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)

Config.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\_CONFIG\_HPP

26 #define SFML\_CONFIG\_HPP

27

28

30 // Define the SFML version

32 #define SFML\_VERSION\_MAJOR 2

33 #define SFML\_VERSION\_MINOR 0

34

35

37 // Identify the operating system

39 #if defined(\_WIN32) || defined(\_\_WIN32\_\_)

40

41  // Windows

42  #define SFML\_SYSTEM\_WINDOWS

43  #ifndef NOMINMAX

44  #define NOMINMAX

45  #endif

46

47 #elif defined(linux) || defined(\_\_linux)

48

49  // Linux

50  #define SFML\_SYSTEM\_LINUX

51

52 #elif defined(\_\_APPLE\_\_) || defined(MACOSX) || defined(macintosh) || defined(Macintosh)

53

54  // MacOS

55  #define SFML\_SYSTEM\_MACOS

56

57 #elif defined(\_\_FreeBSD\_\_) || defined(\_\_FreeBSD\_kernel\_\_)

58

59  // FreeBSD

60  #define SFML\_SYSTEM\_FREEBSD

61

62 #else

63

64  // Unsupported system

65  #error This operating system is not supported by SFML library

66

67 #endif

68

69

71 // Define a portable debug macro

73 #if !defined(NDEBUG)

74

75  #define SFML\_DEBUG

76

77 #endif

78

79

81 // Define helpers to create portable import / export macros for each module

83 #if !defined(SFML\_STATIC)

84

85  #if defined(SFML\_SYSTEM\_WINDOWS)

86

87  // Windows compilers need specific (and different) keywords for export and import

88  #define SFML\_API\_EXPORT \_\_declspec(dllexport)

89  #define SFML\_API\_IMPORT \_\_declspec(dllimport)

90

91  // For Visual C++ compilers, we also need to turn off this annoying C4251 warning

92  #ifdef \_MSC\_VER

93

94  #pragma warning(disable : 4251)

95

96  #endif

97

98  #else // Linux, FreeBSD, Mac OS X

99

100  #if \_\_GNUC\_\_ >= 4

101

102  // GCC 4 has special keywords for showing/hidding symbols,

103  // the same keyword is used for both importing and exporting

104  #define SFML\_API\_EXPORT \_\_attribute\_\_ ((\_\_visibility\_\_ ("default")))

105  #define SFML\_API\_IMPORT \_\_attribute\_\_ ((\_\_visibility\_\_ ("default")))

106

107  #else

108

109  // GCC < 4 has no mechanism to explicitely hide symbols, everything's exported

110  #define SFML\_API\_EXPORT

111  #define SFML\_API\_IMPORT

112

113  #endif

114

115  #endif

116

117 #else

118

119  // Static build doesn't need import/export macros

120  #define SFML\_API\_EXPORT

121  #define SFML\_API\_IMPORT

122

123 #endif

124

125

127 // Define portable fixed-size types

129 namespace sf

130 {

131  // All "common" platforms use the same size for char, short and int

132  // (basically there are 3 types for 3 sizes, so no other match is possible),

133  // we can use them without doing any kind of check

134

135  // 8 bits integer types

136  typedef signed char Int8;

137  typedef unsigned char Uint8;

138

139  // 16 bits integer types

140  typedef signed short Int16;

141  typedef unsigned short Uint16;

142

143  // 32 bits integer types

144  typedef signed int Int32;

145  typedef unsigned int Uint32;

146

147  // 64 bits integer types

148  #if defined(\_MSC\_VER)

149  typedef signed \_\_int64 Int64;

150  typedef unsigned \_\_int64 Uint64;

151  #else

152  typedef signed long long Int64;

153  typedef unsigned long long Uint64;

154  #endif

155

156 } // namespace sf

157

158

159 #endif // SFML\_CONFIG\_HPP

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