object::SetBackground (sf::Image & Img)`

Set a background from an existing `sf::Image`

#### 6.3.3.3 `void sfgui::Object::SetBackground (sf::Color & color)`

Set a background color

#### 6.3.3.4 `void sfgui::Object::SetBackground (sf::Color &, sf::Color &)`

#### 6.3.3.5 `void sfgui::Object::CheckEvent (sf::Event Event)`

Callbacks

Call the callbacks functions if needed on the given Event.

Reimplemented in [sfgui::TextEdit](#).

#### 6.3.3.6 `void sfgui::Object::SetClickCallback (void(*)() clickCallBack)`

Set a pointer to the callback function called when the user click on this object

#### 6.3.3.7 `void sfgui::Object::SetMouseHoverCallback (void(*)() mouseHoverCallback)`

Set a pointer to the callback function called when the mouse is hover the button.



**6.3.3.8 void sfgui::Object::ManageMouse ()**

Manage the mouse events.

**6.3.3.9 void sfgui::Object::Clicked ()**

Call the click callback function. It's automatically called by checkEvent, but you can also call it yourself if needed.

**6.3.3.10 void sfgui::Object::MouseHover ()****6.3.4 Member Data Documentation****6.3.4.1 void(\* sfgui::Object::m\_clickCallback)() [private]**

Pointer to the click callback function.

**6.3.4.2 void(\* sfgui::Object::m\_mouseHoverCallback)() [private]**

Pointer to the callback function called when mouse is hover the button

**6.3.4.3 sf::RenderWindow\* sfgui::Object::m\_parentRenderWindow [protected]**

Pointer to the parent sf::RenderWindow.

**6.3.4.4 sf::Image\* sfgui::Object::m\_BackgroundImg [protected]**

Curent background image.

**6.3.4.5 sf::Event sfgui::Object::m\_Event [protected]**

Copy of the current sfml event.

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

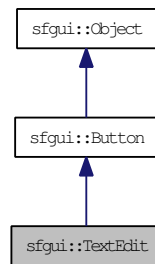
- [object.hpp](#)
- [object.cpp](#)

## 6.4 sfgui::TextEdit Class Reference

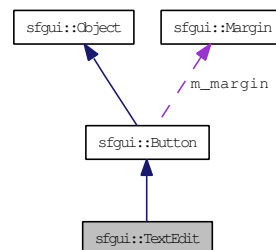
A single line text entry.

```
#include <textedit.hpp>
```

Inheritance diagram for sfgui::TextEdit:



Collaboration diagram for sfgui::TextEdit:



### Public Member Functions

- [TextEdit](#) (sf::RenderWindow \*parentWindow)
- void [SetText](#) (std::string &)
- void [AddChar](#) (char)
- void [DelChar](#) (int)
- void [Activate](#) ()
- void [Deactivate](#) ()
- void [CheckEvent](#) (sf::Event Event)
- void [SetTextChangedCallback](#) (void(\*) (std::string &))
- void [SetReturnPressedCallback](#) (void(\*) ())
- void [SetActivatedCallback](#) (void(\*) ())
- void [SetDeactivatedCallback](#) (void(\*) ())
- void [Show](#) ()

### Private Member Functions

- void [textChanged](#) ()
- void [deactivated](#) ()
- void [activated](#) ()

## Private Attributes

- std::string [m\\_stdText](#)
- int [m\\_nbCharToShow](#)
- bool [m\\_itemActive](#)
- void(\* [m\\_textChangedCallback](#) )(std::string &)
- void(\* [m\\_deactivatedCallback](#) )()
- void(\* [m\\_activatedCallback](#) )()
- void(\* [m\\_returnPressedCallback](#) )()

### 6.4.1 Detailed Description

A single line text entry.

This class provides a single line text entry widget. Thanks to this class, the user can enter text directly in the application, and usefull callbacks are provided, like [textChangedCallback](#), [returnPressedCallback](#).

#### Small tutorial

If you want to enter text, you must have the item activated. You can activate it by calling the [Activate\(\)](#) function (when user click on the [TextEdit](#), this function is automatically called).

If you want to stop typing text in the widget, you must call the [Deactivate\(\)](#) function.

**Important :** you must call the [CheckEvent](#) function in the event loop. If you don't, the widget will just be shown, but won't work

### 6.4.2 Constructor & Destructor Documentation

#### 6.4.2.1 sfgui::TextEdit::TextEdit (sf::RenderWindow \* *parentWindow*)

### 6.4.3 Member Function Documentation

#### 6.4.3.1 void sfgui::TextEdit::textChanged () [private]

If the text is modified, this function is called. It updates the sfString on the screen, and it call the [textChanged](#) callback (if exists)

#### 6.4.3.2 void sfgui::TextEdit::deactivated () [private]

#### 6.4.3.3 void sfgui::TextEdit::activated () [private]

#### 6.4.3.4 void sfgui::TextEdit::SetText (std::string & *text*)

Set the text on the [TextEdit](#)

#### 6.4.3.5 void sfgui::TextEdit::AddChar (char *ch*)

Add one char at the end of the current string

**6.4.3.6 void sfgui::TextEdit::DelChar (int *pos*)**

Delete one char from position *pos*

**6.4.3.7 void sfgui::TextEdit::Activate ()**

Activate the [TextEdit](#). When the [TextEdit](#) is activated, all pressed keys are added to the [TextEdit](#) string

**6.4.3.8 void sfgui::TextEdit::Deactivate ()**

Deactivate the Textedit. Key you press will no longer be added to the [TextEdit](#) (until [Activate\(\)](#) is called)

**6.4.3.9 void sfgui::TextEdit::CheckEvent (sf::Event *Event*)**

Callbacks

Call the callbacks functions if needed on the given Event.

Reimplemented from [sfgui::Object](#).

**6.4.3.10 void sfgui::TextEdit::SetTextChangedCallback (void(\*) (std::string &)  
*textChangedCallback*)**

Set the callback called when the text is modified (add new char, delete one...)

**6.4.3.11 void sfgui::TextEdit::SetReturnPressedCallback (void(\*)() *returnPressedCallback*)**

Set the callback called when the return key is pressed (it can be used to know when the user finished his text entry for instance)

**6.4.3.12 void sfgui::TextEdit::SetActivatedCallback (void(\*)() *activatedCallback*)**

Set the callback called when the [TextEdit](#) is activated (ie has focus).

**6.4.3.13 void sfgui::TextEdit::SetDeactivatedCallback (void(\*)() *deactivatedCallback*)**

Set the callback called when the [TextEdit](#) is deactivated (ie hasn't focus).

**6.4.3.14 void sfgui::TextEdit::Show ()**

Show the [TextEdit](#) on the parent window

Reimplemented from [sfgui::Button](#).

## 6.4.4 Member Data Documentation

**6.4.4.1** `std::string sfgui::TextEdit::m_stdText` [private]

**6.4.4.2** `int sfgui::TextEdit::m_nbCharToShow` [private]

**6.4.4.3** `bool sfgui::TextEdit::m_itemActive` [private]

If true, user can enter text (textedit has focus), if false, it is disabled

**6.4.4.4** `void(* sfgui::TextEdit::m_textChangedCallback)(std::string &)` [private]

**6.4.4.5** `void(* sfgui::TextEdit::m_deactivatedCallback)()` [private]

**6.4.4.6** `void(* sfgui::TextEdit::m_activatedCallback)()` [private]

**6.4.4.7** `void(* sfgui::TextEdit::m_returnPressedCallback)()` [private]

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

- [textedit.hpp](#)
- [textedit.cpp](#)



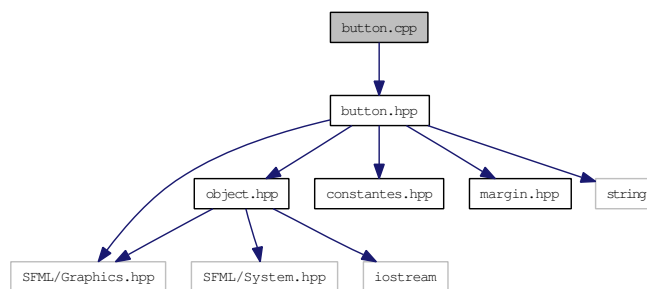
## Chapter 7

# File Documentation

### 7.1 button.cpp File Reference

```
#include "button.hpp"
```

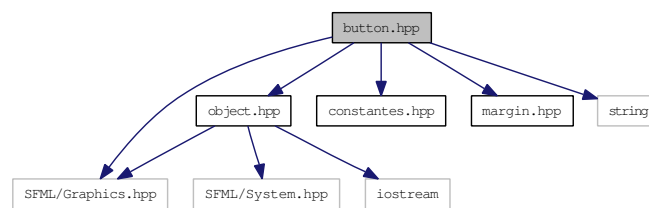
Include dependency graph for button.cpp:



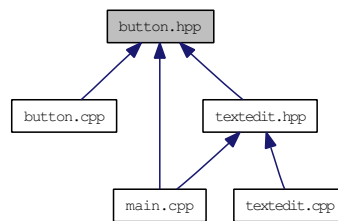
## 7.2 button.hpp File Reference

```
#include <SFML/Graphics.hpp>
#include "object.hpp"
#include "constantes.hpp"
#include "margin.hpp"
#include <string>
```

Include dependency graph for button.hpp:



This graph shows which files directly or indirectly include this file:



### Namespaces

- namespace [sfgui](#)

### Classes

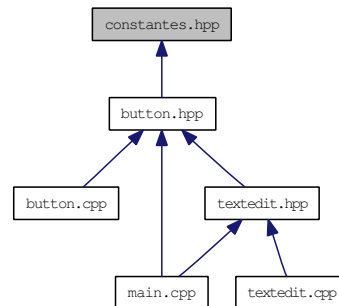
- class [sfgui::Button](#)  
*A push button.*



## 7.3 constantes.hpp File Reference

Description not set.

This graph shows which files directly or indirectly include this file:



### Namespaces

- namespace `sfgui`

### Enumerations

- enum { `sfgui::Left`, `sfgui::Right`, `sfgui::Center` }

#### 7.3.1 Detailed Description

Description not set.

**Author:**

TANGUY Arnaud <[arn.tanguy@gmail.com](mailto:arn.tanguy@gmail.com)>

**Date:**

2009

**Version:**

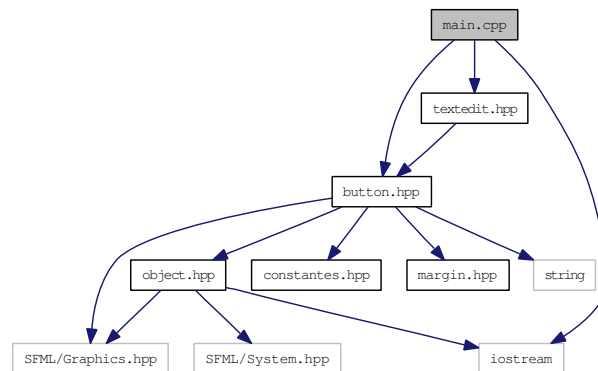
0.1

The big description for doxygen

## 7.4 main.cpp File Reference

```
#include "button.hpp"
#include "textedit.hpp"
#include <iostream>
```

Include dependency graph for main.cpp:



### Functions

- `sf::RenderWindow` [App](#) (`sf::VideoMode(800, 600, 32), "SFML GUI"`)
- `void` [clickedCallback](#) ()
- `void` [textChangeCallback](#) (`std::string &sr`)
- `int` [main](#) ()

### Variables

- `sfgui::Button` [Sprite](#) & [App](#)

#### 7.4.1 Function Documentation

**7.4.1.1** `sf::RenderWindow` [App](#) (`sf::VideoMode800, 600, 32, "SFML GUI"`)

**7.4.1.2** `void` [clickedCallback](#) ()

**7.4.1.3** `int` [main](#) ()

**7.4.1.4** `void` [textChangeCallback](#) (`std::string &sr`)

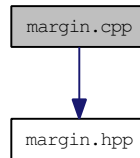
#### 7.4.2 Variable Documentation

**7.4.2.1** `sfgui::Button` [Sprite](#)& [App](#)

## 7.5 margin.cpp File Reference

```
#include "margin.hpp"
```

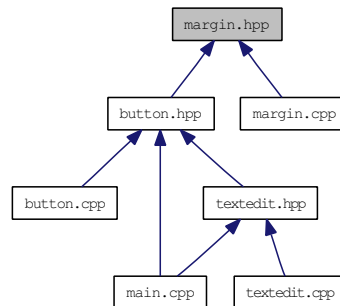
Include dependency graph for margin.cpp:



## 7.6 margin.hpp File Reference

A simple struct for margin.

This graph shows which files directly or indirectly include this file:



### Namespaces

- namespace [sfgui](#)

### Classes

- struct [sfgui::Margin](#)  
*Struct which contains margin values.*

### 7.6.1 Detailed Description

A simple struct for margin.

#### Author:

TANGUY Arnaud <[arn.tanguy@gmail.com](mailto:arn.tanguy@gmail.com)>

#### Date:

2009

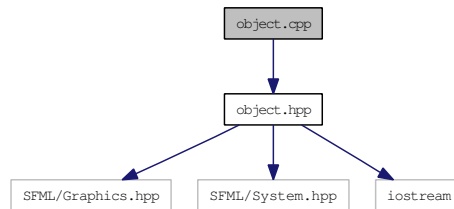
#### Version:

0.1

## 7.7 object.cpp File Reference

```
#include "object.hpp"
```

Include dependency graph for object.cpp:



### Functions

- void [SetBackground](#) (sf::Color &, sf::Color &)

#### 7.7.1 Function Documentation

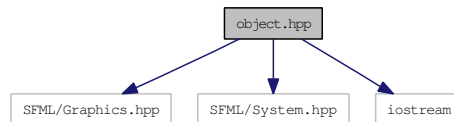
##### 7.7.1.1 void SetBackground (sf::Color &, sf::Color &)

Shaded of between two colors. Not implemented yet.

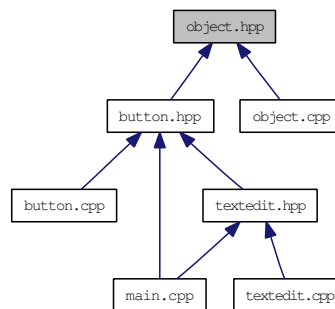
## 7.8 object.hpp File Reference

```
#include <SFML/Graphics.hpp>
#include <SFML/System.hpp>
#include <iostream>
```

Include dependency graph for object.hpp:



This graph shows which files directly or indirectly include this file:



### Namespaces

- namespace [sfgui](#)

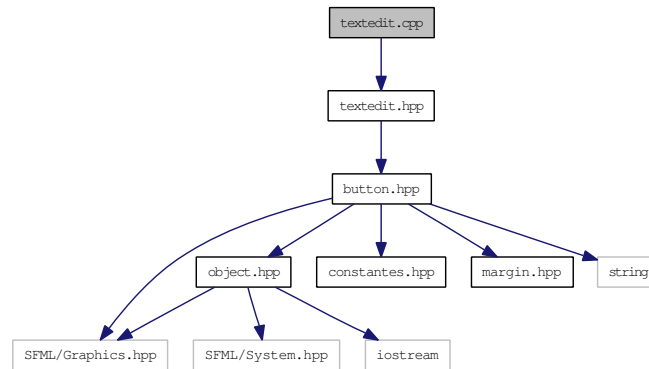
### Classes

- class [sfgui::Object](#)  
*A simple graphic item.*

## 7.9 textedit.cpp File Reference

```
#include "textedit.hpp"
```

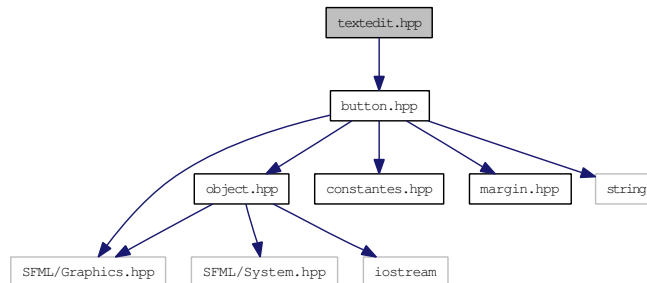
Include dependency graph for textedit.cpp:



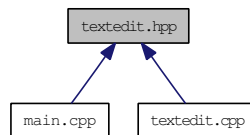
## 7.10 textedit.hpp File Reference

```
#include "button.hpp"
```

Include dependency graph for textedit.hpp:



This graph shows which files directly or indirectly include this file:



### Namespaces

- namespace [sfgui](#)

### Classes

- class [sfgui::TextEdit](#)  
*A single line text entry.*



# Index

- ~Button
  - sfgui::Button, [12](#)
- ~Object
  - sfgui::Object, [18](#)
- Activate
  - sfgui::TextEdit, [22](#)
- activated
  - sfgui::TextEdit, [21](#)
- AddChar
  - sfgui::TextEdit, [21](#)
- App
  - main.cpp, [28](#)
- Bottom
  - sfgui::Margin, [15](#)
- Button
  - sfgui::Button, [12](#)
- button.cpp, [25](#)
- button.hpp, [26](#)
- Center
  - sfgui, [10](#)
- CheckEvent
  - sfgui::Object, [18](#)
  - sfgui::TextEdit, [22](#)
- Clicked
  - sfgui::Object, [19](#)
- clickedCallBack
  - main.cpp, [28](#)
- constantes.hpp, [27](#)
- Deactivate
  - sfgui::TextEdit, [22](#)
- deactivated
  - sfgui::TextEdit, [21](#)
- DelChar
  - sfgui::TextEdit, [21](#)
- GetText
  - sfgui::Button, [13](#)
- getText
  - sfgui::Button, [12](#)
- GetTextColor
  - sfgui::Button, [13](#)
- GetTextFont
  - sfgui::Button, [13](#)
- GetTextSize
  - sfgui::Button, [13](#)
- Left
  - sfgui, [10](#)
  - sfgui::Margin, [15](#)
- m\_activatedCallback
  - sfgui::TextEdit, [23](#)
- m\_BackgroundImg
  - sfgui::Object, [19](#)
- m\_clickCallback
  - sfgui::Object, [19](#)
- m\_deactivatedCallback
  - sfgui::TextEdit, [23](#)
- m\_Event
  - sfgui::Object, [19](#)
- m\_itemActive
  - sfgui::TextEdit, [23](#)
- m\_margin
  - sfgui::Button, [14](#)
- m\_mouseHoverCallback
  - sfgui::Object, [19](#)
- m\_nbCharToShow
  - sfgui::TextEdit, [23](#)
- m\_parentRenderWindow
  - sfgui::Object, [19](#)
- m\_returnPressedCallback
  - sfgui::TextEdit, [23](#)
- m\_stdText
  - sfgui::TextEdit, [23](#)
- m\_text
  - sfgui::Button, [14](#)
- m\_textAlignment
  - sfgui::Button, [14](#)
- m\_textChangedCallback
  - sfgui::TextEdit, [23](#)
- main
  - main.cpp, [28](#)
- main.cpp, [28](#)
  - App, [28](#)
  - clickedCallBack, [28](#)
  - main, [28](#)
  - textChangeCallback, [28](#)

- ManageMouse
  - sfgui::Object, 18
- margin.cpp, 29
- margin.hpp, 30
- MouseHover
  - sfgui::Object, 19
- Move
  - sfgui::Button, 14
- Object
  - sfgui::Object, 18
- object.cpp, 31
  - SetBackground, 31
- object.hpp, 32
- Right
  - sfgui, 10
  - sfgui::Margin, 15
- SetActivatedCallback
  - sfgui::TextEdit, 22
- SetBackground
  - object.cpp, 31
  - sfgui::Object, 18
- SetClickCallback
  - sfgui::Object, 18
- SetDeactivatedCallback
  - sfgui::TextEdit, 22
- SetMargin
  - sfgui::Margin, 15
- SetMouseHoverCallback
  - sfgui::Object, 18
- SetPosition
  - sfgui::Button, 13
- SetReturnPressedCallback
  - sfgui::TextEdit, 22
- SetText
  - sfgui::Button, 12
  - sfgui::TextEdit, 21
- SetTextAlignment
  - sfgui::Button, 13
- SetTextBottomMargin
  - sfgui::Button, 13
- SetTextChangedCallback
  - sfgui::TextEdit, 22
- SetTextColor
  - sfgui::Button, 13
- SetTextFont
  - sfgui::Button, 13
- SetTextLeftMargin
  - sfgui::Button, 13
- SetTextMargin
  - sfgui::Button, 13
- SetTextRightMargin
  - sfgui::Button, 13
- SetTextSize
  - sfgui::Button, 13
- SetTextTopMargin
  - sfgui::Button, 13
- sfgui, 9
  - Center, 10
  - Left, 10
  - Right, 10
- sfgui::Button, 11
  - ~Button, 12
  - Button, 12
  - GetText, 13
  - getText, 12
  - GetTextColor, 13
  - GetTextFont, 13
  - GetTextSize, 13
  - m\_margin, 14
  - m\_text, 14
  - m\_textAlignment, 14
  - Move, 14
  - SetPosition, 13
  - SetText, 12
  - SetTextAlignment, 13
  - SetTextBottomMargin, 13
  - SetTextColor, 13
  - SetTextFont, 13
  - SetTextLeftMargin, 13
  - SetTextMargin, 13
  - SetTextRightMargin, 13
  - SetTextSize, 13
  - SetTextTopMargin, 13
  - Show, 14
  - updateTextPos, 12
- sfgui::Margin, 15
  - Bottom, 15
  - Left, 15
  - Right, 15
  - SetMargin, 15
  - Top, 15
- sfgui::Object, 17
  - ~Object, 18
  - CheckEvent, 18
  - Clicked, 19
  - m\_BackgroundImg, 19
  - m\_clickCallback, 19
  - m\_Event, 19
  - m\_mouseHoverCallback, 19
  - m\_parentRenderWindow, 19
  - ManageMouse, 18
  - MouseHover, 19
  - Object, 18
  - SetBackground, 18
  - SetClickCallback, 18

- SetMouseHoverCallback, [18](#)
- sfgui::TextEdit, [20](#)
  - Activate, [22](#)
  - activated, [21](#)
  - AddChar, [21](#)
  - CheckEvent, [22](#)
  - Deactivate, [22](#)
  - deactivated, [21](#)
  - DelChar, [21](#)
  - m\_activatedCallback, [23](#)
  - m\_deactivatedCallback, [23](#)
  - m\_itemActive, [23](#)
  - m\_nbCharToShow, [23](#)
  - m\_returnPressedCallback, [23](#)
  - m\_stdText, [23](#)
  - m\_textChangedCallback, [23](#)
  - SetActivatedCallback, [22](#)
  - SetDeactivatedCallback, [22](#)
  - SetReturnPressedCallback, [22](#)
  - SetText, [21](#)
  - SetTextChangedCallback, [22](#)
  - Show, [22](#)
  - textChanged, [21](#)
  - TextEdit, [21](#)
- Show
  - sfgui::Button, [14](#)
  - sfgui::TextEdit, [22](#)
- textChangeCallback
  - main.cpp, [28](#)
- textChanged
  - sfgui::TextEdit, [21](#)
- TextEdit
  - sfgui::TextEdit, [21](#)
- textedit.cpp, [33](#)
- textedit.hpp, [34](#)
- Top
  - sfgui::Margin, [15](#)
- updateTextPos
  - sfgui::Button, [12](#)