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)
* [Graphics](http://docs.google.com/dir_aaa96c3797a59111c2945d0d638ce5cf.htm)

Image.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\_IMAGE\_HPP

26 #define SFML\_IMAGE\_HPP

27

29 // Headers

31 #include <SFML/Graphics/Export.hpp>

32 #include <SFML/Graphics/Color.hpp>

33 #include <SFML/Graphics/Rect.hpp>

34 #include <string>

35 #include <vector>

36

37

38 namespace sf

39 {

40 class InputStream;

41

[46](http://docs.google.com/classsf_1_1Image.htm) class SFML\_GRAPHICS\_API [Image](http://docs.google.com/classsf_1_1Image.htm)

47 {

48 public :

49

56  [Image](http://docs.google.com/classsf_1_1Image.htm)();

57

66  void create(unsigned int width, unsigned int height, const [Color](http://docs.google.com/classsf_1_1Color.htm)& color = [Color](http://docs.google.com/classsf_1_1Color.htm)(0, 0, 0));

67

81  void create(unsigned int width, unsigned int height, const Uint8\* pixels);

82

98  bool loadFromFile(const std::string& filename);

99

116  bool loadFromMemory(const void\* data, std::size\_t size);

117

133  bool loadFromStream([InputStream](http://docs.google.com/classsf_1_1InputStream.htm)& stream);

134

150  bool saveToFile(const std::string& filename) const;

151

158  [Vector2u](http://docs.google.com/classsf_1_1Vector2.htm) getSize() const;

159

171  void createMaskFromColor(const [Color](http://docs.google.com/classsf_1_1Color.htm)& color, Uint8 alpha = 0);

172

193  void copy(const [Image](http://docs.google.com/classsf_1_1Image.htm)& source, unsigned int destX, unsigned int destY, const [IntRect](http://docs.google.com/classsf_1_1Rect.htm)& sourceRect = [IntRect](http://docs.google.com/classsf_1_1Rect.htm)(0, 0, 0, 0), bool applyAlpha = false);

194

209  void setPixel(unsigned int x, unsigned int y, const [Color](http://docs.google.com/classsf_1_1Color.htm)& color);

210

226  [Color](http://docs.google.com/classsf_1_1Color.htm) getPixel(unsigned int x, unsigned int y) const;

227

241  const Uint8\* getPixelsPtr() const;

242

247  void flipHorizontally();

248

253  void flipVertically();

254

255 private :

256

258  // Member data

260  [Vector2u](http://docs.google.com/classsf_1_1Vector2.htm) m\_size;

261  std::vector<Uint8> m\_pixels;

262 };

263

264 } // namespace sf

265

266

267 #endif // SFML\_IMAGE\_HPP

268

269

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