

# LittleBenchmark\_HDD

1.1.6

Generated by Doxygen 1.6.3

Tue Oct 12 00:13:53 2010



# Contents

<b>1</b>	<b>Module Index</b>	<b>1</b>
1.1	Modules . . . . .	1
<b>2</b>	<b>Class Index</b>	<b>3</b>
2.1	Class Hierarchy . . . . .	3
<b>3</b>	<b>Class Index</b>	<b>5</b>
3.1	Class List . . . . .	5
<b>4</b>	<b>Module Documentation</b>	<b>7</b>
4.1	Libbuskol . . . . .	7
4.1.1	Detailed Description . . . . .	8
4.1.2	Function Documentation . . . . .	8
4.1.2.1	appToFile . . . . .	8
4.1.2.2	Bandwidth . . . . .	8
4.1.2.3	CopyFileByChar . . . . .	8
4.1.2.4	createDir . . . . .	8
4.1.2.5	FromString . . . . .	8
4.1.2.6	GetLocalTime . . . . .	9
4.1.2.7	GetTime . . . . .	9
4.1.2.8	IsFiles . . . . .	9
4.1.2.9	ReadByChunk . . . . .	9
4.1.2.10	rm . . . . .	9
4.1.2.11	rmAll_inDir . . . . .	9
4.1.2.12	simpleReadToStringByChar . . . . .	9
4.1.2.13	simpleReadToStringByStream . . . . .	10
4.1.2.14	SimpleWriteToFile . . . . .	10
4.1.2.15	SizeFromString . . . . .	10
4.1.2.16	TimeDiff . . . . .	10

4.1.2.17	TimeDiff	10
4.1.2.18	TimeFromString	10
4.1.2.19	TimeToString	10
4.1.2.20	ToString	10
4.1.2.21	touch	11
<b>5</b>	<b>Class Documentation</b>	<b>13</b>
5.1	handler_Configuration Class Reference	13
5.1.1	Member Function Documentation	13
5.1.1.1	setUserDir	13
5.2	handler_Report Class Reference	15
5.3	profileNode Struct Reference	17
5.4	structRow Struct Reference	18
5.4.1	Constructor & Destructor Documentation	18
5.4.1.1	structRow	18
5.5	tester_hdd Class Reference	19
5.5.1	Constructor & Destructor Documentation	19
5.5.1.1	tester_hdd	19
5.5.2	Member Function Documentation	19
5.5.2.1	Run	19
5.6	buskol::ThreadTemplates::thread_1< ClassT > Class Template Reference	20
5.6.1	Detailed Description	20
5.6.2	Constructor & Destructor Documentation	20
5.6.2.1	~thread_1	20
5.6.3	Member Function Documentation	20
5.6.3.1	Execute_	20
5.6.3.2	join	20
5.6.3.3	join	21
5.6.3.4	self_test	21
5.6.3.5	start	21
5.6.3.6	start_self_test	21
5.6.3.7	UpdateStats	21
5.7	thread_tester_hdd< classT > Class Template Reference	22
5.7.1	Member Function Documentation	22
5.7.1.1	Execute	22

# Chapter 1

## Module Index

### 1.1 Modules

Here is a list of all modules:

Libbuskol . . . . . 7



# Chapter 2

## Class Index

### 2.1 Class Hierarchy

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

handler_Configuration . . . . .	13
tester_hdd . . . . .	19
handler_Report . . . . .	15
tester_hdd . . . . .	19
profileNode . . . . .	17
structRow . . . . .	18
buskol::ThreadTemplates::thread_1< ClassT > . . . . .	20
buskol::ThreadTemplates::thread_1< thread_tester_hdd< classT > > . . . . .	20
thread_tester_hdd< classT > . . . . .	22





## Chapter 3

# Class Index

### 3.1 Class List

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

<a href="#">handler_Configuration</a> . . . . .	13
<a href="#">handler_Report</a> . . . . .	15
<a href="#">profileNode</a> . . . . .	17
<a href="#">structRow</a> . . . . .	18
<a href="#">tester_hdd</a> . . . . .	19
<a href="#">buskol::ThreadTemplates::thread_1&lt; ClassT &gt;</a> (Thread_1 is very simple thread template which constainst few controle methods and entry for statistics ) . . . . .	20
<a href="#">thread_tester_hdd&lt; classT &gt;</a> . . . . .	22



# Chapter 4

## Module Documentation

### 4.1 Libbuskol

#### Classes

- class [buskol::ThreadTemplates::thread\\_1 < ClassT >](#)

*Thread\_1 is very simple thread template which constaints few controle methods and entry for statistics.*

#### Functions

- template<typename T >  
std::string [buskol::Conv::ToString](#) (const T &liczba)
- template<typename T >  
T [buskol::Conv::FromString](#) (const std::string &s\_liczba)
- template<typename T >  
std::string [buskol::Conv::Bandwidth](#) (double dTime, const T &ui64Counter, const std::string &strAppEnd="/s")
- template<typename T >  
T [buskol::Conv::SizeFromString](#) (std::string str)
- std::string [buskol::Conv::TimeToString](#) (const double &dTime)
- template<typename T >  
T [buskol::Conv::TimeFromString](#) (std::string str, const double &dMulti=1)
- bool [buskol::IO::simpleReadToStringByChar](#) (const boost::filesystem::path &strFile, std::string \*strBuf, std::list< double > \*p\_listStats=NULL)
- bool [buskol::IO::simpleReadToStringByStream](#) (const boost::filesystem::path &pathFile, std::string \*strBuf, std::list< double > \*p\_listStats=NULL, std::ios\_base::openmode mode=std::ios::in|std::ios::binary)
- bool [buskol::IO::touch](#) (const boost::filesystem::path &pathTo, const mode\_t &mode=0644)
- void [buskol::IO::createDir](#) (const boost::filesystem::path &Path, const mode\_t &mode=0644)
- bool [buskol::IO::SimpleWriteToFile](#) (const boost::filesystem::path &pathTo, const std::string &data, std::ios\_base::openmode mode=std::ios::app|std::ios::binary)
- bool [buskol::IO::appToFile](#) (const boost::filesystem::path &path, const std::string &data, const mode\_t &pmode=0644, const mode\_t &emode=0444)
- void [buskol::IO::rmAll\\_inDir](#) (const boost::filesystem::path &directory, bool bForced=true, const unsigned &intDirScanDepth=1024)

- void `buskol::IO::rm` (const boost::filesystem::path &path, const std::string &strOpt="", bool bforced=false)
- void `buskol::IO::lsFiles` (const boost::filesystem::path &bfsp\_dir, std::list< boost::filesystem::path > \*list\_dir)
- bool `buskol::IO::CopyFileByChar` (const boost::filesystem::path &pathFrom, const boost::filesystem::path &pathTo)
- bool `buskol::IO::ReadByChunk` (const boost::filesystem::path &path, const unsigned &chunk)
- boost::local\_time::local\_date\_time \* `buskol::Time::GetTime` (const std::string &tzzone="MST-07")
- double `buskol::Time::TimeDiff` (const boost::local\_time::local\_date\_time &Higher, const boost::local\_time::local\_date\_time &Lower)
- double `buskol::Time::TimeDiff` (const boost::local\_time::local\_date\_time &Lower)
- std::string `buskol::Time::GetLocalTime` (const std::string &strTimeFormatter "[%Y/%m/%d %H:%M:%S]")

### 4.1.1 Detailed Description

Additional documentation for group libbuskol

### 4.1.2 Function Documentation

**4.1.2.1** `bool buskol::IO::appToFile` (const boost::filesystem::path & *path*, const std::string & *data*, const mode\_t & *pmode* = 0644, const mode\_t & *emode* = 0444) [`inline`]

Function append data to file, support permission change.

**4.1.2.2** `template<typename T> std::string buskol::Conv::Bandwidth` (double *dTime*, const T & *ui64Counter*, const std::string & *strAppEnd* = "/s") [`inline`]

Count bandwidth or scale size. Output is human readable

**4.1.2.3** `bool buskol::IO::CopyFileByChar` (const boost::filesystem::path & *pathFrom*, const boost::filesystem::path & *pathTo*) [`inline`]

Function copy file by character

**4.1.2.4** `void buskol::IO::createDir` (const boost::filesystem::path & *Path*, const mode\_t & *mode* = 0644) [`inline`]

Function is equal to mkdir

**4.1.2.5** `template<typename T> T buskol::Conv::FromString` (const std::string & *s\_liczba*) [`inline`]

Convert string to number

**4.1.2.6** `std::string buskol::Time::GetLocalTime (const std::string & strTimeFormatter = "[%Y/%m/%d %H:%M:%S]") [inline]`

Function return time with defined formatting

< string buffer

< time structure for formatter

**4.1.2.7** `boost::local_time::local_date_time* buskol::Time::GetTime (const std::string & tzone = "MST-07") [inline]`

Function returns pointer to local\_data\_time struct Time and data format can be specified

**4.1.2.8** `void buskol::IO::lsFiles (const boost::filesystem::path & bfs_dir, std::list< boost::filesystem::path > * list_dir) [inline]`

Function list files in directory

**4.1.2.9** `bool buskol::IO::ReadByChunk (const boost::filesystem::path & path, const unsigned & chunk) [inline]`

Function read from file defined data size (Not implemented yet)

**4.1.2.10** `void buskol::IO::rm (const boost::filesystem::path & path, const std::string & strOpt = "", bool bforced = false) [inline]`

Function delete every thing in and give folder or empty file or folder depends on on string option Support forced remove by permission change (only for file or empty folder)

Fast remover for directory

Remove file or empty folder with premissions changing capabilities

**4.1.2.11** `void buskol::IO::rmAll_inDir (const boost::filesystem::path & directory, bool bForced = true, const unsigned & intDirScanDepth = 1024) [inline]`

Function delete all childerns in give directory by recurence Algoritm is fail proof by depth counter Support forced remove by permission change

**4.1.2.12** `bool buskol::IO::simpleReadToStringByChar (const boost::filesystem::path & strFile, std::string * strBuf, std::list< double > * p_listStats = NULL) [inline]`

Function read file by character and feel statistics if defined

< keep result of time subtraction

< Keep start time

< counter for temporary fix

Bug from unknow reason read 1 character more, maybe its special character fault, seekp(std::ios::end) return bad file legnth probably special character...

**4.1.2.13** `bool buskol::IO::simpleReadToStringByStream (const boost::filesystem::path & pathFile,  
std::string * strBuf, std::list< double > * p_listStats = NULL, std::ios_base::openmode  
mode = std::ios::in|std::ios::binary) [inline]`

Function read file by stream and feel statistics if defined

< Stream for read from file to string

< keep result of time subtraction

< Keep start time

**4.1.2.14** `bool buskol::IO::SimpleWriteToFile (const boost::filesystem::path &  
pathTo, const std::string & data, std::ios_base::openmode mode =  
std::ios::app|std::ios::binary) [inline]`

Function write data to file by stream

**4.1.2.15** `template<typename T > T buskol::Conv::SizeFromString (std::string str) [inline]`

Generate size from string

**4.1.2.16** `double buskol::Time::TimeDiff (const boost::local_time::local_date_time & Lower)  
[inline]`

Function return time difference for give local\_data\_time structer and curren time Time is being scaled to seconds!

Returns time difference

**4.1.2.17** `double buskol::Time::TimeDiff (const boost::local_time::local_date_time & Higher,  
const boost::local_time::local_date_time & Lower) [inline]`

Function return time difference for 2 give local\_data\_time structers Time is being scaled to seconds!

**4.1.2.18** `template<typename T > T buskol::Conv::TimeFromString (std::string str, const double  
& dMulti = 1) [inline]`

Convert time string to time as number, can also do scaling if multiplicand is defined

**4.1.2.19** `std::string buskol::Conv::TimeToString (const double & dTime) [inline]`

From time generates human readable string

**4.1.2.20** `template<typename T > std::string buskol::Conv::ToString (const T & liczba)  
[inline]`

Function convert number to string

---

**4.1.2.21** `bool buskol::IO::touch (const boost::filesystem::path & pathTo, const mode_t & mode = 0644) [inline]`

Function touch but can also set permission.



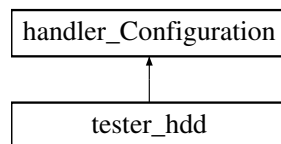


# Chapter 5

## Class Documentation

### 5.1 handler\_Configuration Class Reference

Inheritance diagram for handler\_Configuration:



#### Public Member Functions

- void [setUserDir](#) ()
- void **addNodeToStored** (const string &, const string &)
- void **clearNodes** ()
- void **parseConfigs** ()
- void **saveConfigs** ()

#### Protected Attributes

- boost::filesystem::path [pathProfile](#)  
*Keeps path to config file.*
- boost::filesystem::path [pathConfig](#)  
*Keeps path to config file.*
- bool **bLetUpdate**

#### 5.1.1 Member Function Documentation

##### 5.1.1.1 void handler\_Configuration::setUserDir ()

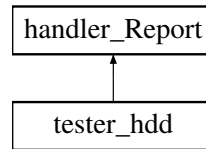
Creates program profile directory for current user

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

- handler\_Configuration.hpp
- handler\_Configuration.cpp

## 5.2 handler\_Report Class Reference

Inheritance diagram for handler\_Report:



### Public Member Functions

- [handler\\_Report](#) ()  
*Constructor set all args with default values.*
- void [addStatData](#) (string &, string &, const unsigned &, const unsigned &)  
*Will add data to specified buffers.*
- uint8\_t **findAndAdd** (string &, string &, unsigned &, unsigned &)
- 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< [structRow](#) > \* [p\\_vecstr\\_formattedTXT](#)  
*keep data in vector*
- boost::filesystem::path [pathReport](#)  
*Keeps path to report file.*
- string [strReportFile](#)  
*File to report will be written.*
- 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:

- handler\_Report.hpp
- handler\_Report.cpp

## 5.3 profileNode Struct Reference

### Public Member Functions

- **profileNode** (const string &, const string &)
- string **getData** ()

### Public Attributes

- string **strVar**
- string **strVal**

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

- handler\_Configuration.hpp
- handler\_Configuration.cpp

## 5.4 structRow Struct Reference

### Public Member Functions

- [structRow](#) (unsigned &)

### Public Attributes

- `vector< string > data`  
*Row.*

### 5.4.1 Constructor & Destructor Documentation

#### 5.4.1.1 structRow::structRow (unsigned & *col*)

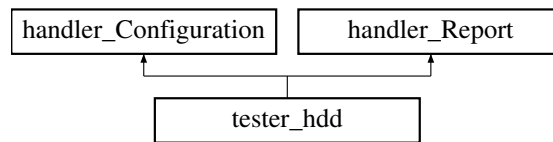
Creates defined number of columns

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

- handler\_Report.hpp
- handler\_Report.cpp

## 5.5 tester\_hdd Class Reference

Inheritance diagram for tester\_hdd:



### Public Member Functions

- [tester\\_hdd](#) (int, char \*\*)
- void [Run](#) ()

### Public Attributes

- bool **bRun**

### 5.5.1 Constructor & Destructor Documentation

#### 5.5.1.1 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

Number of columns

Set output args

### 5.5.2 Member Function Documentation

#### 5.5.2.1 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

## 5.6 buskol::ThreadTemplates::thread\_1< ClassT > Class Template Reference

Thread\_1 is very simple thread template which constaints 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)

#### 5.6.1 Detailed Description

```
template<class ClassT> class buskol::ThreadTemplates::thread_1< ClassT >
```

Thread\_1 is very simple thread template which constaints few controle methods and entry for statistics.

#### 5.6.2 Constructor & Destructor Documentation

**5.6.2.1** `template<class ClassT> virtual buskol::ThreadTemplates::thread_1< ClassT >::~~thread_1 () [inline, virtual]`

Virtual destructor which can show time of execution

#### 5.6.3 Member Function Documentation

**5.6.3.1** `template<class ClassT> static void buskol::ThreadTemplates::thread_1< ClassT >::Execute_ (ClassT *p) [inline, static]`

Static linker for dynamic method

**5.6.3.2** `template<class ClassT> void buskol::ThreadTemplates::thread_1< ClassT >::join (unsigned val) [inline]`

Join thread after specified time in seconds



**5.6.3.3** `template<class ClassT> void buskol::ThreadTemplates::thread_1< ClassT >::join ()`  
`[inline]`

Join thread

**5.6.3.4** `template<class ClassT> static void buskol::ThreadTemplates::thread_1< ClassT`  
`>::self_test () [inline, static]`

This method is only for test purpose!

**5.6.3.5** `template<class ClassT> void buskol::ThreadTemplates::thread_1< ClassT >::start ()`  
`[inline]`

Creates Thread and links it dynamic using static method

**5.6.3.6** `template<class ClassT> void buskol::ThreadTemplates::thread_1< ClassT`  
`>::start_self_test () [inline]`

This method is only for test purpose!

**5.6.3.7** `template<class ClassT> void buskol::ThreadTemplates::thread_1< ClassT`  
`>::UpdateStats (string * str) [inline]`

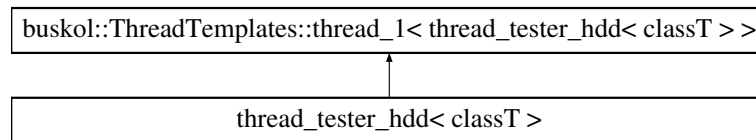
Push statistic information

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

- myThreadTemplates.hpp

## 5.7 thread\_tester\_hdd< classT > Class Template Reference

Inheritance diagram for thread\_tester\_hdd< classT >:



### Public Member Functions

- **thread\_tester\_hdd** (vector< uint64\_t > &, boost::filesystem::path &, unsigned &, unsigned &, unsigned &, unsigned &, uint16\_t &, mode\_t &, bool, bool, bool, classT \*)
- **thread\_tester\_hdd** (classT \*)
- void **setNewData** (vector< unsigned > &, boost::filesystem::path &)
- void [Execute](#) ()

```
template<class classT> class thread_tester_hdd< classT >
```

### 5.7.1 Member Function Documentation

#### 5.7.1.1 template<class classT > void thread\_tester\_hdd< classT >::Execute () [inline]

Write test

Both read test

Cleaning after rw tests

hybrird drive test which is normal test but with reversed order

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

# Index

- ~thread\_1
  - buskol::ThreadTemplates::thread\_1, [20](#)
- appToFile
  - libbuskol, [8](#)
- Bandwidth
  - libbuskol, [8](#)
- buskol::ThreadTemplates::thread\_1, [20](#)
  - ~thread\_1, [20](#)
  - Execute\_, [20](#)
  - join, [20](#)
  - self\_test, [21](#)
  - start, [21](#)
  - start\_self\_test, [21](#)
  - UpdateStats, [21](#)
- CopyFileByChar
  - libbuskol, [8](#)
- createDir
  - libbuskol, [8](#)
- Execute
  - thread\_tester\_hdd, [22](#)
- Execute\_
  - buskol::ThreadTemplates::thread\_1, [20](#)
- FromString
  - libbuskol, [8](#)
- GetLocalTime
  - libbuskol, [8](#)
- GetTime
  - libbuskol, [9](#)
- handler\_Configuration, [13](#)
  - setUserDir, [13](#)
- handler\_Report, [15](#)
- join
  - buskol::ThreadTemplates::thread\_1, [20](#)
- Libbuskol, [7](#)
- libbuskol
  - appToFile, [8](#)
  - Bandwidth, [8](#)
  - CopyFileByChar, [8](#)
  - createDir, [8](#)
  - FromString, [8](#)
  - GetLocalTime, [8](#)
  - GetTime, [9](#)
  - lsFiles, [9](#)
  - ReadByChunk, [9](#)
  - rm, [9](#)
  - rmAll\_inDir, [9](#)
  - simpleReadToStringByChar, [9](#)
  - simpleReadToStringByStream, [9](#)
  - SimpleWriteToFile, [10](#)
  - SizeFromString, [10](#)
  - TimeDiff, [10](#)
  - TimeFromString, [10](#)
  - TimeToString, [10](#)
  - ToString, [10](#)
  - touch, [10](#)
- lsFiles
  - libbuskol, [9](#)
- profileNode, [17](#)
- ReadByChunk
  - libbuskol, [9](#)
- rm
  - libbuskol, [9](#)
- rmAll\_inDir
  - libbuskol, [9](#)
- Run
  - tester\_hdd, [19](#)
- self\_test
  - buskol::ThreadTemplates::thread\_1, [21](#)
- setUserDir
  - handler\_Configuration, [13](#)
- simpleReadToStringByChar
  - libbuskol, [9](#)
- simpleReadToStringByStream
  - libbuskol, [9](#)
- SimpleWriteToFile
  - libbuskol, [10](#)
- SizeFromString
  - libbuskol, [10](#)
- start

- buskol::ThreadTemplates::thread\_1, [21](#)
- start\_self\_test
  - buskol::ThreadTemplates::thread\_1, [21](#)
- structRow, [18](#)
  - structRow, [18](#)
- tester\_hdd, [19](#)
  - Run, [19](#)
  - tester\_hdd, [19](#)
  - tester\_hdd, [19](#)
- thread\_tester\_hdd, [22](#)
  - Execute, [22](#)
- TimeDiff
  - libbuskol, [10](#)
- TimeFromString
  - libbuskol, [10](#)
- TimeToString
  - libbuskol, [10](#)
- ToString
  - libbuskol, [10](#)
- touch
  - libbuskol, [10](#)
- UpdateStats
  - buskol::ThreadTemplates::thread\_1, [21](#)