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)
* [Window](http://docs.google.com/dir_91aff02cfffdbbdd31d48df547831556.htm)

Window/Window.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\_WINDOW\_HPP

26 #define SFML\_WINDOW\_HPP

27

29 // Headers

31 #include <SFML/Window/Export.hpp>

32 #include <SFML/Window/ContextSettings.hpp>

33 #include <SFML/Window/VideoMode.hpp>

34 #include <SFML/Window/WindowHandle.hpp>

35 #include <SFML/Window/WindowStyle.hpp>

36 #include <SFML/Window/GlResource.hpp>

37 #include <SFML/System/Clock.hpp>

38 #include <SFML/System/Vector2.hpp>

39 #include <SFML/System/NonCopyable.hpp>

40 #include <SFML/System/String.hpp>

41

42

43 namespace sf

44 {

45 namespace priv

46 {

47  class GlContext;

48  class WindowImpl;

49 }

50

51 class Event;

52

[57](http://docs.google.com/classsf_1_1Window.htm) class SFML\_WINDOW\_API [Window](http://docs.google.com/classsf_1_1Window.htm) : [GlResource](http://docs.google.com/classsf_1_1GlResource.htm), [NonCopyable](http://docs.google.com/classsf_1_1NonCopyable.htm)

58 {

59 public :

60

68  [Window](http://docs.google.com/classsf_1_1Window.htm)();

69

89  [Window](http://docs.google.com/classsf_1_1Window.htm)([VideoMode](http://docs.google.com/classsf_1_1VideoMode.htm) mode, const [String](http://docs.google.com/classsf_1_1String.htm)& title, Uint32 style = [Style::Default](http://docs.google.com/group__window.htm#gga8d7a3b8425c907a2872cb57e32cea5b8a5597cd420fc461807e4a201c92adea37), const [ContextSettings](http://docs.google.com/structsf_1_1ContextSettings.htm)& settings = [ContextSettings](http://docs.google.com/structsf_1_1ContextSettings.htm)());

90

105  explicit [Window](http://docs.google.com/classsf_1_1Window.htm)(WindowHandle handle, const [ContextSettings](http://docs.google.com/structsf_1_1ContextSettings.htm)& settings = [ContextSettings](http://docs.google.com/structsf_1_1ContextSettings.htm)());

106

113  virtual ~[Window](http://docs.google.com/classsf_1_1Window.htm)();

114

128  void create([VideoMode](http://docs.google.com/classsf_1_1VideoMode.htm) mode, const [String](http://docs.google.com/classsf_1_1String.htm)& title, Uint32 style = [Style::Default](http://docs.google.com/group__window.htm#gga8d7a3b8425c907a2872cb57e32cea5b8a5597cd420fc461807e4a201c92adea37), const [ContextSettings](http://docs.google.com/structsf_1_1ContextSettings.htm)& settings = [ContextSettings](http://docs.google.com/structsf_1_1ContextSettings.htm)());

129

141  void create(WindowHandle handle, const [ContextSettings](http://docs.google.com/structsf_1_1ContextSettings.htm)& settings = [ContextSettings](http://docs.google.com/structsf_1_1ContextSettings.htm)());

142

153  void close();

154

165  bool isOpen() const;

166

178  const [ContextSettings](http://docs.google.com/structsf_1_1ContextSettings.htm)& getSettings() const;

179

203  bool pollEvent([Event](http://docs.google.com/classsf_1_1Event.htm)& event);

204

230  bool waitEvent([Event](http://docs.google.com/classsf_1_1Event.htm)& event);

231

240  [Vector2i](http://docs.google.com/classsf_1_1Vector2.htm) getPosition() const;

241

254  void setPosition(const [Vector2i](http://docs.google.com/classsf_1_1Vector2.htm)& position);

255

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

268

277  void setSize(const [Vector2u](http://docs.google.com/classsf_1_1Vector2.htm) size);

278

287  void setTitle(const [String](http://docs.google.com/classsf_1_1String.htm)& title);

288

304  void setIcon(unsigned int width, unsigned int height, const Uint8\* pixels);

305

314  void setVisible(bool visible);

315

329  void setVerticalSyncEnabled(bool enabled);

330

339  void setMouseCursorVisible(bool visible);

340

353  void setKeyRepeatEnabled(bool enabled);

354

370  void setFramerateLimit(unsigned int limit);

371

383  void setJoystickThreshold(float threshold);

384

400  bool setActive(bool active = true) const;

401

410  void display();

411

424  WindowHandle getSystemHandle() const;

425

426 protected :

427

436  virtual void onCreate();

437

445  virtual void onResize();

446

447 private:

448

461  bool filterEvent(const [Event](http://docs.google.com/classsf_1_1Event.htm)& event);

462

467  void initialize();

468

470  // Member data

472  priv::WindowImpl\* m\_impl;

473  priv::GlContext\* m\_context;

474  [Clock](http://docs.google.com/classsf_1_1Clock.htm) m\_clock;

475  [Time](http://docs.google.com/classsf_1_1Time.htm) m\_frameTimeLimit;

476 };

477

478 } // namespace sf

479

480

481 #endif // SFML\_WINDOW\_HPP

482

483

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