StreamAdapter

0.1

Generated by Doxygen 1.8.9.1

Fri Nov 27 2015 15:43:43

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 lax::StreamAdapter Class Reference

StreamAdapter - adapts incoming data into chunks of N bytes.

`#include <streamadapter.h>`

Public Member Functions

- StreamAdapter (std::size_t capacity=2 ∗1024 ∗1024)

  StreamAdapter constructor.
- **StreamAdapter** (const StreamAdapter &)=delete
- StreamAdapter & **operator=** (const StreamAdapter &)=delete
- void clear ()

  Removes all buffers from adapter.
- void push (unsigned char ∗buf, std::size_t size)

  Adds the data from buf to the data stored inside adapter and takes ownership of the buffer.
- void copy (unsigned char ∗dest, std::size_t offset, std::size_t size) const

  Copies size bytes of data starting at offset out of the buffers contained in current dapter into an array dest provided by the caller. The buffer is copied from [offset offset + size)
- void flush (std::size_t flush)

  Flushes the first flush bytes in the adapter.
- std::size_t available () const

  Gets the maximum amount of bytes available, that is it returns the maximum value that can be supplied.
- std::unique_ptr< unsigned char[ ]> take (std::size_t nbytes)

  Returns a freshly allocated buffer containing the first nbytes bytes of the adapter. The returned bytes will be flushed from the adapter.Caller owns returned value.
- std::size_t max_size () const

  get the capacity in bytes of this adapter

2.1.1 Detailed Description

StreamAdapter - adapts incoming data into chunks of N bytes.

Remarks

Note that this class is reentrant, but not thread-safe!

2.1.2 Constructor & Destructor Documentation

2.1.2.1 lax::StreamAdapter::StreamAdapter (std::size_t *capacity* = 2*1024*1024) [explicit]

StreamAdapter constructor.

Parameters

capacity	the max number of bytes this adapter can hold

2.1.3 Member Function Documentation

2.1.3.1 std::size_t lax::StreamAdapter::available ( ) const

Gets the maximum amount of bytes available, that is it returns the maximum value that can be supplied.

Returns

number of bytes available in adapter

2.1.3.2 void lax::StreamAdapter::copy (unsigned char $*$ *dest,* std::size_t *offset,* std::size_t *size*) const

Copies *size* bytes of data starting at *offset* out of the buffers contained in current dapter into an array *dest* provided by the caller. The buffer is copied from [*offset offset* + *size*)

The array *dest* should be large enough to contain *size* bytes. The user should check that the adapter has (*offset* + *size*) bytes available before calling this function.

Parameters

dest	the memory to copy into
offset	the bytes offset in the adapter to start from
size	the number of bytes to copy

Exceptions

std::length_error	if *offset* > `available()` or (*offset* + *size*) > `available()`

2.1.3.3 void lax::StreamAdapter::flush (std::size_t *flush*)

Flushes the first *flush* bytes in the adapter.

The caller must ensure that at least this *flush* bytes are available.

Parameters

flush	the number of bytes to flush

2.1.3.4 std::size_t lax::StreamAdapter::max_size ( ) const

get the capacity in bytes of this adapter

Returns

the capacity of this adapter in bytes

2.1.3.5 void lax::StreamAdapter::push (unsigned char $*$ *buf,* std::size_t *size*)

Adds the data from *buf* to the data stored inside adapter and takes ownership of the buffer.

Parameters

buf	a buffer to add to queue in the adapter
size	the number of bytes to copy from *buf*

Exceptions

std::length_error	if `available()` + *size* $>$ `max_size()`

2.1.3.6 std::unique_ptr< unsigned char[ ]> lax::StreamAdapter::take (std::size_t *nbytes*)

Returns a freshly allocated buffer containing the first *nbytes* bytes of the adapter. The returned bytes will be flushed from the adapter.Caller owns returned value.

Parameters

nbytes	the number of bytes to take

Returns

oven-fresh hot data, or `nullptr` if *nbytes* bytes are not available.

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

- streamadapter.h
- streamadapter.cpp

Index