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)

Rect.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\_RECT\_HPP

26 #define SFML\_RECT\_HPP

27

29 // Headers

31 #include <SFML/System/Vector2.hpp>

32 #include <algorithm>

33

34

35 namespace sf

36 {

41 template <typename T>

[42](http://docs.google.com/classsf_1_1Rect.htm) class [Rect](http://docs.google.com/classsf_1_1Rect.htm)

43 {

44 public :

45

53  [Rect](http://docs.google.com/classsf_1_1Rect.htm#a0f87ebaef9722a6222fd2e04ce8efb37)();

54

67  [Rect](http://docs.google.com/classsf_1_1Rect.htm#a0f87ebaef9722a6222fd2e04ce8efb37)(T rectLeft, T rectTop, T rectWidth, T rectHeight);

68

79  [Rect](http://docs.google.com/classsf_1_1Rect.htm#a0f87ebaef9722a6222fd2e04ce8efb37)(const [Vector2<T>](http://docs.google.com/classsf_1_1Vector2.htm)& position, const [Vector2<T>](http://docs.google.com/classsf_1_1Vector2.htm)& size);

80

92  template <typename U>

93  explicit [Rect](http://docs.google.com/classsf_1_1Rect.htm#a0f87ebaef9722a6222fd2e04ce8efb37)(const [Rect<U>](http://docs.google.com/classsf_1_1Rect.htm)& rectangle);

94

106  bool [contains](http://docs.google.com/classsf_1_1Rect.htm#aa8a5364c84de6dd5299f833b54e31ef1)(T x, T y) const;

107

118  bool [contains](http://docs.google.com/classsf_1_1Rect.htm#aa8a5364c84de6dd5299f833b54e31ef1)(const [Vector2<T>](http://docs.google.com/classsf_1_1Vector2.htm)& point) const;

119

130  bool [intersects](http://docs.google.com/classsf_1_1Rect.htm#a566740c8f58e01bb052266f47e7e1011)(const [Rect<T>](http://docs.google.com/classsf_1_1Rect.htm)& rectangle) const;

131

146  bool [intersects](http://docs.google.com/classsf_1_1Rect.htm#a566740c8f58e01bb052266f47e7e1011)(const [Rect<T>](http://docs.google.com/classsf_1_1Rect.htm)& rectangle, [Rect<T>](http://docs.google.com/classsf_1_1Rect.htm)& intersection) const;

147

149  // Member data

[151](http://docs.google.com/classsf_1_1Rect.htm#aa49960fa465103d9cb7069ceb25c7c32)  T [left](http://docs.google.com/classsf_1_1Rect.htm#aa49960fa465103d9cb7069ceb25c7c32);

[152](http://docs.google.com/classsf_1_1Rect.htm#abd3d3a2d0ad211ef0082bd0aa1a5c0e3)  T [top](http://docs.google.com/classsf_1_1Rect.htm#abd3d3a2d0ad211ef0082bd0aa1a5c0e3);

[153](http://docs.google.com/classsf_1_1Rect.htm#a4dd5b9d4333bebbc51bd309298fd500f)  T [width](http://docs.google.com/classsf_1_1Rect.htm#a4dd5b9d4333bebbc51bd309298fd500f);

[154](http://docs.google.com/classsf_1_1Rect.htm#a6fa0fc7de1636d78cae1a1b54eef95cd)  T [height](http://docs.google.com/classsf_1_1Rect.htm#a6fa0fc7de1636d78cae1a1b54eef95cd);

155 };

156

169 template <typename T>

170 bool operator ==(const [Rect<T>](http://docs.google.com/classsf_1_1Rect.htm)& left, const [Rect<T>](http://docs.google.com/classsf_1_1Rect.htm)& right);

171

184 template <typename T>

185 bool operator !=(const [Rect<T>](http://docs.google.com/classsf_1_1Rect.htm)& left, const [Rect<T>](http://docs.google.com/classsf_1_1Rect.htm)& right);

186

187 #include <SFML/Graphics/Rect.inl>

188

189 // Create typedefs for the most common types

190 typedef [Rect<int>](http://docs.google.com/classsf_1_1Rect.htm) [IntRect](http://docs.google.com/classsf_1_1Rect.htm);

191 typedef [Rect<float>](http://docs.google.com/classsf_1_1Rect.htm) [FloatRect](http://docs.google.com/classsf_1_1Rect.htm);

192

193 } // namespace sf

194

195

196 #endif // SFML\_RECT\_HPP

197

198

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