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**
* [SoundStream](http://docs.google.com/classsf_1_1SoundStream.htm)

[Classes](#_gjdgxs) | [Public Types](#_30j0zll) | [Public Member Functions](#_1fob9te) | [Protected Member Functions](#_3znysh7) | [Protected Attributes](#_2et92p0) | [List of all members](http://docs.google.com/classsf_1_1SoundStream-members.htm)

sf::SoundStream Class Referenceabstract

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

Abstract base class for streamed audio sources. [More...](http://docs.google.com/classsf_1_1SoundStream.htm#details)

#include <[SoundStream.hpp](http://docs.google.com/SoundStream_8hpp_source.htm)>

Inheritance diagram for sf::SoundStream:



| Classes | |
| --- | --- |
| struct | [Chunk](http://docs.google.com/structsf_1_1SoundStream_1_1Chunk.htm) |
|  | Structure defining a chunk of audio data to stream. [More...](http://docs.google.com/structsf_1_1SoundStream_1_1Chunk.htm#details) |
|  | |

| 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 | |
| --- | --- |
| virtual | [~SoundStream](http://docs.google.com/classsf_1_1SoundStream.htm#a1fafb9f1ca572d23d7d6a17921860d85) () |
|  | Destructor. |
|  | |
| void | [play](http://docs.google.com/classsf_1_1SoundStream.htm#afdc08b69cab5f243d9324940a85a1144) () |
|  | Start or resume playing the audio stream. |
|  | |
| void | [pause](http://docs.google.com/classsf_1_1SoundStream.htm#a932ff181e661503cad288b4bb6fe45ca) () |
|  | Pause the audio stream. |
|  | |
| void | [stop](http://docs.google.com/classsf_1_1SoundStream.htm#a16cc6a0404b32e42c4dce184bb94d0f4) () |
|  | Stop playing the audio stream. |
|  | |
| unsigned int | [getChannelCount](http://docs.google.com/classsf_1_1SoundStream.htm#a68dedd0a2c26e6937c80fab3d235edea) () const |
|  | Return the number of channels of the stream. |
|  | |
| unsigned int | [getSampleRate](http://docs.google.com/classsf_1_1SoundStream.htm#aad1da51c7a752682ca208bf11e3349bb) () const |
|  | Get the stream sample rate of the stream. |
|  | |
| [Status](http://docs.google.com/classsf_1_1SoundSource.htm#ac43af72c98c077500b239bc75b812f03) | [getStatus](http://docs.google.com/classsf_1_1SoundStream.htm#a75f722e7edcfa9952ff0c643966c6858) () const |
|  | Get the current status of the stream (stopped, paused, playing) |
|  | |
| void | [setPlayingOffset](http://docs.google.com/classsf_1_1SoundStream.htm#af416a5f84c8750d2acb9821d78bc8646) ([Time](http://docs.google.com/classsf_1_1Time.htm) timeOffset) |
|  | Change the current playing position of the stream. |
|  | |
| [Time](http://docs.google.com/classsf_1_1Time.htm) | [getPlayingOffset](http://docs.google.com/classsf_1_1SoundStream.htm#a6070416e1e1a11b5915e9314dd6638f7) () const |
|  | Get the current playing position of the stream. |
|  | |
| void | [setLoop](http://docs.google.com/classsf_1_1SoundStream.htm#a43fade018ffba7e4f847a9f00b353f3d) (bool loop) |
|  | Set whether or not the stream should loop after reaching the end. |
|  | |
| bool | [getLoop](http://docs.google.com/classsf_1_1SoundStream.htm#ad8bbf6c0d2ff2d75e19035dea3fd77c3) () const |
|  | Tell whether or not the stream is in loop mode. |
|  | |
| 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 Member Functions | |
| --- | --- |
|  | [SoundStream](http://docs.google.com/classsf_1_1SoundStream.htm#a769d08f4c3c6b4340ef3a838329d2e5c) () |
|  | Default constructor. |
|  | |
| void | [initialize](http://docs.google.com/classsf_1_1SoundStream.htm#a9c351711198ee1aa77c2fefd3ced4d2c) (unsigned int channelCount, unsigned int sampleRate) |
|  | Define the audio stream parameters. |
|  | |
| virtual bool | [onGetData](http://docs.google.com/classsf_1_1SoundStream.htm#a968ec024a6e45490962c8a1121cb7c5f) ([Chunk](http://docs.google.com/structsf_1_1SoundStream_1_1Chunk.htm) &data)=0 |
|  | Request a new chunk of audio samples from the stream source. |
|  | |
| virtual void | [onSeek](http://docs.google.com/classsf_1_1SoundStream.htm#a907036dd2ca7d3af5ead316e54b75997) ([Time](http://docs.google.com/classsf_1_1Time.htm) timeOffset)=0 |
|  | Change the current playing position in the stream source. |
|  | |

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

## Detailed Description

Abstract base class for streamed audio sources.

Unlike audio buffers (see [sf::SoundBuffer](http://docs.google.com/classsf_1_1SoundBuffer.htm)), audio streams are never completely loaded in memory.

Instead, the audio data is acquired continuously while the stream is playing. This behaviour allows to play a sound with no loading delay, and keeps the memory consumption very low.

[Sound](http://docs.google.com/classsf_1_1Sound.htm) sources that need to be streamed are usually big files (compressed audio musics that would eat hundreds of MB in memory) or files that would take a lot of time to be received (sounds played over the network).

[sf::SoundStream](http://docs.google.com/classsf_1_1SoundStream.htm) is a base class that doesn't care about the stream source, which is left to the derived class. SFML provides a built-in specialization for big files (see [sf::Music](http://docs.google.com/classsf_1_1Music.htm)). No network stream source is provided, but you can write your own by combining this class with the network module.

A derived class has to override two virtual functions:

* onGetData fills a new chunk of audio data to be played
* onSeek changes the current playing position in the source

It is important to note that each [SoundStream](http://docs.google.com/classsf_1_1SoundStream.htm) is played in its own separate thread, so that the streaming loop doesn't block the rest of the program. In particular, the OnGetData and OnSeek virtual functions may sometimes be called from this separate thread. It is important to keep this in mind, because you may have to take care of synchronization issues if you share data between threads.

Usage example:

class CustomStream : public [sf::SoundStream](http://docs.google.com/classsf_1_1SoundStream.htm)

{

public :

bool open(const std::string& location)

{

// Open the source and get audio settings

...

unsigned int channelCount = ...;

unsigned int sampleRate = ...;

// Initialize the stream -- important!

[initialize](http://docs.google.com/classsf_1_1SoundStream.htm#a9c351711198ee1aa77c2fefd3ced4d2c)(channelCount, sampleRate);

}

private :

virtual bool [onGetData](http://docs.google.com/classsf_1_1SoundStream.htm#a968ec024a6e45490962c8a1121cb7c5f)(Chunk& data)

{

// Fill the chunk with audio data from the stream source

data.samples = ...;

data.sampleCount = ...;

// Return true to continue playing

return true;

}

virtual void [onSeek](http://docs.google.com/classsf_1_1SoundStream.htm#a907036dd2ca7d3af5ead316e54b75997)(Uint32 timeOffset)

{

// Change the current position in the stream source

...

}

}

// Usage

CustomStream stream;

stream.open("path/to/stream");

stream.play();

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

Definition at line [44](http://docs.google.com/SoundStream_8hpp_source.htm#l00044) of file [SoundStream.hpp](http://docs.google.com/SoundStream_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

| | virtual sf::SoundStream::~SoundStream | ( |  | ) |  | | --- | --- | --- | --- | --- | | virtual |
| --- | --- | --- | --- | --- | --- | --- |

Destructor.

| | sf::SoundStream::SoundStream | ( |  | ) |  | | --- | --- | --- | --- | --- | | protected |
| --- | --- | --- | --- | --- | --- | --- |

Default constructor.

This constructor is only meant to be called by derived classes.

## 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)

| unsigned int sf::SoundStream::getChannelCount | ( |  | ) | const |
| --- | --- | --- | --- | --- |

Return the number of channels of the stream.

1 channel means a mono sound, 2 means stereo, etc.

ReturnsNumber of channels

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

Tell whether or not the stream is in loop mode.

ReturnsTrue if the stream is looping, false otherwise See Also[setLoop](http://docs.google.com/classsf_1_1SoundStream.htm#a43fade018ffba7e4f847a9f00b353f3d)

| | 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::SoundStream::getPlayingOffset | ( |  | ) | const |
| --- | --- | --- | --- | --- |

Get the current playing position of the stream.

ReturnsCurrent playing position, from the beginning of the stream See Also[setPlayingOffset](http://docs.google.com/classsf_1_1SoundStream.htm#af416a5f84c8750d2acb9821d78bc8646)

| | [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)

| unsigned int sf::SoundStream::getSampleRate | ( |  | ) | const |
| --- | --- | --- | --- | --- |

Get the stream sample rate of the stream.

The sample rate is the number of audio samples played per second. The higher, the better the quality.

ReturnsSample rate, in number of samples per second

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

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

ReturnsCurrent status

| | 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)

| | void sf::SoundStream::initialize | ( | unsigned int | *channelCount*, | | --- | --- | --- | --- | |  |  | unsigned int | *sampleRate* | |  | ) |  |  | | protected |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |

Define the audio stream parameters.

This function must be called by derived classes as soon as they know the audio settings of the stream to play. Any attempt to manipulate the stream ([play()](http://docs.google.com/classsf_1_1SoundStream.htm#afdc08b69cab5f243d9324940a85a1144), ...) before calling this function will fail. It can be called multiple times if the settings of the audio stream change, but only when the stream is stopped.

Parameters

| channelCount | Number of channels of the stream |
| --- | --- |
| sampleRate | Sample rate, in samples per second |

| | 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)

| | virtual bool sf::SoundStream::onGetData | ( | [Chunk](http://docs.google.com/structsf_1_1SoundStream_1_1Chunk.htm) & | *data* | ) |  | | --- | --- | --- | --- | --- | --- | | protectedpure virtual |
| --- | --- | --- | --- | --- | --- | --- | --- |

Request a new chunk of audio samples from the stream source.

This function must be overriden by derived classes to provide the audio samples to play. It is called continuously by the streaming loop, in a separate thread. The source can choose to stop the streaming loop at any time, by returning false to the caller.

Parameters

| data | [Chunk](http://docs.google.com/structsf_1_1SoundStream_1_1Chunk.htm) of data to fill |
| --- | --- |

ReturnsTrue to continue playback, false to stop

Implemented in [sf::Music](http://docs.google.com/classsf_1_1Music.htm#aca1bcb4e5d56a854133e74bd86374463).

| | virtual void sf::SoundStream::onSeek | ( | [Time](http://docs.google.com/classsf_1_1Time.htm) | *timeOffset* | ) |  | | --- | --- | --- | --- | --- | --- | | protectedpure virtual |
| --- | --- | --- | --- | --- | --- | --- | --- |

Change the current playing position in the stream source.

This function must be overriden by derived classes to allow random seeking into the stream source.

Parameters

| timeOffset | New playing position, relative to the beginning of the stream |
| --- | --- |

Implemented in [sf::Music](http://docs.google.com/classsf_1_1Music.htm#a15119cc0419c16bb334fa0698699c02e).

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

Pause the audio stream.

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

See Also[play](http://docs.google.com/classsf_1_1SoundStream.htm#afdc08b69cab5f243d9324940a85a1144), [stop](http://docs.google.com/classsf_1_1SoundStream.htm#a16cc6a0404b32e42c4dce184bb94d0f4)

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

Start or resume playing the audio stream.

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 stream is played.

See Also[pause](http://docs.google.com/classsf_1_1SoundStream.htm#a932ff181e661503cad288b4bb6fe45ca), [stop](http://docs.google.com/classsf_1_1SoundStream.htm#a16cc6a0404b32e42c4dce184bb94d0f4)

| | 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::SoundStream::setLoop | ( | bool | *loop* | ) |  |
| --- | --- | --- | --- | --- | --- |

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

If set, the stream 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 streams is false.

Parameters

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

See Also[getLoop](http://docs.google.com/classsf_1_1SoundStream.htm#ad8bbf6c0d2ff2d75e19035dea3fd77c3)

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

Change the current playing position of the stream.

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

Parameters

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

See Also[getPlayingOffset](http://docs.google.com/classsf_1_1SoundStream.htm#a6070416e1e1a11b5915e9314dd6638f7)

| | 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::SoundStream::stop | ( |  | ) |  |
| --- | --- | --- | --- | --- |

Stop playing the audio stream.

This function stops the stream 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_1SoundStream.htm#a932ff181e661503cad288b4bb6fe45ca)).

See Also[play](http://docs.google.com/classsf_1_1SoundStream.htm#afdc08b69cab5f243d9324940a85a1144), [pause](http://docs.google.com/classsf_1_1SoundStream.htm#a932ff181e661503cad288b4bb6fe45ca)

## 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:

* [SoundStream.hpp](http://docs.google.com/SoundStream_8hpp_source.htm)

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