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)
* [File List](http://docs.google.com/files.htm)
* [include](http://docs.google.com/dir_f3190241575fd2bd132a392ae6942f4a.htm)
* [SFML](http://docs.google.com/dir_692f376662c82a26cfe4cfa3aceebe24.htm)
* [Audio](http://docs.google.com/dir_3c6cd700807083a0c327f22d879e95e3.htm)

SoundStream.hpp

1

2 //

3 // SFML - Simple and Fast Multimedia Library

4 // Copyright (C) 2007-2013 Laurent Gomila (laurent.gom@gmail.com)

5 //

6 // This software is provided 'as-is', without any express or implied warranty.

7 // In no event will the authors be held liable for any damages arising from the use of this software.

8 //

9 // Permission is granted to anyone to use this software for any purpose,

10 // including commercial applications, and to alter it and redistribute it freely,

11 // subject to the following restrictions:

12 //

13 // 1. The origin of this software must not be misrepresented;

14 // you must not claim that you wrote the original software.

15 // If you use this software in a product, an acknowledgment

16 // in the product documentation would be appreciated but is not required.

17 //

18 // 2. Altered source versions must be plainly marked as such,

19 // and must not be misrepresented as being the original software.

20 //

21 // 3. This notice may not be removed or altered from any source distribution.

22 //

24

25 #ifndef SFML\_SOUNDSTREAM\_HPP

26 #define SFML\_SOUNDSTREAM\_HPP

27

29 // Headers

31 #include <SFML/Audio/Export.hpp>

32 #include <SFML/Audio/SoundSource.hpp>

33 #include <SFML/System/Thread.hpp>

34 #include <SFML/System/Time.hpp>

35 #include <cstdlib>

36

37

38 namespace sf

39 {

[44](http://docs.google.com/classsf_1_1SoundStream.htm) class SFML\_AUDIO\_API [SoundStream](http://docs.google.com/classsf_1_1SoundStream.htm) : public [SoundSource](http://docs.google.com/classsf_1_1SoundSource.htm)

45 {

46 public :

47

[52](http://docs.google.com/structsf_1_1SoundStream_1_1Chunk.htm)  struct [Chunk](http://docs.google.com/structsf_1_1SoundStream_1_1Chunk.htm)

53  {

[54](http://docs.google.com/structsf_1_1SoundStream_1_1Chunk.htm#aa3b84d69adbe663a17a7671626076df4)  const Int16\* [samples](http://docs.google.com/structsf_1_1SoundStream_1_1Chunk.htm#aa3b84d69adbe663a17a7671626076df4);

[55](http://docs.google.com/structsf_1_1SoundStream_1_1Chunk.htm#af47f5d94012acf8b11f056ba77aff97a)  std::size\_t [sampleCount](http://docs.google.com/structsf_1_1SoundStream_1_1Chunk.htm#af47f5d94012acf8b11f056ba77aff97a);

56  };

57

62  virtual ~[SoundStream](http://docs.google.com/classsf_1_1SoundStream.htm)();

63

76  void play();

77

87  void pause();

88

99  void stop();

100

109  unsigned int getChannelCount() const;

110

120  unsigned int getSampleRate() const;

121

128  [Status](http://docs.google.com/classsf_1_1SoundSource.htm#ac43af72c98c077500b239bc75b812f03) getStatus() const;

129

141  void setPlayingOffset([Time](http://docs.google.com/classsf_1_1Time.htm) timeOffset);

142

151  [Time](http://docs.google.com/classsf_1_1Time.htm) getPlayingOffset() const;

152

166  void setLoop(bool loop);

167

176  bool getLoop() const;

177

178 protected :

179

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

187

202  void initialize(unsigned int channelCount, unsigned int sampleRate);

203

218  virtual bool onGetData([Chunk](http://docs.google.com/structsf_1_1SoundStream_1_1Chunk.htm)& data) = 0;

219

229  virtual void onSeek([Time](http://docs.google.com/classsf_1_1Time.htm) timeOffset) = 0;

230

231 private :

232

240  void streamData();

241

255  bool fillAndPushBuffer(unsigned int bufferNum);

256

266  bool fillQueue();

267

274  void clearQueue();

275

276  enum

277  {

278  BufferCount = 3

279  };

280

282  // Member data

284  [Thread](http://docs.google.com/classsf_1_1Thread.htm) m\_thread;

285  bool m\_isStreaming;

286  unsigned int m\_buffers[BufferCount];

287  unsigned int m\_channelCount;

288  unsigned int m\_sampleRate;

289  Uint32 m\_format;

290  bool m\_loop;

291  Uint64 m\_samplesProcessed;

292  bool m\_endBuffers[BufferCount];

293 };

294

295 } // namespace sf

296

297

298 #endif // SFML\_SOUNDSTREAM\_HPP

299

300

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