LittleBenchmark_HDD

0.10.6

Generated by Doxygen 1.6.3

Tue Oct 5 21:37:45 2010

Contents

Chapter 1

Class Index

1.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Chapter 2

Class Index

2.1 Class List

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

Chapter 3

Class Documentation

3.1 profileUpdateStore Class Reference

Public Member Functions

- **profileUpdateStore** (string, string="")
- string **GetstrVar** ()
- string **GetstrVal** ()
- string **GetIt** ()

Friends

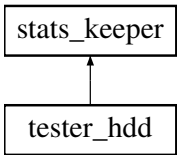
- template<typename T >
  std::ostream & **operator**<< (T &Output, profileUpdateStore &w)

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

- profileUpdateStore.hpp
- profileUpdateStore.cpp

3.2 stats_keeper Class Reference

Inheritance diagram for stats_keeper:



Public Member Functions

- stats_keeper ()

  Constractor set all args with default values.

- void addStatData (string *, string *, unsigned=1, unsigned=1)

  Will add data to specified buffers.

- uint8_t **findAndAdd** (string *, string *, unsigned=1, unsigned=1)
- void **FormatDataInVector** ()
- void **SaveToDisk** (boost::filesystem::path)
- string GeneratDataFromVector ()

  Return refernce to string value generated from vector.

- string **GenerateXMLFromVector** ()

Protected Member Functions

- void setArgs ()

  Set Args determinates functionality.

Protected Attributes

- vector< string > * p_vecstr_Log

  keep log output in vector

- vector< vector_str > * p_vecstr_formattedTXT

  keep data in vector

- bool bLog

  To log.

- bool bFormattedTxt

  Generate formatted txt.

- bool bGenXML

  Generate xml.

- unsigned uiMultiplySpacer

  Multiply spacer.

- unsigned uiMaxCols

  Max columns.

- string strSpacerChar

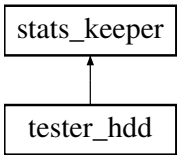
  spacer character

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

- stats_keeper.hpp
- stats_keeper.cpp

3.3 tester_hdd Class Reference

Inheritance diagram for tester_hdd:



Public Member Functions

- tester_hdd (int, char **)
- void Run ()

Public Attributes

- bool **bRun**

3.3.1 Constructor & Destructor Documentation

3.3.1.1 tester_hdd::tester_hdd (int *ac*, char ** *av*)

Important notes for Linux: * In file /etc/nsswitch.conf change passwd compat to passwd file in order to prevent memory leak

Creates user profile before parsing args

Number of columns

Set output args

3.3.2 Member Function Documentation

3.3.2.1 void tester_hdd::Run ()

Create threads and insert it to list

Running threads from list

If multithreading is disable program will wait for thread to do its job before running next one

With multithreading wating for all started threads to end its jobs

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

- tester_hdd.hpp
- tester_hdd.cpp

3.4 myThreadTemplates::thread_1< ClassT > Class Template Reference

Thread_1 is very simple thread template which constainst few controle methods and entry for statistics.

`#include <myThreadTemplates.hpp>`

Public Member Functions

- **thread_1** (ClassT *parent, list< string * > *list=NULL, bool show=false)
- void start ()
- void start_self_test ()
- void join ()
- void join (unsigned val)
- string **GetThreadID** ()
- void UpdateStats (string *str)
- virtual ~thread_1 ()

Static Public Member Functions

- static void self_test ()
- static void Execute_ (ClassT *p)

3.4.1 Detailed Description

template<class ClassT> class myThreadTemplates::thread_1< ClassT >

Thread_1 is very simple thread template which constainst few controle methods and entry for statistics.

3.4.2 Constructor & Destructor Documentation

3.4.2.1 template<class ClassT> virtual myThreadTemplates::thread_1< ClassT >::~thread_1 () [inline, virtual]

Virtual destructor which can show time of execution

3.4.3 Member Function Documentation

3.4.3.1 template<class ClassT> static void myThreadTemplates::thread_1< ClassT >::Execute_ (ClassT * *p*) [inline, static]

Static linker for dynamic method

3.4.3.2 template<class ClassT> void myThreadTemplates::thread_1< ClassT >::join (unsigned *val*) [inline]

Join thread after specified time in seconds

3.4.3.3 template<class ClassT> void myThreadTemplates::thread_1< ClassT >::join () [inline]

Join thread

3.4.3.4 template<class ClassT> static void myThreadTemplates::thread_1< ClassT >::self_test () [inline, static]

This method is only for test purpose!

3.4.3.5 template<class ClassT> void myThreadTemplates::thread_1< ClassT >::start () [inline]

Creates Thread and links it dynamic using static method

3.4.3.6 template<class ClassT> void myThreadTemplates::thread_1< ClassT >::start_self_test () [inline]

This method is only for test purpose!

3.4.3.7 template<class ClassT> void myThreadTemplates::thread_1< ClassT >::UpdateStats (string * *str*) [inline]

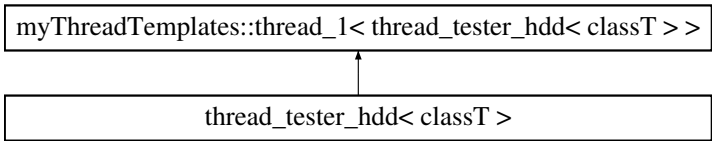
Push statistic information

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

- myThreadTemplates.hpp

3.5 thread_tester_hdd< classT > Class Template Reference

Inheritance diagram for thread_tester_hdd< classT >:



Public Member Functions

- **thread_tester_hdd** (vector< unsigned > &, boost::filesystem::path &, unsigned &, unsigned &, unsigned &, uint8_t &, mode_t &, bool, bool, classT *)
- **thread_tester_hdd** (classT *)
- void **setNewData** (vector< unsigned > &, boost::filesystem::path &)
- void **setBuffer** (const unsigned *)
- string & **getSummary** ()
- void Execute ()

template<class classT> class thread_tester_hdd< classT >

3.5.1 Member Function Documentation

3.5.1.1 template<class classT > void thread_tester_hdd< classT >::Execute () [inline]

Write test

Both read test

Cleaning after rw tests

Write test

Both read test

Cleaning after rw tests

Write test

Cleaning after rw tests

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

- thread_tester_hdd.hpp
- thread_tester_hdd.cpp

3.6 vector_str Struct Reference

Public Member Functions

- **vector_str** (unsigned &)

Public Attributes

- vector< string > **data**

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

- stats_keeper.hpp
- stats_keeper.cpp

Index