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)
* [Class List](http://docs.google.com/annotated.htm)
* [Class Index](http://docs.google.com/classes.htm)
* [Class Hierarchy](http://docs.google.com/hierarchy.htm)
* [Class Members](http://docs.google.com/functions.htm)
* **sf**
* [NonCopyable](http://docs.google.com/classsf_1_1NonCopyable.htm)

[Protected Member Functions](#_gjdgxs) | [List of all members](http://docs.google.com/classsf_1_1NonCopyable-members.htm)

sf::NonCopyable Class Reference

[System module](http://docs.google.com/group__system.htm)

Utility class that makes any derived class non-copyable. [More...](http://docs.google.com/classsf_1_1NonCopyable.htm#details)

#include <[NonCopyable.hpp](http://docs.google.com/NonCopyable_8hpp_source.htm)>

Inheritance diagram for sf::NonCopyable:



| Protected Member Functions | |
| --- | --- |
|  | [NonCopyable](http://docs.google.com/classsf_1_1NonCopyable.htm#a2110add170580fdb946f887719da6860) () |
|  | Default constructor. |
|  | |

## Detailed Description

Utility class that makes any derived class non-copyable.

This class makes its instances non-copyable, by explicitely disabling its copy constructor and its assignment operator.

To create a non-copyable class, simply inherit from [sf::NonCopyable](http://docs.google.com/classsf_1_1NonCopyable.htm).

The type of inheritance (public or private) doesn't matter, the copy constructor and assignment operator are declared private in [sf::NonCopyable](http://docs.google.com/classsf_1_1NonCopyable.htm) so they will end up being inaccessible in both cases. Thus you can use a shorter syntax for inheriting from it (see below).

Usage example:

class MyNonCopyableClass : [sf::NonCopyable](http://docs.google.com/classsf_1_1NonCopyable.htm)

{

...

};

Deciding whether the instances of a class can be copied or not is a very important design choice. You are strongly encouraged to think about it before writing a class, and to use [sf::NonCopyable](http://docs.google.com/classsf_1_1NonCopyable.htm) when necessary to prevent many potential future errors when using it. This is also a very important indication to users of your class.

Definition at line [41](http://docs.google.com/NonCopyable_8hpp_source.htm#l00041) of file [NonCopyable.hpp](http://docs.google.com/NonCopyable_8hpp_source.htm).

## Constructor & Destructor Documentation

| | sf::NonCopyable::NonCopyable | ( |  | ) |  | | --- | --- | --- | --- | --- | | inlineprotected |
| --- | --- | --- | --- | --- | --- | --- |

Default constructor.

Because this class has a copy constructor, the compiler will not automatically generate the default constructor. That's why we must define it explicitely.

Definition at line [53](http://docs.google.com/NonCopyable_8hpp_source.htm#l00053) of file [NonCopyable.hpp](http://docs.google.com/NonCopyable_8hpp_source.htm).

The documentation for this class was generated from the following file:

* [NonCopyable.hpp](http://docs.google.com/NonCopyable_8hpp_source.htm)

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