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)

Transformable.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\_TRANSFORMABLE\_HPP

26 #define SFML\_TRANSFORMABLE\_HPP

27

29 // Headers

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

32 #include <SFML/Graphics/Transform.hpp>

33

34

35 namespace sf

36 {

[41](http://docs.google.com/classsf_1_1Transformable.htm) class SFML\_GRAPHICS\_API [Transformable](http://docs.google.com/classsf_1_1Transformable.htm)

42 {

43 public :

44

49  [Transformable](http://docs.google.com/classsf_1_1Transformable.htm)();

50

55  virtual ~[Transformable](http://docs.google.com/classsf_1_1Transformable.htm)();

56

70  void setPosition(float x, float y);

71

84  void setPosition(const [Vector2f](http://docs.google.com/classsf_1_1Vector2.htm)& position);

85

98  void setRotation(float angle);

99

113  void setScale(float factorX, float factorY);

114

127  void setScale(const [Vector2f](http://docs.google.com/classsf_1_1Vector2.htm)& factors);

128

145  void setOrigin(float x, float y);

146

162  void setOrigin(const [Vector2f](http://docs.google.com/classsf_1_1Vector2.htm)& origin);

163

172  const [Vector2f](http://docs.google.com/classsf_1_1Vector2.htm)& getPosition() const;

173

184  float getRotation() const;

185

194  const [Vector2f](http://docs.google.com/classsf_1_1Vector2.htm)& getScale() const;

195

204  const [Vector2f](http://docs.google.com/classsf_1_1Vector2.htm)& getOrigin() const;

205

223  void move(float offsetX, float offsetY);

224

240  void move(const [Vector2f](http://docs.google.com/classsf_1_1Vector2.htm)& offset);

241

255  void rotate(float angle);

256

274  void scale(float factorX, float factorY);

275

292  void scale(const [Vector2f](http://docs.google.com/classsf_1_1Vector2.htm)& factor);

293

302  const [Transform](http://docs.google.com/classsf_1_1Transform.htm)& getTransform() const;

303

312  const [Transform](http://docs.google.com/classsf_1_1Transform.htm)& getInverseTransform() const;

313

314 private :

315

317  // Member data

319  [Vector2f](http://docs.google.com/classsf_1_1Vector2.htm) m\_origin;

320  [Vector2f](http://docs.google.com/classsf_1_1Vector2.htm) m\_position;

321  float m\_rotation;

322  [Vector2f](http://docs.google.com/classsf_1_1Vector2.htm) m\_scale;

323  mutable [Transform](http://docs.google.com/classsf_1_1Transform.htm) m\_transform;

324  mutable bool m\_transformNeedUpdate;

325  mutable [Transform](http://docs.google.com/classsf_1_1Transform.htm) m\_inverseTransform;

326  mutable bool m\_inverseTransformNeedUpdate;

327 };

328

329 } // namespace sf

330

331

332 #endif // SFML\_TRANSFORMABLE\_HPP

333

334

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