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**
* [ThreadLocalPtr](http://docs.google.com/classsf_1_1ThreadLocalPtr.htm)

[Public Member Functions](#_gjdgxs) | [Private Member Functions](#_30j0zll) | [List of all members](http://docs.google.com/classsf_1_1ThreadLocalPtr-members.htm)

sf::ThreadLocalPtr< T > Class Template Reference

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

Pointer to a thread-local variable. [More...](http://docs.google.com/classsf_1_1ThreadLocalPtr.htm#details)

#include <[ThreadLocalPtr.hpp](http://docs.google.com/ThreadLocalPtr_8hpp_source.htm)>

Inheritance diagram for sf::ThreadLocalPtr< T >:



| Public Member Functions | |
| --- | --- |
|  | [ThreadLocalPtr](http://docs.google.com/classsf_1_1ThreadLocalPtr.htm#a8c678211d7828d2a8c41cb534422d649) (T \*value=NULL) |
|  | Default constructor. |
|  | |
| T & | [operator\*](http://docs.google.com/classsf_1_1ThreadLocalPtr.htm#aa3bac9a08e8739613961659d10e0fadd) () const |
|  | Overload of unary operator \*. |
|  | |
| T \* | [operator->](http://docs.google.com/classsf_1_1ThreadLocalPtr.htm#aa0b559f78929b22cb2585cb2966edfb2) () const |
|  | Overload of operator -> |
|  | |
|  | [operator T \*](http://docs.google.com/classsf_1_1ThreadLocalPtr.htm#ab4a6a341c26b58f0ed3ef86502bd9572) () const |
|  | Cast operator to implicitely convert the pointer to its raw pointer type (T\*) |
|  | |
| [ThreadLocalPtr](http://docs.google.com/classsf_1_1ThreadLocalPtr.htm)< T > & | [operator=](http://docs.google.com/classsf_1_1ThreadLocalPtr.htm#a14dcf1cdf5f6b3bcdd633014b2b671f5) (T \*value) |
|  | Assignment operator for a raw pointer parameter. |
|  | |
| [ThreadLocalPtr](http://docs.google.com/classsf_1_1ThreadLocalPtr.htm)< T > & | [operator=](http://docs.google.com/classsf_1_1ThreadLocalPtr.htm#a6792a6a808af06f0d13e3ceecf2fc947) (const [ThreadLocalPtr](http://docs.google.com/classsf_1_1ThreadLocalPtr.htm)< T > &right) |
|  | Assignment operator for a [ThreadLocalPtr](http://docs.google.com/classsf_1_1ThreadLocalPtr.htm) parameter. |
|  | |

| Private Member Functions | |
| --- | --- |
| void | [setValue](http://docs.google.com/classsf_1_1ThreadLocal.htm#ab7e334c83d77644a8e67ee31c3230007) (void \*value) |
|  | Set the thread-specific value of the variable. |
|  | |
| void \* | [getValue](http://docs.google.com/classsf_1_1ThreadLocal.htm#aef35a39686eac4b6634a6e5605aacfd7) () const |
|  | Retrieve the thread-specific value of the variable. |
|  | |

## Detailed Description

template<typename T>

class sf::ThreadLocalPtr< T >

Pointer to a thread-local variable.

[sf::ThreadLocalPtr](http://docs.google.com/classsf_1_1ThreadLocalPtr.htm) is a type-safe wrapper for storing pointers to thread-local variables.

A thread-local variable holds a different value for each different thread, unlike normal variable that are shared.

Its usage is completely transparent, so that it is similar to manipulating the raw pointer directly (like any smart pointer).

Usage example:

MyClass object1;

MyClass object2;

[sf::ThreadLocalPtr<MyClass>](http://docs.google.com/classsf_1_1ThreadLocalPtr.htm) objectPtr;

void thread1()

{

objectPtr = &object1; // doesn't impact thread2

...

}

void thread2()

{

objectPtr = &object2; // doesn't impact thread1

...

}

int main()

{

// Create and launch the two threads

[sf::Thread](http://docs.google.com/classsf_1_1Thread.htm) t1(&thread1);

[sf::Thread](http://docs.google.com/classsf_1_1Thread.htm) t2(&thread2);

t1.launch();

t2.launch();

return 0;

}

[ThreadLocalPtr](http://docs.google.com/classsf_1_1ThreadLocalPtr.htm) is designed for internal use; however you can use it if you feel like it fits well your implementation.

Definition at line [41](http://docs.google.com/ThreadLocalPtr_8hpp_source.htm#l00041) of file [ThreadLocalPtr.hpp](http://docs.google.com/ThreadLocalPtr_8hpp_source.htm).

## Constructor & Destructor Documentation

template<typename T >

| [sf::ThreadLocalPtr](http://docs.google.com/classsf_1_1ThreadLocalPtr.htm)< T >::[ThreadLocalPtr](http://docs.google.com/classsf_1_1ThreadLocalPtr.htm) | ( | T \* | *value* = NULL | ) |  |
| --- | --- | --- | --- | --- | --- |

Default constructor.

Parameters

| value | Optional value to initalize the variable |
| --- | --- |

Definition at line [30](http://docs.google.com/ThreadLocalPtr_8inl_source.htm#l00030) of file [ThreadLocalPtr.inl](http://docs.google.com/ThreadLocalPtr_8inl_source.htm).

## Member Function Documentation

template<typename T >

| [sf::ThreadLocalPtr](http://docs.google.com/classsf_1_1ThreadLocalPtr.htm)< T >::operator T \* | ( |  | ) | const |
| --- | --- | --- | --- | --- |

Cast operator to implicitely convert the pointer to its raw pointer type (T\*)

ReturnsPointer to the actual object

Definition at line [54](http://docs.google.com/ThreadLocalPtr_8inl_source.htm#l00054) of file [ThreadLocalPtr.inl](http://docs.google.com/ThreadLocalPtr_8inl_source.htm).

template<typename T >

| T & [sf::ThreadLocalPtr](http://docs.google.com/classsf_1_1ThreadLocalPtr.htm)< T >::operator\* | ( |  | ) | const |
| --- | --- | --- | --- | --- |

Overload of unary operator \*.

Like raw pointers, applying the \* operator returns a reference to the pointed object.

ReturnsReference to the pointed object

Definition at line [38](http://docs.google.com/ThreadLocalPtr_8inl_source.htm#l00038) of file [ThreadLocalPtr.inl](http://docs.google.com/ThreadLocalPtr_8inl_source.htm).

template<typename T >

| T \* [sf::ThreadLocalPtr](http://docs.google.com/classsf_1_1ThreadLocalPtr.htm)< T >::operator-> | ( |  | ) | const |
| --- | --- | --- | --- | --- |

Overload of operator ->

Like raw pointers, applying the -> operator returns the pointed object.

ReturnsPointed object

Definition at line [46](http://docs.google.com/ThreadLocalPtr_8inl_source.htm#l00046) of file [ThreadLocalPtr.inl](http://docs.google.com/ThreadLocalPtr_8inl_source.htm).

template<typename T >

| [ThreadLocalPtr](http://docs.google.com/classsf_1_1ThreadLocalPtr.htm)< T > & [sf::ThreadLocalPtr](http://docs.google.com/classsf_1_1ThreadLocalPtr.htm)< T >::operator= | ( | T \* | *value* | ) |  |
| --- | --- | --- | --- | --- | --- |

Assignment operator for a raw pointer parameter.

Parameters

| value | Pointer to assign |
| --- | --- |

ReturnsReference to self

Definition at line [62](http://docs.google.com/ThreadLocalPtr_8inl_source.htm#l00062) of file [ThreadLocalPtr.inl](http://docs.google.com/ThreadLocalPtr_8inl_source.htm).

template<typename T >

| [ThreadLocalPtr](http://docs.google.com/classsf_1_1ThreadLocalPtr.htm)< T > & [sf::ThreadLocalPtr](http://docs.google.com/classsf_1_1ThreadLocalPtr.htm)< T >::operator= | ( | const [ThreadLocalPtr](http://docs.google.com/classsf_1_1ThreadLocalPtr.htm)< T > & | *right* | ) |  |
| --- | --- | --- | --- | --- | --- |

Assignment operator for a [ThreadLocalPtr](http://docs.google.com/classsf_1_1ThreadLocalPtr.htm) parameter.

Parameters

| right | [ThreadLocalPtr](http://docs.google.com/classsf_1_1ThreadLocalPtr.htm) to assign |
| --- | --- |

ReturnsReference to self

Definition at line [71](http://docs.google.com/ThreadLocalPtr_8inl_source.htm#l00071) of file [ThreadLocalPtr.inl](http://docs.google.com/ThreadLocalPtr_8inl_source.htm).

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

* [ThreadLocalPtr.hpp](http://docs.google.com/ThreadLocalPtr_8hpp_source.htm)
* [ThreadLocalPtr.inl](http://docs.google.com/ThreadLocalPtr_8inl_source.htm)

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