# LFC1 Libraries

0.0.1

Generated by Doxygen 1.8.3.1

Tue May 7 2013 07:38:04

# Contents

# Chapter 1

# Main Page

This library is an extension of the C++ standard library and the Boost C++ library. It enhances C++ code reliability by providing the following capabilities:

- A set of error codes and an error category for this library.
- A set of templates which provides exception/error handling for inserters, extractors and manipulators of user-defined types. These templates handle exceptions derived from std::bad_alloc, std::exception and unknown exceptions.
- A set of miscellaneous classes to support the compiler, e.g. name demangling.
- A set of type definitions and templates which represent numbers stored in various ways.
- A set of classes which extends the Boost filesystem library.
- A set of date and time classes which provide millisecond precision.
- A set of classes for checksum calculation.
- A set of classes for logging.
- A set of classes for code conversion.
- A set of classes representing ISO standards.
- A set of classes which provides the ability to read and write ID3 v1.0 tags.
- A set of classes which provides the ability to read and write ID3 v1.1 tags.
- A set of classes common to all ID3 v2.x tags.
- A set of classes which provides the ability to read and write ID3 v2.2 tags.
- A set of classes which provides the ability to read and write ID3 v2.3 tags.
- A set of classes which are wrappers to the C ODBC API.
- This library contains a set of classes representing ISO standards whose data is obtained from a database.

**Note**

> String data handled by this library uses the UTF-8 character set. This library is thread-safe. The code in this library complies to the recommendations contained in the books C++ Coding Standards and Effective C++ and the document LFC-CS-0003 - C++ Coding Standards.doc.

# Chapter 2

# Module Index

## 2.1 Modules

Here is a list of all modules:

# Chapter 3

# Class Index

## 3.1 Class List

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

lfc1::udtsup::CManipBase< T1 >
This class template provides exception/error handling for single argument manipulators of user-defined types . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 13

# Chapter 4

# Module Documentation

## 4.1 User-defined type support class library

### Classes

- class lfc1::udtsup::CManipBase< T1 >

  *This class template provides exception/error handling for single argument manipulators of user-defined types.*

### Functions

- template<typename T2 >
  void lfc1::udtsup::gvDoManip (std::basic_ios< char > &arIos, const CManipBase< T2 > &arManipBase)

  *This template function calls the user-defined manipulator function and handles exceptions from it.*
- template<typename T3 >
  std::istream & lfc1::udtsup::operator>> (std::istream &arStream, const CManipBase< T3 > &arManipBase)

  *This template function invokes a user-defined manipulator for an input stream.*
- template<typename T3 >
  std::ostream & lfc1::udtsup::operator<< (std::ostream &arStream, const CManipBase< T3 > &arManipBase)

  *This template function invokes a user-defined manipulator for an output stream.*
- template<typename UDT >
  std::istream & lfc1::udtsup::gvInput (std::istream &arStream, UDT &arUdt)

  *This template function is meant to be the only function called by the extractor of a user-defined type. This function provides exception/error handling for the extractor of a user-defined type.*
- template<typename UDT >
  std::istream & lfc1::udtsup::gvInputGuarded (std::istream &arStream, UDT &arUdt)

  *This template function is meant to be the only function called by the extractor of a user-defined type. This function provides exception/error handling for the extractor of a user-defined type.*
- template<typename UDT >
  std::ostream & lfc1::udtsup::gvOutput (std::ostream &arStream, const UDT &arUdt)

  *This template function is meant to be the only function called by the inserter of a user-defined type. This function provides exception/error handling for the extractor of a user-defined type.*
- template<typename UDT >
  std::ostream & lfc1::udtsup::gvOutputGuarded (std::ostream &arStream, const UDT &arUdt)

  *This template function is meant to be the only function called by the inserter of a user-defined type. This function provides exception/error handling for the extractor of a user-defined type.*

### 4.1.1 Detailed Description

This library contains a set of templates which provides exception/error handling for inserters, extractors and manipulators of user-defined types. These templates handle exceptions derived from std::bad_alloc, std::exception and unknown exceptions.

### 4.1.2 Function Documentation

#### 4.1.2.1 template< typename T2 > void lfc1::udtsup::gvDoManip ( std::basic_ios< char > & *arIos,* const CManipBase< T2 > & *arManipBase* )

This template function calls the user-defined manipulator function and handles exceptions from it.

**Template Parameters**

| | |
|---|---|
| *T2* | The user-defined manipulator function argument type. |

**Parameters**

| | |
|---|---|
| *arIos* | The stream's state object. |
| *arManipBase* | The user-defined manipulator function container. |

#### 4.1.2.2 template< typename UDT > std::istream& lfc1::udtsup::gvInput ( std::istream & *arStream,* UDT & *arUdt* )

This template function is meant to be the only function called by the extractor of a user-defined type. This function provides exception/error handling for the extractor of a user-defined type.

This template was derived from Section 3.1.5 of the book Standard C++ IOStreams and Locales. The exception handling sets the stream state and allows the handled exception to propagate if the stream allows it.

The user-defined type is expected to contain a function with the following signature:

```
std::ios_base::iostate mvInput(std::istream&)
```

The mvInput() function can call extractors for built-in types thus this function does not instantiate a stream sentry object.

**Template Parameters**

| | |
|---|---|
| *UDT* | The user-defined type. |

**Parameters**

| | | |
|---|---|---|
| `in` | *arStream* | The source stream. |
| `out` | *arUdt* | The user-defined type. |

**Returns**

The source stream.

#### 4.1.2.3 template< typename UDT > std::istream& lfc1::udtsup::gvInputGuarded ( std::istream & *arStream,* UDT & *arUdt* )

This template function is meant to be the only function called by the extractor of a user-defined type. This function provides exception/error handling for the extractor of a user-defined type.

This template was derived from Section 3.1.5 of the book Standard C++ IOStreams and Locales. The exception handling sets the stream state and allows the handled exception to propagate if the stream allows it.

The user-defined type is expected to contain a function with the following signature:

```
std::ios_base::iostate mvInput(std::istream&)
```

The mvInput() function must not call extractors for built-in types because this function instantiates a stream sentry object.

**Template Parameters**

| | |
|---|---|
| *UDT* | The user-defined type. |

**Parameters**

| | | |
|---|---|---|
| `in` | *arStream* | The source stream. |
| `out` | *arUdt* | The user-defined type. |

**Returns**

The source stream.

#### 4.1.2.4 template< typename UDT > std::ostream& lfc1::udtsup::gvOutput ( std::ostream & *arStream*, const UDT & *arUdt* )

This template function is meant to be the only function called by the inserter of a user-defined type. This function provides exception/error handling for the extractor of a user-defined type.

This template was derived from Section 3.1.5 of the book Standard C++ IOStreams and Locales. The exception handling sets the stream state and allows the handled exception to propagate if the stream allows it.

The user-defined type is expected to contain a function with the following signature:

```
std::ios_base::iostate mvOutput(std::istream&)
```

The mvOutput() function can call inserters for built-in types thus this function does not instantiate a stream sentry object.

**Template Parameters**

| | |
|---|---|
| *UDT* | The user-defined type. |

**Parameters**

| | | |
|---|---|---|
| `in` | *arStream* | The destination stream. |
| `out` | *arUdt* | The user-defined type. |

**Returns**

The destination stream.

#### 4.1.2.5 template< typename UDT > std::ostream& lfc1::udtsup::gvOutputGuarded ( std::ostream & *arStream*, const UDT & *arUdt* )

This template function is meant to be the only function called by the inserter of a user-defined type. This function provides exception/error handling for the extractor of a user-defined type.

This template was derived from Section 3.1.5 of the book Standard C++ IOStreams and Locales. The exception handling sets the stream state and allows the handled exception to propagate if the stream allows it.

The user-defined type is expected to contain a function with the following signature:

```
std::ios_base::iostate mvOutput(std::istream&)
```

The mvOutput() function must not call inserters for built-in types because this function instantiates a stream sentry object.

**Template Parameters**

| | |
|---|---|
| *UDT* | The user-defined type. |

**Parameters**

| | | |
|---|---|---|
| `in` | *arStream* | The destination stream. |
| `out` | *arUdt* | The user-defined type. |

**Returns**

The destination stream.

### 4.1.2.6 template< typename T3 > std::ostream& lfc1::udtsup::operator<< ( std::ostream & *arStream*, const CManipBase< T3 > & *arManipBase* )

This template function invokes a user-defined manipulator for an output stream.

**Template Parameters**

| | |
|---|---|
| *T3* | The user-defined manipulator function argument type. |

**Parameters**

| | | |
|---|---|---|
| `in` | *arStream* | The destination stream. |
| `in` | *arManipBase* | The CManipBase object. |

**Returns**

The destination stream.

### 4.1.2.7 template< typename T3 > std::istream& lfc1::udtsup::operator>> ( std::istream & *arStream*, const CManipBase< T3 > & *arManipBase* )

This template function invokes a user-defined manipulator for an input stream.

**Template Parameters**

| | |
|---|---|
| *T3* | The user-defined manipulator function argument type. |

**Parameters**

| | | |
|---|---|---|
| `in` | *arStream* | The source stream. |
| `out` | *arManipBase* | The CManipBase object. |

**Returns**

The source stream.

# Chapter 5

# Class Documentation

## 5.1 lfc1::udtsup::CManipBase< T1 > Class Template Reference

This class template provides exception/error handling for single argument manipulators of user-defined types.

`#include <lfc1/udtsup/cmanipbase.hpp>`

### Public Types

- typedef void(* TManipFunc )(std::ios_base &, T1)

  *Single argument manipulator signature.*

### Public Member Functions

- CManipBase (TManipFunc apManipFunc, T1 &arArg)

  *This template function creates a default CManipBase object.*
- CManipBase (const CManipBase &)=default

  *Uses default implementation.*
- virtual ~CManipBase () noexcept

  *This template function destroys a CManipBase object.*
- CManipBase & operator= (const CManipBase &)=default

  *Uses default implementation.*

### Friends

- template< typename T2 >
  void gvDoManip (std::basic_ios< char > &arIos, const CManipBase< T2 > &arManipBase)

  *This template function calls the user-defined manipulator function and handles exceptions from it.*

### 5.1.1 Detailed Description

**template< typename T1 > class lfc1::udtsup::CManipBase< T1 >**

This class template provides exception/error handling for single argument manipulators of user-defined types.

This template was derived from Section 3.2.2.4.1 of the book Standard C++ IOStreams and Locales. The exception handling sets the stream state and allows the handled exception to propagate if the stream allows it.

**Template Parameters**

| | |
|---|---|
| *T1* | Manipulator function argument type. |

## 5.1.2 Constructor & Destructor Documentation

### 5.1.2.1 template<typename T1 > lfc1::udtsup::CManipBase< T1 >::CManipBase ( TManipFunc *apManipFunc*, T1 & *arArg* )

This template function creates a default CManipBase object.

**Template Parameters**

| | |
|---|---|
| *T1* | Manipulator function argument type. |

**Parameters**

| | | |
|---|---|---|
| `in` | *apManipFunc* | The user-defined manipulator function. |
| `in` | *arArg* | The user-defined manipulator function argument. |

### 5.1.2.2 template<typename T1 > lfc1::udtsup::CManipBase< T1 >::∼CManipBase ( ) `[virtual]`, `[noexcept]`

This template function destroys a CManipBase object.

**Template Parameters**

| | |
|---|---|
| *T1* | Manipulator function argument type. |

## 5.1.3 Friends And Related Function Documentation

### 5.1.3.1 template<typename T1> template<typename T2 > void gvDoManip ( std::basic_ios< char > & *arIos*, const CManipBase< T2 > & *arManipBase* ) `[friend]`

This template function calls the user-defined manipulator function and handles exceptions from it.

**Template Parameters**

| | |
|---|---|
| *T2* | The user-defined manipulator function argument type. |

**Parameters**

| | |
|---|---|
| *arIos* | The stream's state object. |
| *arManipBase* | The user-defined manipulator function container. |

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

- include/lfc1/udtsup/cmanipbase.hpp

# Index