SFML

* [Main Page](http://docs.google.com/index.htm)
* [Modules](http://docs.google.com/modules.htm)
* [Classes](http://docs.google.com/annotated.htm)
* [Files](http://docs.google.com/files.htm)
* [Class List](http://docs.google.com/annotated.htm)
* [Class Index](http://docs.google.com/classes.htm)
* [Class Hierarchy](http://docs.google.com/hierarchy.htm)
* [Class Members](http://docs.google.com/functions.htm)
* **sf**
* [Sound](http://docs.google.com/classsf_1_1Sound.htm)

[Public Types](#_gjdgxs) | [Public Member Functions](#_30j0zll) | [Protected Attributes](#_1fob9te) | [List of all members](http://docs.google.com/classsf_1_1Sound-members.htm)

sf::Sound Class Reference

[Audio module](http://docs.google.com/group__audio.htm)

Regular sound that can be played in the audio environment. [More...](http://docs.google.com/classsf_1_1Sound.htm#details)

#include <[Sound.hpp](http://docs.google.com/Sound_8hpp_source.htm)>

Inheritance diagram for sf::Sound:



| Public Types | |
| --- | --- |
| enum | [Status](http://docs.google.com/classsf_1_1SoundSource.htm#ac43af72c98c077500b239bc75b812f03) {  [Stopped](http://docs.google.com/classsf_1_1SoundSource.htm#ac43af72c98c077500b239bc75b812f03adabb01e8aa85b2f54b344890addf764a),  [Paused](http://docs.google.com/classsf_1_1SoundSource.htm#ac43af72c98c077500b239bc75b812f03ac3ca1fcc0394267c9bdbe3dc0a8a7e41),  [Playing](http://docs.google.com/classsf_1_1SoundSource.htm#ac43af72c98c077500b239bc75b812f03af07bdea9f70ef7606dfc9f955beeee18)  } |
|  | Enumeration of the sound source states. [More...](http://docs.google.com/classsf_1_1SoundSource.htm#ac43af72c98c077500b239bc75b812f03) |
|  | |

| Public Member Functions | |
| --- | --- |
|  | [Sound](http://docs.google.com/classsf_1_1Sound.htm#a36ab74beaaa953d9879c933ddd246282) () |
|  | Default constructor. |
|  | |
|  | [Sound](http://docs.google.com/classsf_1_1Sound.htm#a3b1cfc19a856d4ff8c079ee41bb78e69) (const [SoundBuffer](http://docs.google.com/classsf_1_1SoundBuffer.htm) &buffer) |
|  | Construct the sound with a buffer. |
|  | |
|  | [Sound](http://docs.google.com/classsf_1_1Sound.htm#ae05eeed6377932694d86b3011be366c0) (const [Sound](http://docs.google.com/classsf_1_1Sound.htm) &copy) |
|  | Copy constructor. |
|  | |
|  | [~Sound](http://docs.google.com/classsf_1_1Sound.htm#ad0792c35310eba2dffd8489c80fad076) () |
|  | Destructor. |
|  | |
| void | [play](http://docs.google.com/classsf_1_1Sound.htm#a2953ffe632536e72e696fd880ced2532) () |
|  | Start or resume playing the sound. |
|  | |
| void | [pause](http://docs.google.com/classsf_1_1Sound.htm#a5eeb25815bfa8cdc4a6cc000b7b19ad5) () |
|  | Pause the sound. |
|  | |
| void | [stop](http://docs.google.com/classsf_1_1Sound.htm#aa9c91c34f7c6d344d5ee9b997511f754) () |
|  | stop playing the sound |
|  | |
| void | [setBuffer](http://docs.google.com/classsf_1_1Sound.htm#a8b395e9713d0efa48a18628c8ec1972e) (const [SoundBuffer](http://docs.google.com/classsf_1_1SoundBuffer.htm) &buffer) |
|  | Set the source buffer containing the audio data to play. |
|  | |
| void | [setLoop](http://docs.google.com/classsf_1_1Sound.htm#af23ab4f78f975bbabac031102321612b) (bool loop) |
|  | Set whether or not the sound should loop after reaching the end. |
|  | |
| void | [setPlayingOffset](http://docs.google.com/classsf_1_1Sound.htm#ab905677846558042022dd6ab15cddff0) ([Time](http://docs.google.com/classsf_1_1Time.htm) timeOffset) |
|  | Change the current playing position of the sound. |
|  | |
| const [SoundBuffer](http://docs.google.com/classsf_1_1SoundBuffer.htm) \* | [getBuffer](http://docs.google.com/classsf_1_1Sound.htm#a5f2f1ec6603f73625f393f0e9ab64476) () const |
|  | Get the audio buffer attached to the sound. |
|  | |
| bool | [getLoop](http://docs.google.com/classsf_1_1Sound.htm#a603fbd32b519ae1680c40d8ddfc7ea51) () const |
|  | Tell whether or not the sound is in loop mode. |
|  | |
| [Time](http://docs.google.com/classsf_1_1Time.htm) | [getPlayingOffset](http://docs.google.com/classsf_1_1Sound.htm#aeca741a3442d9934369a986ad40f19c0) () const |
|  | Get the current playing position of the sound. |
|  | |
| [Status](http://docs.google.com/classsf_1_1SoundSource.htm#ac43af72c98c077500b239bc75b812f03) | [getStatus](http://docs.google.com/classsf_1_1Sound.htm#ae8b4084ff8f460b7a2bf9d39e846dab9) () const |
|  | Get the current status of the sound (stopped, paused, playing) |
|  | |
| [Sound](http://docs.google.com/classsf_1_1Sound.htm) & | [operator=](http://docs.google.com/classsf_1_1Sound.htm#a08c64c9c1dabeebc59fbf2540d81d4dd) (const [Sound](http://docs.google.com/classsf_1_1Sound.htm) &right) |
|  | Overload of assignment operator. |
|  | |
| void | [resetBuffer](http://docs.google.com/classsf_1_1Sound.htm#acb7289d45e06fb76b8292ac84beb82a7) () |
|  | Reset the internal buffer of the sound. |
|  | |
| void | [setPitch](http://docs.google.com/classsf_1_1SoundSource.htm#a72a13695ed48b7f7b55e7cd4431f4bb6) (float pitch) |
|  | Set the pitch of the sound. |
|  | |
| void | [setVolume](http://docs.google.com/classsf_1_1SoundSource.htm#a2f192f2b49fb8e2b82f3498d3663fcc2) (float volume) |
|  | Set the volume of the sound. |
|  | |
| void | [setPosition](http://docs.google.com/classsf_1_1SoundSource.htm#a0480257ea25d986eba6cc3c1a6f8d7c2) (float x, float y, float z) |
|  | Set the 3D position of the sound in the audio scene. |
|  | |
| void | [setPosition](http://docs.google.com/classsf_1_1SoundSource.htm#a17ba9ed01925395652181a7b2a7d3aef) (const [Vector3f](http://docs.google.com/classsf_1_1Vector3.htm) &position) |
|  | Set the 3D position of the sound in the audio scene. |
|  | |
| void | [setRelativeToListener](http://docs.google.com/classsf_1_1SoundSource.htm#ac478a8b813faf7dd575635b102081d0d) (bool relative) |
|  | Make the sound's position relative to the listener or absolute. |
|  | |
| void | [setMinDistance](http://docs.google.com/classsf_1_1SoundSource.htm#a75bbc2c34addc8b25a14edb908508afe) (float distance) |
|  | Set the minimum distance of the sound. |
|  | |
| void | [setAttenuation](http://docs.google.com/classsf_1_1SoundSource.htm#aa2adff44cd2f8b4e3c7315d7c2a45626) (float attenuation) |
|  | Set the attenuation factor of the sound. |
|  | |
| float | [getPitch](http://docs.google.com/classsf_1_1SoundSource.htm#aedad6aff442aeb6dcd267befd4fdbb59) () const |
|  | Get the pitch of the sound. |
|  | |
| float | [getVolume](http://docs.google.com/classsf_1_1SoundSource.htm#aafb0558fce9cbebfc6828d932cbcce2f) () const |
|  | Get the volume of the sound. |
|  | |
| [Vector3f](http://docs.google.com/classsf_1_1Vector3.htm) | [getPosition](http://docs.google.com/classsf_1_1SoundSource.htm#a4c3bc60286f488aaf2941ab76476eebc) () const |
|  | Get the 3D position of the sound in the audio scene. |
|  | |
| bool | [isRelativeToListener](http://docs.google.com/classsf_1_1SoundSource.htm#a5cb9107e1c47f65ab82c4885436061ef) () const |
|  | Tell whether the sound's position is relative to the listener or is absolute. |
|  | |
| float | [getMinDistance](http://docs.google.com/classsf_1_1SoundSource.htm#a3379b9f7a0f0e31ab9a4e5fa1762986e) () const |
|  | Get the minimum distance of the sound. |
|  | |
| float | [getAttenuation](http://docs.google.com/classsf_1_1SoundSource.htm#ac5f5ffef8930bb573f43d47cbc779bf6) () const |
|  | Get the attenuation factor of the sound. |
|  | |

| Protected Attributes | |
| --- | --- |
| unsigned int | [m\_source](http://docs.google.com/classsf_1_1SoundSource.htm#a0223cef4b1c587e6e1e17b4c92c4479c) |
|  | OpenAL source identifier. |
|  | |

## Detailed Description

Regular sound that can be played in the audio environment.

[sf::Sound](http://docs.google.com/classsf_1_1Sound.htm) is the class to use to play sounds.

It provides:

* Control (play, pause, stop)
* Ability to modify output parameters in real-time (pitch, volume, ...)
* 3D spatial features (position, attenuation, ...).

[sf::Sound](http://docs.google.com/classsf_1_1Sound.htm) is perfect for playing short sounds that can fit in memory and require no latency, like foot steps or gun shots. For longer sounds, like background musics or long speeches, rather see [sf::Music](http://docs.google.com/classsf_1_1Music.htm) (which is based on streaming).

In order to work, a sound must be given a buffer of audio data to play. Audio data (samples) is stored in [sf::SoundBuffer](http://docs.google.com/classsf_1_1SoundBuffer.htm), and attached to a sound with the [setBuffer()](http://docs.google.com/classsf_1_1Sound.htm#a8b395e9713d0efa48a18628c8ec1972e) function. The buffer object attached to a sound must remain alive as long as the sound uses it. Note that multiple sounds can use the same sound buffer at the same time.

Usage example:

[sf::SoundBuffer](http://docs.google.com/classsf_1_1SoundBuffer.htm) buffer;

buffer.[loadFromFile](http://docs.google.com/classsf_1_1SoundBuffer.htm#a2be6a8025c97eb622a7dff6cf2594394)("sound.wav");

[sf::Sound](http://docs.google.com/classsf_1_1Sound.htm) sound;

sound.[setBuffer](http://docs.google.com/classsf_1_1Sound.htm#a8b395e9713d0efa48a18628c8ec1972e)(buffer);

sound.[play](http://docs.google.com/classsf_1_1Sound.htm#a2953ffe632536e72e696fd880ced2532)();

See Also[sf::SoundBuffer](http://docs.google.com/classsf_1_1SoundBuffer.htm), [sf::Music](http://docs.google.com/classsf_1_1Music.htm)

Definition at line [45](http://docs.google.com/Sound_8hpp_source.htm#l00045) of file [Sound.hpp](http://docs.google.com/Sound_8hpp_source.htm).

## Member Enumeration Documentation

| | enum [sf::SoundSource::Status](http://docs.google.com/classsf_1_1SoundSource.htm#ac43af72c98c077500b239bc75b812f03) | | --- | | inherited |
| --- | --- | --- |

Enumeration of the sound source states.

**Enumerator:**

| *Stopped* | [Sound](http://docs.google.com/classsf_1_1Sound.htm) is not playing. |
| --- | --- |
| *Paused* | [Sound](http://docs.google.com/classsf_1_1Sound.htm) is paused. |
| *Playing* | [Sound](http://docs.google.com/classsf_1_1Sound.htm) is playing. |

Definition at line [49](http://docs.google.com/SoundSource_8hpp_source.htm#l00049) of file [SoundSource.hpp](http://docs.google.com/SoundSource_8hpp_source.htm).

## Constructor & Destructor Documentation

| sf::Sound::Sound | ( |  | ) |  |
| --- | --- | --- | --- | --- |

Default constructor.

| | sf::Sound::Sound | ( | const [SoundBuffer](http://docs.google.com/classsf_1_1SoundBuffer.htm) & | *buffer* | ) |  | | --- | --- | --- | --- | --- | --- | | explicit |
| --- | --- | --- | --- | --- | --- | --- | --- |

Construct the sound with a buffer.

Parameters

| buffer | [Sound](http://docs.google.com/classsf_1_1Sound.htm) buffer containing the audio data to play with the sound |
| --- | --- |

| sf::Sound::Sound | ( | const [Sound](http://docs.google.com/classsf_1_1Sound.htm) & | *copy* | ) |  |
| --- | --- | --- | --- | --- | --- |

Copy constructor.

Parameters

| copy | Instance to copy |
| --- | --- |

| sf::Sound::~Sound | ( |  | ) |  |
| --- | --- | --- | --- | --- |

Destructor.

## Member Function Documentation

| | float sf::SoundSource::getAttenuation | ( |  | ) | const | | --- | --- | --- | --- | --- | | inherited |
| --- | --- | --- | --- | --- | --- | --- |

Get the attenuation factor of the sound.

ReturnsAttenuation factor of the sound See Also[setAttenuation](http://docs.google.com/classsf_1_1SoundSource.htm#aa2adff44cd2f8b4e3c7315d7c2a45626), [getMinDistance](http://docs.google.com/classsf_1_1SoundSource.htm#a3379b9f7a0f0e31ab9a4e5fa1762986e)

| const [SoundBuffer](http://docs.google.com/classsf_1_1SoundBuffer.htm)\* sf::Sound::getBuffer | ( |  | ) | const |
| --- | --- | --- | --- | --- |

Get the audio buffer attached to the sound.

Returns[Sound](http://docs.google.com/classsf_1_1Sound.htm) buffer attached to the sound (can be NULL)

| bool sf::Sound::getLoop | ( |  | ) | const |
| --- | --- | --- | --- | --- |

Tell whether or not the sound is in loop mode.

ReturnsTrue if the sound is looping, false otherwise See Also[setLoop](http://docs.google.com/classsf_1_1Sound.htm#af23ab4f78f975bbabac031102321612b)

| | float sf::SoundSource::getMinDistance | ( |  | ) | const | | --- | --- | --- | --- | --- | | inherited |
| --- | --- | --- | --- | --- | --- | --- |

Get the minimum distance of the sound.

ReturnsMinimum distance of the sound See Also[setMinDistance](http://docs.google.com/classsf_1_1SoundSource.htm#a75bbc2c34addc8b25a14edb908508afe), [getAttenuation](http://docs.google.com/classsf_1_1SoundSource.htm#ac5f5ffef8930bb573f43d47cbc779bf6)

| | float sf::SoundSource::getPitch | ( |  | ) | const | | --- | --- | --- | --- | --- | | inherited |
| --- | --- | --- | --- | --- | --- | --- |

Get the pitch of the sound.

ReturnsPitch of the sound See Also[setPitch](http://docs.google.com/classsf_1_1SoundSource.htm#a72a13695ed48b7f7b55e7cd4431f4bb6)

| [Time](http://docs.google.com/classsf_1_1Time.htm) sf::Sound::getPlayingOffset | ( |  | ) | const |
| --- | --- | --- | --- | --- |

Get the current playing position of the sound.

ReturnsCurrent playing position, from the beginning of the sound See Also[setPlayingOffset](http://docs.google.com/classsf_1_1Sound.htm#ab905677846558042022dd6ab15cddff0)

| | [Vector3f](http://docs.google.com/classsf_1_1Vector3.htm) sf::SoundSource::getPosition | ( |  | ) | const | | --- | --- | --- | --- | --- | | inherited |
| --- | --- | --- | --- | --- | --- | --- |

Get the 3D position of the sound in the audio scene.

ReturnsPosition of the sound See Also[setPosition](http://docs.google.com/classsf_1_1SoundSource.htm#a0480257ea25d986eba6cc3c1a6f8d7c2)

| [Status](http://docs.google.com/classsf_1_1SoundSource.htm#ac43af72c98c077500b239bc75b812f03) sf::Sound::getStatus | ( |  | ) | const |
| --- | --- | --- | --- | --- |

Get the current status of the sound (stopped, paused, playing)

ReturnsCurrent status of the sound

| | float sf::SoundSource::getVolume | ( |  | ) | const | | --- | --- | --- | --- | --- | | inherited |
| --- | --- | --- | --- | --- | --- | --- |

Get the volume of the sound.

ReturnsVolume of the sound, in the range [0, 100] See Also[setVolume](http://docs.google.com/classsf_1_1SoundSource.htm#a2f192f2b49fb8e2b82f3498d3663fcc2)

| | bool sf::SoundSource::isRelativeToListener | ( |  | ) | const | | --- | --- | --- | --- | --- | | inherited |
| --- | --- | --- | --- | --- | --- | --- |

Tell whether the sound's position is relative to the listener or is absolute.

ReturnsTrue if the position is relative, false if it's absolute See Also[setRelativeToListener](http://docs.google.com/classsf_1_1SoundSource.htm#ac478a8b813faf7dd575635b102081d0d)

| [Sound](http://docs.google.com/classsf_1_1Sound.htm)& sf::Sound::operator= | ( | const [Sound](http://docs.google.com/classsf_1_1Sound.htm) & | *right* | ) |  |
| --- | --- | --- | --- | --- | --- |

Overload of assignment operator.

Parameters

| right | Instance to assign |
| --- | --- |

ReturnsReference to self

| void sf::Sound::pause | ( |  | ) |  |
| --- | --- | --- | --- | --- |

Pause the sound.

This function pauses the sound if it was playing, otherwise (sound already paused or stopped) it has no effect.

See Also[play](http://docs.google.com/classsf_1_1Sound.htm#a2953ffe632536e72e696fd880ced2532), [stop](http://docs.google.com/classsf_1_1Sound.htm#aa9c91c34f7c6d344d5ee9b997511f754)

| void sf::Sound::play | ( |  | ) |  |
| --- | --- | --- | --- | --- |

Start or resume playing the sound.

This function starts the stream if it was stopped, resumes it if it was paused, and restarts it from beginning if it was it already playing. This function uses its own thread so that it doesn't block the rest of the program while the sound is played.

See Also[pause](http://docs.google.com/classsf_1_1Sound.htm#a5eeb25815bfa8cdc4a6cc000b7b19ad5), [stop](http://docs.google.com/classsf_1_1Sound.htm#aa9c91c34f7c6d344d5ee9b997511f754)

| void sf::Sound::resetBuffer | ( |  | ) |  |
| --- | --- | --- | --- | --- |

Reset the internal buffer of the sound.

This function is for internal use only, you don't have to use it. It is called by the [sf::SoundBuffer](http://docs.google.com/classsf_1_1SoundBuffer.htm) that this sound uses, when it is destroyed in order to prevent the sound from using a dead buffer.

| | void sf::SoundSource::setAttenuation | ( | float | *attenuation* | ) |  | | --- | --- | --- | --- | --- | --- | | inherited |
| --- | --- | --- | --- | --- | --- | --- | --- |

Set the attenuation factor of the sound.

The attenuation is a multiplicative factor which makes the sound more or less loud according to its distance from the listener. An attenuation of 0 will produce a non-attenuated sound, i.e. its volume will always be the same whether it is heard from near or from far. On the other hand, an attenuation value such as 100 will make the sound fade out very quickly as it gets further from the listener. The default value of the attenuation is 1.

Parameters

| attenuation | New attenuation factor of the sound |
| --- | --- |

See Also[getAttenuation](http://docs.google.com/classsf_1_1SoundSource.htm#ac5f5ffef8930bb573f43d47cbc779bf6), [setMinDistance](http://docs.google.com/classsf_1_1SoundSource.htm#a75bbc2c34addc8b25a14edb908508afe)

| void sf::Sound::setBuffer | ( | const [SoundBuffer](http://docs.google.com/classsf_1_1SoundBuffer.htm) & | *buffer* | ) |  |
| --- | --- | --- | --- | --- | --- |

Set the source buffer containing the audio data to play.

It is important to note that the sound buffer is not copied, thus the [sf::SoundBuffer](http://docs.google.com/classsf_1_1SoundBuffer.htm) instance must remain alive as long as it is attached to the sound.

Parameters

| buffer | [Sound](http://docs.google.com/classsf_1_1Sound.htm) buffer to attach to the sound |
| --- | --- |

See Also[getBuffer](http://docs.google.com/classsf_1_1Sound.htm#a5f2f1ec6603f73625f393f0e9ab64476)

| void sf::Sound::setLoop | ( | bool | *loop* | ) |  |
| --- | --- | --- | --- | --- | --- |

Set whether or not the sound should loop after reaching the end.

If set, the sound will restart from beginning after reaching the end and so on, until it is stopped or setLoop(false) is called. The default looping state for sound is false.

Parameters

| loop | True to play in loop, false to play once |
| --- | --- |

See Also[getLoop](http://docs.google.com/classsf_1_1Sound.htm#a603fbd32b519ae1680c40d8ddfc7ea51)

| | void sf::SoundSource::setMinDistance | ( | float | *distance* | ) |  | | --- | --- | --- | --- | --- | --- | | inherited |
| --- | --- | --- | --- | --- | --- | --- | --- |

Set the minimum distance of the sound.

The "minimum distance" of a sound is the maximum distance at which it is heard at its maximum volume. Further than the minimum distance, it will start to fade out according to its attenuation factor. A value of 0 ("inside the head of the listener") is an invalid value and is forbidden. The default value of the minimum distance is 1.

Parameters

| distance | New minimum distance of the sound |
| --- | --- |

See Also[getMinDistance](http://docs.google.com/classsf_1_1SoundSource.htm#a3379b9f7a0f0e31ab9a4e5fa1762986e), [setAttenuation](http://docs.google.com/classsf_1_1SoundSource.htm#aa2adff44cd2f8b4e3c7315d7c2a45626)

| | void sf::SoundSource::setPitch | ( | float | *pitch* | ) |  | | --- | --- | --- | --- | --- | --- | | inherited |
| --- | --- | --- | --- | --- | --- | --- | --- |

Set the pitch of the sound.

The pitch represents the perceived fundamental frequency of a sound; thus you can make a sound more acute or grave by changing its pitch. A side effect of changing the pitch is to modify the playing speed of the sound as well. The default value for the pitch is 1.

Parameters

| pitch | New pitch to apply to the sound |
| --- | --- |

See Also[getPitch](http://docs.google.com/classsf_1_1SoundSource.htm#aedad6aff442aeb6dcd267befd4fdbb59)

| void sf::Sound::setPlayingOffset | ( | [Time](http://docs.google.com/classsf_1_1Time.htm) | *timeOffset* | ) |  |
| --- | --- | --- | --- | --- | --- |

Change the current playing position of the sound.

The playing position can be changed when the sound is either paused or playing.

Parameters

| timeOffset | New playing position, from the beginning of the sound |
| --- | --- |

See Also[getPlayingOffset](http://docs.google.com/classsf_1_1Sound.htm#aeca741a3442d9934369a986ad40f19c0)

| | void sf::SoundSource::setPosition | ( | float | *x*, | | --- | --- | --- | --- | |  |  | float | *y*, | |  |  | float | *z* | |  | ) |  |  | | inherited |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |

Set the 3D position of the sound in the audio scene.

Only sounds with one channel (mono sounds) can be spatialized. The default position of a sound is (0, 0, 0).

Parameters

| x | X coordinate of the position of the sound in the scene |
| --- | --- |
| y | Y coordinate of the position of the sound in the scene |
| z | Z coordinate of the position of the sound in the scene |

See Also[getPosition](http://docs.google.com/classsf_1_1SoundSource.htm#a4c3bc60286f488aaf2941ab76476eebc)

| | void sf::SoundSource::setPosition | ( | const [Vector3f](http://docs.google.com/classsf_1_1Vector3.htm) & | *position* | ) |  | | --- | --- | --- | --- | --- | --- | | inherited |
| --- | --- | --- | --- | --- | --- | --- | --- |

Set the 3D position of the sound in the audio scene.

Only sounds with one channel (mono sounds) can be spatialized. The default position of a sound is (0, 0, 0).

Parameters

| position | Position of the sound in the scene |
| --- | --- |

See Also[getPosition](http://docs.google.com/classsf_1_1SoundSource.htm#a4c3bc60286f488aaf2941ab76476eebc)

| | void sf::SoundSource::setRelativeToListener | ( | bool | *relative* | ) |  | | --- | --- | --- | --- | --- | --- | | inherited |
| --- | --- | --- | --- | --- | --- | --- | --- |

Make the sound's position relative to the listener or absolute.

Making a sound relative to the listener will ensure that it will always be played the same way regardless the position of the listener. This can be useful for non-spatialized sounds, sounds that are produced by the listener, or sounds attached to it. The default value is false (position is absolute).

Parameters

| relative | True to set the position relative, false to set it absolute |
| --- | --- |

See Also[isRelativeToListener](http://docs.google.com/classsf_1_1SoundSource.htm#a5cb9107e1c47f65ab82c4885436061ef)

| | void sf::SoundSource::setVolume | ( | float | *volume* | ) |  | | --- | --- | --- | --- | --- | --- | | inherited |
| --- | --- | --- | --- | --- | --- | --- | --- |

Set the volume of the sound.

The volume is a value between 0 (mute) and 100 (full volume). The default value for the volume is 100.

Parameters

| volume | Volume of the sound |
| --- | --- |

See Also[getVolume](http://docs.google.com/classsf_1_1SoundSource.htm#aafb0558fce9cbebfc6828d932cbcce2f)

| void sf::Sound::stop | ( |  | ) |  |
| --- | --- | --- | --- | --- |

stop playing the sound

This function stops the sound if it was playing or paused, and does nothing if it was already stopped. It also resets the playing position (unlike [pause()](http://docs.google.com/classsf_1_1Sound.htm#a5eeb25815bfa8cdc4a6cc000b7b19ad5)).

See Also[play](http://docs.google.com/classsf_1_1Sound.htm#a2953ffe632536e72e696fd880ced2532), [pause](http://docs.google.com/classsf_1_1Sound.htm#a5eeb25815bfa8cdc4a6cc000b7b19ad5)

## Member Data Documentation

| | unsigned int sf::SoundSource::m\_source | | --- | | protectedinherited |
| --- | --- | --- |

OpenAL source identifier.

Definition at line [263](http://docs.google.com/SoundSource_8hpp_source.htm#l00263) of file [SoundSource.hpp](http://docs.google.com/SoundSource_8hpp_source.htm).

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

* [Sound.hpp](http://docs.google.com/Sound_8hpp_source.htm)

Copyright � Laurent Gomila  ::  Documentation generated by [doxygen](http://www.doxygen.org/)  ::