Threading Class

1.0

Generated by Doxygen 1.8.7

Sun Jul 13 2014 22:44:34

Contents

Chapter 1

Class Index

1.1 Class List

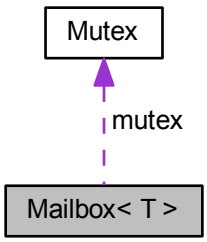
Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 2

Class Documentation

2.1 Mailbox< T > Class Template Reference

Collaboration diagram for Mailbox< T >:



Public Member Functions

- Mailbox ()
- Mailbox (int size)
- int SetSize (int size)
- int PutMessage (T message)
- bool IsEmpty ()
- bool IsFull ()
- T GetMessage ()

Protected Attributes

- std::vector< T > m_mailBox
- int m_maxSize
- Mutex mutex

2.1.1 Constructor & Destructor Documentation

2.1.1.1 template<class T > Mailbox< T >::Mailbox ( )

Create the mailbox with no size limet

2.1.1.2 template<class T > Mailbox< T >::Mailbox (int *size*)

Parameters

size	The maximum size of the mailbox 0 for an unlimmeted size

2.1.2 Member Function Documentation

2.1.2.1 template<class T > T Mailbox< T >::GetMessage ( )

Gives back the last message run IsEmpty() first!

Returns

the last message

Warning

if there is no massage in the buffer it returns random vallues

2.1.2.2 template<class T > bool Mailbox< T >::IsEmpty ( )

Checks if the mailbox is empty

Returns

true if empty else false

2.1.2.3 template<class T > bool Mailbox< T >::IsFull ( )

Checks if the mailbox is full

Returns

true if full else false

2.1.2.4 template<class T > int Mailbox< T >::PutMessage (T *message*)

Add a message to the mailbox its smart to run IsFull() first

Parameters

message	the message that has to be mailt to the other thread

Returns

0 on success else >0

2.1.2.5 template<class T > int Mailbox< T >::SetSize (int *size*)

Set the size of the mailbox

Parameters

size	the maximum size of the mailbox

Returns

0 on success else >0

2.1.3 Member Data Documentation

2.1.3.1 template<class T > std::vector<T> Mailbox< T >::m_mailBox [protected]

We use a vector to store the messages

2.1.3.2 template<class T > int Mailbox< T >::m_maxSize [protected]

The maximum size of the mailbox

2.1.3.3 template<class T > Mutex Mailbox< T >::mutex [protected]

The mutex to make it threadsafe

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

- mailbox.h

2.2 Mutex Class Reference

Public Member Functions

- int Lock ()
- int Unlock ()

2.2.1 Member Function Documentation

2.2.1.1 int Mutex::Lock ( )

Lock the mutex

Returns

0 on success else >0

2.2.1.2 int Mutex::Unlock ( )

Lock the mutex

Returns

0 on success else >0

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

- mutex.h

2.3 Thread Class Reference

Public Member Functions

- int SetThreadFunction (int(*p_function)(void *))
- int SetParam (void *p_param)
- int Start ()
- int Stop ()
- int GetThreadId ()

Protected Attributes

- void * mp_param
- int(* mp_function)(void *)

2.3.1 Member Function Documentation

2.3.1.1 int Thread::GetThreadId ( )

Warning

this function is not implemented

Returns

The thread ID

2.3.1.2 int Thread::SetParam (void * *p_param*)

Set the parameters that have to be send to the thread

Returns

0 on success else >07-2014

2.3.1.3 int Thread::SetThreadFunction (int(*)(void *) *p_function*)

Set the function that has to run and tell if it has to run in a while (true) loop

Parameters

p_function	the pointer to the function that has to run as a thread format (int FunctionName (void*))

Returns

0 on success else >0

2.3.1.4 int Thread::Start ( )

Start the thread

Returns

0 on success else >0

2.3.1.5 int Thread::Stop ( )

Stop the thread

Returns

0 on success else >0

2.3.2 Member Data Documentation

2.3.2.1 int(* Thread::mp_function)(void *) [protected]

A pointer to the function that has to run as a thread

2.3.2.2 void* Thread::mp_param [protected]

A pointer to the param that has to be send to the thread

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

- thread.h
- thread_general.cpp

Index