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**
* [IpAddress](http://docs.google.com/classsf_1_1IpAddress.htm)

[Public Member Functions](#_gjdgxs) | [Static Public Member Functions](#_30j0zll) | [Static Public Attributes](#_1fob9te) | [List of all members](http://docs.google.com/classsf_1_1IpAddress-members.htm)

sf::IpAddress Class Reference

[Network module](http://docs.google.com/group__network.htm)

Encapsulate an IPv4 network address. [More...](http://docs.google.com/classsf_1_1IpAddress.htm#details)

#include <[IpAddress.hpp](http://docs.google.com/IpAddress_8hpp_source.htm)>

| Public Member Functions | |
| --- | --- |
|  | [IpAddress](http://docs.google.com/classsf_1_1IpAddress.htm#af32a0574baa0f46e48deb2d83ca7658b) () |
|  | Default constructor. |
|  | |
|  | [IpAddress](http://docs.google.com/classsf_1_1IpAddress.htm#a656b7445ab04cabaa7398685bc09c3f7) (const std::string &address) |
|  | Construct the address from a string. |
|  | |
|  | [IpAddress](http://docs.google.com/classsf_1_1IpAddress.htm#a92f2a9be74334a61b96c2fc79fe6eb78) (const char \*address) |
|  | Construct the address from a string. |
|  | |
|  | [IpAddress](http://docs.google.com/classsf_1_1IpAddress.htm#a1d289dcb9ce7a64c600c6f84cba88cc6) (Uint8 byte0, Uint8 byte1, Uint8 byte2, Uint8 byte3) |
|  | Construct the address from 4 bytes. |
|  | |
|  | [IpAddress](http://docs.google.com/classsf_1_1IpAddress.htm#a8ed34ba3a40d70eb9f09ac5ae779a162) (Uint32 address) |
|  | Construct the address from a 32-bits integer. |
|  | |
| std::string | [toString](http://docs.google.com/classsf_1_1IpAddress.htm#a52f4be92fb0ceb689abc469e4a85fd82) () const |
|  | Get a string representation of the address. |
|  | |
| Uint32 | [toInteger](http://docs.google.com/classsf_1_1IpAddress.htm#af42678b08b23def2560aed7d98b24d89) () const |
|  | Get an integer representation of the address. |
|  | |

| Static Public Member Functions | |
| --- | --- |
| static [IpAddress](http://docs.google.com/classsf_1_1IpAddress.htm) | [getLocalAddress](http://docs.google.com/classsf_1_1IpAddress.htm#a4c31622ad87edca48adbb8e8ed00ee4a) () |
|  | Get the computer's local address. |
|  | |
| static [IpAddress](http://docs.google.com/classsf_1_1IpAddress.htm) | [getPublicAddress](http://docs.google.com/classsf_1_1IpAddress.htm#a5c5cbf67e4aacf23c24f2ad991df4c55) ([Time](http://docs.google.com/classsf_1_1Time.htm) timeout=[Time::Zero](http://docs.google.com/classsf_1_1Time.htm#a8db127b632fa8da21550e7282af11fa0)) |
|  | Get the computer's public address. |
|  | |

| Static Public Attributes | |
| --- | --- |
| static const [IpAddress](http://docs.google.com/classsf_1_1IpAddress.htm) | [None](http://docs.google.com/classsf_1_1IpAddress.htm#a4619b4abbe3c8fef056e7299db967404) |
|  | Value representing an empty/invalid address. |
|  | |
| static const [IpAddress](http://docs.google.com/classsf_1_1IpAddress.htm) | [LocalHost](http://docs.google.com/classsf_1_1IpAddress.htm#a594d3a8e2559f8fa8ab0a96fa597333b) |
|  | The "localhost" address (for connecting a computer to itself locally) |
|  | |
| static const [IpAddress](http://docs.google.com/classsf_1_1IpAddress.htm) | [Broadcast](http://docs.google.com/classsf_1_1IpAddress.htm#aa93d1d57b65d243f2baf804b6035465c) |
|  | The "broadcast" address (for sending UDP messages to everyone on a local network) |
|  | |

## Detailed Description

Encapsulate an IPv4 network address.

[sf::IpAddress](http://docs.google.com/classsf_1_1IpAddress.htm) is a utility class for manipulating network addresses.

It provides a set a implicit constructors and conversion functions to easily build or transform an IP address from/to various representations.

Usage example:

[sf::IpAddress](http://docs.google.com/classsf_1_1IpAddress.htm) a0; // an invalid address

[sf::IpAddress](http://docs.google.com/classsf_1_1IpAddress.htm) a1 = [sf::IpAddress::None](http://docs.google.com/classsf_1_1IpAddress.htm#a4619b4abbe3c8fef056e7299db967404); // an invalid address (same as a0)

[sf::IpAddress](http://docs.google.com/classsf_1_1IpAddress.htm) a2("127.0.0.1"); // the local host address

[sf::IpAddress](http://docs.google.com/classsf_1_1IpAddress.htm) a3 = [sf::IpAddress::Broadcast](http://docs.google.com/classsf_1_1IpAddress.htm#aa93d1d57b65d243f2baf804b6035465c); // the broadcast address

[sf::IpAddress](http://docs.google.com/classsf_1_1IpAddress.htm) a4(192, 168, 1, 56); // a local address

[sf::IpAddress](http://docs.google.com/classsf_1_1IpAddress.htm) a5("my\_computer"); // a local address created from a network name

[sf::IpAddress](http://docs.google.com/classsf_1_1IpAddress.htm) a6("89.54.1.169"); // a distant address

[sf::IpAddress](http://docs.google.com/classsf_1_1IpAddress.htm) a7("www.google.com"); // a distant address created from a network name

[sf::IpAddress](http://docs.google.com/classsf_1_1IpAddress.htm) a8 = [sf::IpAddress::getLocalAddress](http://docs.google.com/classsf_1_1IpAddress.htm#a4c31622ad87edca48adbb8e8ed00ee4a)(); // my address on the local network

[sf::IpAddress](http://docs.google.com/classsf_1_1IpAddress.htm) a9 = [sf::IpAddress::getPublicAddress](http://docs.google.com/classsf_1_1IpAddress.htm#a5c5cbf67e4aacf23c24f2ad991df4c55)(); // my address on the internet

Note that [sf::IpAddress](http://docs.google.com/classsf_1_1IpAddress.htm) currently doesn't support IPv6 nor other types of network addresses.

Definition at line [44](http://docs.google.com/IpAddress_8hpp_source.htm#l00044) of file [IpAddress.hpp](http://docs.google.com/IpAddress_8hpp_source.htm).

## Constructor & Destructor Documentation

| sf::IpAddress::IpAddress | ( |  | ) |  |
| --- | --- | --- | --- | --- |

Default constructor.

This constructor creates an empty (invalid) address

| sf::IpAddress::IpAddress | ( | const std::string & | *address* | ) |  |
| --- | --- | --- | --- | --- | --- |

Construct the address from a string.

Here *address* can be either a decimal address (ex: "192.168.1.56") or a network name (ex: "localhost").

Parameters

| address | IP address or network name |
| --- | --- |

| sf::IpAddress::IpAddress | ( | const char \* | *address* | ) |  |
| --- | --- | --- | --- | --- | --- |

Construct the address from a string.

Here *address* can be either a decimal address (ex: "192.168.1.56") or a network name (ex: "localhost"). This is equivalent to the constructor taking a std::string parameter, it is defined for convenience so that the implicit conversions from literal strings to [IpAddress](http://docs.google.com/classsf_1_1IpAddress.htm) work.

Parameters

| address | IP address or network name |
| --- | --- |

| sf::IpAddress::IpAddress | ( | Uint8 | *byte0*, |
| --- | --- | --- | --- |
|  |  | Uint8 | *byte1*, |
|  |  | Uint8 | *byte2*, |
|  |  | Uint8 | *byte3* |
|  | ) |  |  |

Construct the address from 4 bytes.

Calling IpAddress(a, b, c, d) is equivalent to calling [IpAddress](http://docs.google.com/classsf_1_1IpAddress.htm)("a.b.c.d"), but safer as it doesn't have to parse a string to get the address components.

Parameters

| byte0 | First byte of the address |
| --- | --- |
| byte1 | Second byte of the address |
| byte2 | Third byte of the address |
| byte3 | Fourth byte of the address |

| | sf::IpAddress::IpAddress | ( | Uint32 | *address* | ) |  | | --- | --- | --- | --- | --- | --- | | explicit |
| --- | --- | --- | --- | --- | --- | --- | --- |

Construct the address from a 32-bits integer.

This constructor uses the internal representation of the address directly. It should be used for optimization purposes, and only if you got that representation from IpAddress::ToInteger().

Parameters

| address | 4 bytes of the address packed into a 32-bits integer |
| --- | --- |

See Also[toInteger](http://docs.google.com/classsf_1_1IpAddress.htm#af42678b08b23def2560aed7d98b24d89)

## Member Function Documentation

| | static [IpAddress](http://docs.google.com/classsf_1_1IpAddress.htm) sf::IpAddress::getLocalAddress | ( |  | ) |  | | --- | --- | --- | --- | --- | | static |
| --- | --- | --- | --- | --- | --- | --- |

Get the computer's local address.

The local address is the address of the computer from the LAN point of view, i.e. something like 192.168.1.56. It is meaningful only for communications over the local network. Unlike getPublicAddress, this function is fast and may be used safely anywhere.

ReturnsLocal IP address of the computer See Also[getPublicAddress](http://docs.google.com/classsf_1_1IpAddress.htm#a5c5cbf67e4aacf23c24f2ad991df4c55)

| | static [IpAddress](http://docs.google.com/classsf_1_1IpAddress.htm) sf::IpAddress::getPublicAddress | ( | [Time](http://docs.google.com/classsf_1_1Time.htm) | *timeout* = [Time::Zero](http://docs.google.com/classsf_1_1Time.htm#a8db127b632fa8da21550e7282af11fa0) | ) |  | | --- | --- | --- | --- | --- | --- | | static |
| --- | --- | --- | --- | --- | --- | --- | --- |

Get the computer's public address.

The public address is the address of the computer from the internet point of view, i.e. something like 89.54.1.169. It is necessary for communications over the world wide web. The only way to get a public address is to ask it to a distant website; as a consequence, this function depends on both your network connection and the server, and may be very slow. You should use it as few as possible. Because this function depends on the network connection and on a distant server, you may use a time limit if you don't want your program to be possibly stuck waiting in case there is a problem; this limit is deactivated by default.

Parameters

| timeout | Maximum time to wait |
| --- | --- |

ReturnsPublic IP address of the computer See Also[getLocalAddress](http://docs.google.com/classsf_1_1IpAddress.htm#a4c31622ad87edca48adbb8e8ed00ee4a)

| Uint32 sf::IpAddress::toInteger | ( |  | ) | const |
| --- | --- | --- | --- | --- |

Get an integer representation of the address.

The returned number is the internal representation of the address, and should be used for optimization purposes only (like sending the address through a socket). The integer produced by this function can then be converted back to a [sf::IpAddress](http://docs.google.com/classsf_1_1IpAddress.htm) with the proper constructor.

Returns32-bits unsigned integer representation of the address See Also[toString](http://docs.google.com/classsf_1_1IpAddress.htm#a52f4be92fb0ceb689abc469e4a85fd82)

| std::string sf::IpAddress::toString | ( |  | ) | const |
| --- | --- | --- | --- | --- |

Get a string representation of the address.

The returned string is the decimal representation of the IP address (like "192.168.1.56"), even if it was constructed from a host name.

Returns[String](http://docs.google.com/classsf_1_1String.htm) representation of the address See Also[toInteger](http://docs.google.com/classsf_1_1IpAddress.htm#af42678b08b23def2560aed7d98b24d89)

## Member Data Documentation

| | const [IpAddress](http://docs.google.com/classsf_1_1IpAddress.htm) sf::IpAddress::Broadcast | | --- | | static |
| --- | --- | --- |

The "broadcast" address (for sending UDP messages to everyone on a local network)

Definition at line [186](http://docs.google.com/IpAddress_8hpp_source.htm#l00186) of file [IpAddress.hpp](http://docs.google.com/IpAddress_8hpp_source.htm).

| | const [IpAddress](http://docs.google.com/classsf_1_1IpAddress.htm) sf::IpAddress::LocalHost | | --- | | static |
| --- | --- | --- |

The "localhost" address (for connecting a computer to itself locally)

Definition at line [185](http://docs.google.com/IpAddress_8hpp_source.htm#l00185) of file [IpAddress.hpp](http://docs.google.com/IpAddress_8hpp_source.htm).

| | const [IpAddress](http://docs.google.com/classsf_1_1IpAddress.htm) sf::IpAddress::None | | --- | | static |
| --- | --- | --- |

Value representing an empty/invalid address.

Definition at line [184](http://docs.google.com/IpAddress_8hpp_source.htm#l00184) of file [IpAddress.hpp](http://docs.google.com/IpAddress_8hpp_source.htm).

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

* [IpAddress.hpp](http://docs.google.com/IpAddress_8hpp_source.htm)

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