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)

Class Hierarchy

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

[detail level 123]

| [sf::SoundStream::Chunk](http://docs.google.com/structsf_1_1SoundStream_1_1Chunk.htm) | Structure defining a chunk of audio data to stream |
| --- | --- |
| [sf::Clock](http://docs.google.com/classsf_1_1Clock.htm) | Utility class that measures the elapsed time |
| [sf::Color](http://docs.google.com/classsf_1_1Color.htm) | Utility class for manpulating RGBA colors |
| [sf::ContextSettings](http://docs.google.com/structsf_1_1ContextSettings.htm) | Structure defining the settings of the OpenGL context attached to a window |
| [sf::Shader::CurrentTextureType](http://docs.google.com/structsf_1_1Shader_1_1CurrentTextureType.htm) | Special type/value that can be passed to setParameter, and that represents the texture of the object being drawn |
| [sf::Drawable](http://docs.google.com/classsf_1_1Drawable.htm) | Abstract base class for objects that can be drawn to a render target |
| [sf::Shape](http://docs.google.com/classsf_1_1Shape.htm) | Base class for textured shapes with outline |
| [sf::CircleShape](http://docs.google.com/classsf_1_1CircleShape.htm) | Specialized shape representing a circle |
| [sf::ConvexShape](http://docs.google.com/classsf_1_1ConvexShape.htm) | Specialized shape representing a convex polygon |
| [sf::RectangleShape](http://docs.google.com/classsf_1_1RectangleShape.htm) | Specialized shape representing a rectangle |
| [sf::Sprite](http://docs.google.com/classsf_1_1Sprite.htm) | [Drawable](http://docs.google.com/classsf_1_1Drawable.htm) representation of a texture, with its own transformations, color, etc |
| [sf::Text](http://docs.google.com/classsf_1_1Text.htm) | Graphical text that can be drawn to a render target |
| [sf::VertexArray](http://docs.google.com/classsf_1_1VertexArray.htm) | Define a set of one or more 2D primitives |
| [sf::Event](http://docs.google.com/classsf_1_1Event.htm) | Defines a system event and its parameters |
| [sf::Font](http://docs.google.com/classsf_1_1Font.htm) | Class for loading and manipulating character fonts |
| [sf::GlResource](http://docs.google.com/classsf_1_1GlResource.htm) | Base class for classes that require an OpenGL context |
| [sf::Context](http://docs.google.com/classsf_1_1Context.htm) | Class holding a valid drawing context |
| [sf::Shader](http://docs.google.com/classsf_1_1Shader.htm) | [Shader](http://docs.google.com/classsf_1_1Shader.htm) class (vertex and fragment) |
| [sf::Texture](http://docs.google.com/classsf_1_1Texture.htm) | [Image](http://docs.google.com/classsf_1_1Image.htm) living on the graphics card that can be used for drawing |
| [sf::Window](http://docs.google.com/classsf_1_1Window.htm) | [Window](http://docs.google.com/classsf_1_1Window.htm) that serves as a target for OpenGL rendering |
| [sf::RenderWindow](http://docs.google.com/classsf_1_1RenderWindow.htm) | [Window](http://docs.google.com/classsf_1_1Window.htm) that can serve as a target for 2D drawing |
| [sf::Glyph](http://docs.google.com/classsf_1_1Glyph.htm) | Structure describing a glyph |
| [sf::Image](http://docs.google.com/classsf_1_1Image.htm) | Class for loading, manipulating and saving images |
| [sf::InputStream](http://docs.google.com/classsf_1_1InputStream.htm) | Abstract class for custom file input streams |
| [sf::IpAddress](http://docs.google.com/classsf_1_1IpAddress.htm) | Encapsulate an IPv4 network address |
| [sf::Joystick](http://docs.google.com/classsf_1_1Joystick.htm) | Give access to the real-time state of the joysticks |
| [sf::Event::JoystickButtonEvent](http://docs.google.com/structsf_1_1Event_1_1JoystickButtonEvent.htm) | [Joystick](http://docs.google.com/classsf_1_1Joystick.htm) buttons events parameters (JoystickButtonPressed, JoystickButtonReleased) |
| [sf::Event::JoystickConnectEvent](http://docs.google.com/structsf_1_1Event_1_1JoystickConnectEvent.htm) | [Joystick](http://docs.google.com/classsf_1_1Joystick.htm) connection events parameters (JoystickConnected, JoystickDisconnected) |
| [sf::Event::JoystickMoveEvent](http://docs.google.com/structsf_1_1Event_1_1JoystickMoveEvent.htm) | [Joystick](http://docs.google.com/classsf_1_1Joystick.htm) axis move event parameters (JoystickMoved) |
| [sf::Keyboard](http://docs.google.com/classsf_1_1Keyboard.htm) | Give access to the real-time state of the keyboard |
| [sf::Event::KeyEvent](http://docs.google.com/structsf_1_1Event_1_1KeyEvent.htm) | [Keyboard](http://docs.google.com/classsf_1_1Keyboard.htm) event parameters (KeyPressed, KeyReleased) |
| [sf::Listener](http://docs.google.com/classsf_1_1Listener.htm) | The audio listener is the point in the scene from where all the sounds are heard |
| [sf::Mouse](http://docs.google.com/classsf_1_1Mouse.htm) | Give access to the real-time state of the mouse |
| [sf::Event::MouseButtonEvent](http://docs.google.com/structsf_1_1Event_1_1MouseButtonEvent.htm) | [Mouse](http://docs.google.com/classsf_1_1Mouse.htm) buttons events parameters (MouseButtonPressed, MouseButtonReleased) |
| [sf::Event::MouseMoveEvent](http://docs.google.com/structsf_1_1Event_1_1MouseMoveEvent.htm) | [Mouse](http://docs.google.com/classsf_1_1Mouse.htm) move event parameters (MouseMoved) |
| [sf::Event::MouseWheelEvent](http://docs.google.com/structsf_1_1Event_1_1MouseWheelEvent.htm) | [Mouse](http://docs.google.com/classsf_1_1Mouse.htm) wheel events parameters (MouseWheelMoved) |
| [sf::NonCopyable](http://docs.google.com/classsf_1_1NonCopyable.htm) | Utility class that makes any derived class non-copyable |
| [sf::Context](http://docs.google.com/classsf_1_1Context.htm) | Class holding a valid drawing context |
| [sf::Ftp](http://docs.google.com/classsf_1_1Ftp.htm) | A FTP client |
| [sf::Http](http://docs.google.com/classsf_1_1Http.htm) | A HTTP client |
| [sf::Lock](http://docs.google.com/classsf_1_1Lock.htm) | Automatic wrapper for locking and unlocking mutexes |
| [sf::Mutex](http://docs.google.com/classsf_1_1Mutex.htm) | Blocks concurrent access to shared resources from multiple threads |
| [sf::RenderTarget](http://docs.google.com/classsf_1_1RenderTarget.htm) | Base class for all render targets (window, texture, ...) |
| [sf::RenderTexture](http://docs.google.com/classsf_1_1RenderTexture.htm) | Target for off-screen 2D rendering into a texture |
| [sf::RenderWindow](http://docs.google.com/classsf_1_1RenderWindow.htm) | [Window](http://docs.google.com/classsf_1_1Window.htm) that can serve as a target for 2D drawing |
| [sf::Shader](http://docs.google.com/classsf_1_1Shader.htm) | [Shader](http://docs.google.com/classsf_1_1Shader.htm) class (vertex and fragment) |
| [sf::Socket](http://docs.google.com/classsf_1_1Socket.htm) | Base class for all the socket types |
| [sf::TcpListener](http://docs.google.com/classsf_1_1TcpListener.htm) | [Socket](http://docs.google.com/classsf_1_1Socket.htm) that listens to new TCP connections |
| [sf::TcpSocket](http://docs.google.com/classsf_1_1TcpSocket.htm) | Specialized socket using the TCP protocol |
| [sf::UdpSocket](http://docs.google.com/classsf_1_1UdpSocket.htm) | Specialized socket using the UDP protocol |
| [sf::Thread](http://docs.google.com/classsf_1_1Thread.htm) | Utility class to manipulate threads |
| [sf::ThreadLocal](http://docs.google.com/classsf_1_1ThreadLocal.htm) | Defines variables with thread-local storage |
| [sf::ThreadLocalPtr< T >](http://docs.google.com/classsf_1_1ThreadLocalPtr.htm) | Pointer to a thread-local variable |
| [sf::Window](http://docs.google.com/classsf_1_1Window.htm) | [Window](http://docs.google.com/classsf_1_1Window.htm) that serves as a target for OpenGL rendering |
| [sf::Packet](http://docs.google.com/classsf_1_1Packet.htm) | Utility class to build blocks of data to transfer over the network |
| [sf::Rect< T >](http://docs.google.com/classsf_1_1Rect.htm) | Utility class for manipulating 2D axis aligned rectangles |
| [sf::Rect< float >](http://docs.google.com/classsf_1_1Rect.htm) |  |
| [sf::Rect< int >](http://docs.google.com/classsf_1_1Rect.htm) |  |
| [sf::RenderStates](http://docs.google.com/classsf_1_1RenderStates.htm) | Define the states used for drawing to a [RenderTarget](http://docs.google.com/classsf_1_1RenderTarget.htm) |
| [sf::Http::Request](http://docs.google.com/classsf_1_1Http_1_1Request.htm) | Define a HTTP request |
| [sf::Ftp::Response](http://docs.google.com/classsf_1_1Ftp_1_1Response.htm) | Define a FTP response |
| [sf::Ftp::DirectoryResponse](http://docs.google.com/classsf_1_1Ftp_1_1DirectoryResponse.htm) | Specialization of FTP response returning a directory |
| [sf::Ftp::ListingResponse](http://docs.google.com/classsf_1_1Ftp_1_1ListingResponse.htm) | Specialization of FTP response returning a filename lisiting |
| [sf::Http::Response](http://docs.google.com/classsf_1_1Http_1_1Response.htm) | Define a HTTP response |
| [sf::Event::SizeEvent](http://docs.google.com/structsf_1_1Event_1_1SizeEvent.htm) | Size events parameters (Resized) |
| [sf::SocketSelector](http://docs.google.com/classsf_1_1SocketSelector.htm) | Multiplexer that allows to read from multiple sockets |
| [sf::SoundBuffer](http://docs.google.com/classsf_1_1SoundBuffer.htm) | Storage for audio samples defining a sound |
| [sf::SoundRecorder](http://docs.google.com/classsf_1_1SoundRecorder.htm) | Abstract base class for capturing sound data |
| [sf::SoundBufferRecorder](http://docs.google.com/classsf_1_1SoundBufferRecorder.htm) | Specialized [SoundRecorder](http://docs.google.com/classsf_1_1SoundRecorder.htm) which stores the captured audio data into a sound buffer |
| [sf::SoundSource](http://docs.google.com/classsf_1_1SoundSource.htm) | Base class defining a sound's properties |
| [sf::Sound](http://docs.google.com/classsf_1_1Sound.htm) | Regular sound that can be played in the audio environment |
| [sf::SoundStream](http://docs.google.com/classsf_1_1SoundStream.htm) | Abstract base class for streamed audio sources |
| [sf::Music](http://docs.google.com/classsf_1_1Music.htm) | Streamed music played from an audio file |
| [sf::String](http://docs.google.com/classsf_1_1String.htm) | Utility string class that automatically handles conversions between types and encodings |
| [sf::Event::TextEvent](http://docs.google.com/structsf_1_1Event_1_1TextEvent.htm) | [Text](http://docs.google.com/classsf_1_1Text.htm) event parameters (TextEntered) |
| [sf::Time](http://docs.google.com/classsf_1_1Time.htm) | Represents a time value |
| [sf::Transform](http://docs.google.com/classsf_1_1Transform.htm) | Define a 3x3 transform matrix |
| [sf::Transformable](http://docs.google.com/classsf_1_1Transformable.htm) | Decomposed transform defined by a position, a rotation and a scale |
| [sf::Shape](http://docs.google.com/classsf_1_1Shape.htm) | Base class for textured shapes with outline |
| [sf::Sprite](http://docs.google.com/classsf_1_1Sprite.htm) | [Drawable](http://docs.google.com/classsf_1_1Drawable.htm) representation of a texture, with its own transformations, color, etc |
| [sf::Text](http://docs.google.com/classsf_1_1Text.htm) | Graphical text that can be drawn to a render target |
| [Utf< N >](http://docs.google.com/classUtf.htm) |  |
| [sf::Utf](http://docs.google.com/classsf_1_1Utf.htm) | Utility class providing generic functions for UTF conversions |
| [sf::Utf< 16 >](http://docs.google.com/classsf_1_1Utf_3_0116_01_4.htm) | Specialization of the [Utf](http://docs.google.com/classsf_1_1Utf.htm) template for UTF-16 |
| [sf::Utf< 32 >](http://docs.google.com/classsf_1_1Utf_3_0132_01_4.htm) | Specialization of the [Utf](http://docs.google.com/classsf_1_1Utf.htm) template for UTF-32 |
| [sf::Utf< 8 >](http://docs.google.com/classsf_1_1Utf_3_018_01_4.htm) | Specialization of the [Utf](http://docs.google.com/classsf_1_1Utf.htm) template for UTF-8 |
| [sf::Vector2< T >](http://docs.google.com/classsf_1_1Vector2.htm) | Utility template class for manipulating 2-dimensional vectors |
| [sf::Vector2< float >](http://docs.google.com/classsf_1_1Vector2.htm) |  |
| [sf::Vector2< unsigned int >](http://docs.google.com/classsf_1_1Vector2.htm) |  |
| [sf::Vector3< T >](http://docs.google.com/classsf_1_1Vector3.htm) | Utility template class for manipulating 3-dimensional vectors |
| [sf::Vertex](http://docs.google.com/classsf_1_1Vertex.htm) | Define a point with color and texture coordinates |
| [sf::VideoMode](http://docs.google.com/classsf_1_1VideoMode.htm) | [VideoMode](http://docs.google.com/classsf_1_1VideoMode.htm) defines a video mode (width, height, bpp) |
| [sf::View](http://docs.google.com/classsf_1_1View.htm) | 2D camera that defines what region is shown on screen |

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