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)

Transform.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\_TRANSFORM\_HPP

26 #define SFML\_TRANSFORM\_HPP

27

29 // Headers

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

32 #include <SFML/Graphics/Rect.hpp>

33 #include <SFML/System/Vector2.hpp>

34

35

36 namespace sf

37 {

[42](http://docs.google.com/classsf_1_1Transform.htm) class SFML\_GRAPHICS\_API [Transform](http://docs.google.com/classsf_1_1Transform.htm)

43 {

44 public :

45

52  [Transform](http://docs.google.com/classsf_1_1Transform.htm)();

53

68  [Transform](http://docs.google.com/classsf_1_1Transform.htm)(float a00, float a01, float a02,

69  float a10, float a11, float a12,

70  float a20, float a21, float a22);

71

87  const float\* getMatrix() const;

88

98  [Transform](http://docs.google.com/classsf_1_1Transform.htm) getInverse() const;

99

109  [Vector2f](http://docs.google.com/classsf_1_1Vector2.htm) transformPoint(float x, float y) const;

110

119  [Vector2f](http://docs.google.com/classsf_1_1Vector2.htm) transformPoint(const [Vector2f](http://docs.google.com/classsf_1_1Vector2.htm)& point) const;

120

135  [FloatRect](http://docs.google.com/classsf_1_1Rect.htm) transformRect(const [FloatRect](http://docs.google.com/classsf_1_1Rect.htm)& rectangle) const;

136

149  [Transform](http://docs.google.com/classsf_1_1Transform.htm)& combine(const [Transform](http://docs.google.com/classsf_1_1Transform.htm)& transform);

150

169  [Transform](http://docs.google.com/classsf_1_1Transform.htm)& translate(float x, float y);

170

188  [Transform](http://docs.google.com/classsf_1_1Transform.htm)& translate(const [Vector2f](http://docs.google.com/classsf_1_1Vector2.htm)& offset);

189

207  [Transform](http://docs.google.com/classsf_1_1Transform.htm)& rotate(float angle);

208

233  [Transform](http://docs.google.com/classsf_1_1Transform.htm)& rotate(float angle, float centerX, float centerY);

234

258  [Transform](http://docs.google.com/classsf_1_1Transform.htm)& rotate(float angle, const [Vector2f](http://docs.google.com/classsf_1_1Vector2.htm)& center);

259

278  [Transform](http://docs.google.com/classsf_1_1Transform.htm)& scale(float scaleX, float scaleY);

279

305  [Transform](http://docs.google.com/classsf_1_1Transform.htm)& scale(float scaleX, float scaleY, float centerX, float centerY);

306

324  [Transform](http://docs.google.com/classsf_1_1Transform.htm)& scale(const [Vector2f](http://docs.google.com/classsf_1_1Vector2.htm)& factors);

325

349  [Transform](http://docs.google.com/classsf_1_1Transform.htm)& scale(const [Vector2f](http://docs.google.com/classsf_1_1Vector2.htm)& factors, const [Vector2f](http://docs.google.com/classsf_1_1Vector2.htm)& center);

350

352  // Static member data

[354](http://docs.google.com/classsf_1_1Transform.htm#aa4eb1eecbcb9979d76e2543b337fdb13)  static const [Transform](http://docs.google.com/classsf_1_1Transform.htm) [Identity](http://docs.google.com/classsf_1_1Transform.htm#aa4eb1eecbcb9979d76e2543b337fdb13);

355

356 private:

357

359  // Member data

361  float m\_matrix[16];

362 };

363

376 SFML\_GRAPHICS\_API [Transform](http://docs.google.com/classsf_1_1Transform.htm) operator \*(const [Transform](http://docs.google.com/classsf_1_1Transform.htm)& left, const [Transform](http://docs.google.com/classsf_1_1Transform.htm)& right);

377

390 SFML\_GRAPHICS\_API [Transform](http://docs.google.com/classsf_1_1Transform.htm)& operator \*=([Transform](http://docs.google.com/classsf_1_1Transform.htm)& left, const [Transform](http://docs.google.com/classsf_1_1Transform.htm)& right);

391

404 SFML\_GRAPHICS\_API [Vector2f](http://docs.google.com/classsf_1_1Vector2.htm) operator \*(const [Transform](http://docs.google.com/classsf_1_1Transform.htm)& left, const [Vector2f](http://docs.google.com/classsf_1_1Vector2.htm)& right);

405

406 } // namespace sf

407

408

409 #endif // SFML\_TRANSFORM\_HPP

410

411

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