

LFC1 Libraries

0.0.1

Generated by Doxygen 1.8.3.1

Tue May 7 2013 07:01:24

Contents

1	Main Page	1
2	Module Index	3
2.1	Modules	3
3	Hierarchical Index	5
3.1	Class Hierarchy	5
4	Class Index	7
4.1	Class List	7
5	Module Documentation	9
5.1	System class library	9
5.1.1	Detailed Description	9
5.1.2	Function Documentation	10
5.1.2.1	grCategory	10
5.2	Filesystem class library	11
5.2.1	Detailed Description	11
5.2.2	Function Documentation	11
5.2.2.1	swap	11
5.2.2.2	swap	11
6	Class Documentation	13
6.1	lfc1::filesystem::CDirectory Class Reference	13
6.1.1	Detailed Description	14
6.1.2	Constructor & Destructor Documentation	14
6.1.2.1	CDirectory	14
6.1.2.2	CDirectory	14
6.1.2.3	CDirectory	14
6.1.3	Member Function Documentation	14
6.1.3.1	begin	14
6.1.3.2	end	15
6.1.3.3	mvGetNumEntries	15

6.1.3.4	mvGetPath	15
6.1.3.5	mvRefresh	15
6.1.3.6	operator=	15
6.1.3.7	operator=	15
6.1.3.8	swap	16
6.2	lfc1::system::CErrorCategory Class Reference	16
6.2.1	Detailed Description	17
6.2.2	Member Function Documentation	17
6.2.2.1	message	17
6.2.2.2	name	17
6.2.2.3	smrGetErrorCategory	17
6.3	lfc1::filesystem::CRecDirectory Class Reference	18
6.3.1	Detailed Description	18
6.3.2	Constructor & Destructor Documentation	19
6.3.2.1	CRecDirectory	19
6.3.2.2	CRecDirectory	19
6.3.2.3	CRecDirectory	19
6.3.3	Member Function Documentation	19
6.3.3.1	begin	19
6.3.3.2	end	19
6.3.3.3	mvGetLevelCount	20
6.3.3.4	mvGetNumEntries	20
6.3.3.5	mvGetPath	20
6.3.3.6	mvRefresh	20
6.3.3.7	operator=	20
6.3.3.8	operator=	20
6.3.3.9	swap	21

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:

System class library	9
Filesystem class library	11

Chapter 3

Hierarchical Index

3.1 Class Hierarchy

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

lfc1::filesystem::CDirectory	13
lfc1::filesystem::CRecDirectory	18
std::error_category	
lfc1::system::CErrorCategory	16

Chapter 4

Class Index

4.1 Class List

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

lfc1::filesystem::CDirectory	
This class wraps a directory iterator so that directory entries could be loaded and sorted	13
lfc1::system::CErrorCategory	
This class is an error category for this library	16
lfc1::filesystem::CRecDirectory	
This class wraps a recursive directory iterator so that directory entries could be loaded and sorted	18

Chapter 5

Module Documentation

5.1 System class library

Classes

- class `lfc1::system::CErrorCategory`

This class is an error category for this library.

Enumerations

- enum `lfc1::system::ErrC` {
 E_EC_SUCCESS,
 E_EC_DIR_OPEN_ERROR,
 E_EC_DIR_READ_ERROR,
 E_EC_DUP_NAMEVALUE,
 E_EC_MISSING_NAMEVALUE,
 E_EC_INVALID_NAMEVALUE,
 E_EC_MISSING_LANGUAGE_PART,
 E_EC_MISSING_CURRENCY_PART,
 E_EC_MISSING_COUNTRY_PART,
 E_EC_READ_PAST_EOF,
 E_EC_INVALID_TEXT_ENCODING,
 E_EC_MISSING_UTF16_BOM,
 E_EC_NO_FRAMES,
 E_EC_READ_PAST_EOT,
 E_EC_CHECKSUM_MISMATCH }

This enumeration defines the error codes for this library.

Functions

- `std::error_category & lfc1::system::grCategory () noexcept`

This function obtains a reference to the single error category object.

5.1.1 Detailed Description

This library contains a set of error codes and an error category for this library.

5.1.2 Function Documentation

5.1.2.1 `std::error_category & lfc1::system::grCategory ()` `[noexcept]`

This function obtains a reference to the single error category object.

Returns

A reference to the single error category object.

5.2 Filesystem class library

Classes

- class `lfc1::filesystem::CDirectory`
This class wraps a directory iterator so that directory entries could be loaded and sorted.
- class `lfc1::filesystem::CRecDirectory`
This class wraps a recursive directory iterator so that directory entries could be loaded and sorted.

Functions

- void `lfc1::filesystem::swap` (CDirectory &arLHS, CDirectory &arRHS) noexcept
This function overloads `std::swap` for `CDirectory` objects.
- void `lfc1::filesystem::swap` (CRecDirectory &arLHS, CRecDirectory &arRHS) noexcept
This function overloads `std::swap` for `CRecDirectory` objects.

5.2.1 Detailed Description

This library contains a set of classes which extends the standard/Boost filesystem library.

5.2.2 Function Documentation

5.2.2.1 void `lfc1::filesystem::swap` (CDirectory & *arLHS*, CDirectory & *arRHS*) [noexcept]

This function overloads `std::swap` for `CDirectory` objects.

Parameters

<code>in, out</code>	<i>arLHS</i>	The first object to swap.
<code>in, out</code>	<i>arRHS</i>	The second object to swap.

5.2.2.2 void `lfc1::filesystem::swap` (CRecDirectory & *arLHS*, CRecDirectory & *arRHS*) [noexcept]

This function overloads `std::swap` for `CRecDirectory` objects.

Parameters

<code>in, out</code>	<i>arLHS</i>	The first object to swap.
<code>in, out</code>	<i>arRHS</i>	The second object to swap.

Chapter 6

Class Documentation

6.1 lfc1::filesystem::CDirectory Class Reference

This class wraps a directory iterator so that directory entries could be loaded and sorted.

```
#include <lfc1/filesystem/cdirectory.hpp>
```

Public Types

- typedef
boost::filesystem::directory_entry TEntry
The type of a directory entry.
- typedef std::vector< TEntry > TEntries
The type of the directory entries container.
- typedef TEntries::const_iterator TIterator
The type of the directory entries container iterator.

Public Member Functions

- CDirectory (const boost::filesystem::path &arPath)
This function creates a CDirectory object.
- CDirectory (const CDirectory &arRHS)
This function copy constructs a CDirectory object.
- CDirectory (CDirectory &&arRHS)
This function move constructs a CDirectory object.
- ~CDirectory () noexcept
This function destroys a CDirectory object.
- CDirectory & operator= (const CDirectory &arRHS)
This function assigns a CDirectory object to another CDirectory object.
- CDirectory & operator= (CDirectory &&arRHS) noexcept
This function moves a CDirectory object to another CDirectory object.
- boost::filesystem::path mvGetPath () const
This function obtains the name of the directory.
- unsigned mvGetNumEntries () const noexcept
This function obtains the number of directory entries.
- TIterator begin () const noexcept
This function obtains the beginning iterator to the list of directory entries.
- TIterator end () const noexcept

This function obtains the ending iterator to the list of directory entries.

- void [swap](#) ([CDirectory](#) &arRHS) noexcept

This function swaps a [CDirectory](#) object with another [CDirectory](#) object.

- void [mvRefresh](#) ()

This function loads and sorts the contents of a directory.

6.1.1 Detailed Description

This class wraps a directory iterator so that directory entries could be loaded and sorted.

6.1.2 Constructor & Destructor Documentation

6.1.2.1 lfc1::filesystem::CDirectory::CDirectory (const boost::filesystem::path & arPath)

This function creates a [CDirectory](#) object.

Parameters

in	<i>arPath</i>	The name of the directory.
----	---------------	----------------------------

6.1.2.2 lfc1::filesystem::CDirectory::CDirectory (const CDirectory & arRHS)

This function copy constructs a [CDirectory](#) object.

Parameters

in	<i>arRHS</i>	The CDirectory object to be copied.
----	--------------	---

6.1.2.3 lfc1::filesystem::CDirectory::CDirectory (CDirectory && arRHS)

This function move constructs a [CDirectory](#) object.

Parameters

in	<i>arRHS</i>	The CDirectory object to be moved.
----	--------------	--

6.1.3 Member Function Documentation

6.1.3.1 CDirectory::TIterator lfc1::filesystem::CDirectory::begin () const [noexcept]

This function obtains the beginning iterator to the list of directory entries.

Returns

The beginning iterator to the list of directory entries.

Note

This function does not follow this library's member function naming convention in order to support range-base for loops.

6.1.3.2 CDirectory::TIterator lfc1::filesystem::CDirectory::end () const [noexcept]

This function obtains the ending iterator to the list of directory entries.

Returns

The ending iterator to the list of directory entries.

Note

This function does not follow this library's member function naming convention in order to support range-base for loops.

6.1.3.3 unsigned lfc1::filesystem::CDirectory::mvGetNumEntries () const [noexcept]

This function obtains the number of directory entries.

Returns

The number of directory entries.

6.1.3.4 boost::filesystem::path lfc1::filesystem::CDirectory::mvGetPath () const

This function obtains the name of the directory.

Returns

The name of the directory.

6.1.3.5 void lfc1::filesystem::CDirectory::mvRefresh ()

This function loads and sorts the contents of a directory.

Exceptions

<i>std::system_error</i>	Indicates a directory open or read failure.
--------------------------	---

6.1.3.6 CDirectory & lfc1::filesystem::CDirectory::operator= (const CDirectory & arRHS)

This function assigns a [CDirectory](#) object to another [CDirectory](#) object.

Parameters

<i>in</i>	<i>arRHS</i>	The CDirectory object to assign to another CDirectory object.
-----------	--------------	---

Returns

The [CDirectory](#) object assigned to.

6.1.3.7 CDirectory & lfc1::filesystem::CDirectory::operator= (CDirectory && arRHS) [noexcept]

This function moves a [CDirectory](#) object to another [CDirectory](#) object.

Parameters

in	arRHS	The CDirectory object to move to another CDirectory object.
----	-------	---

Returns

The [CDirectory](#) object assigned to.

6.1.3.8 void lfc1::filesystem::CDirectory::swap (CDirectory & arRHS) [noexcept]

This function swaps a [CDirectory](#) object with another [CDirectory](#) object.

Parameters

in, out	arRHS	The CDirectory object to swap with.
---------	-------	---

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

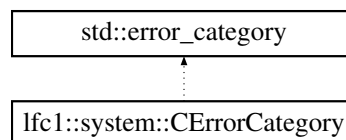
- include/lfc1/filesystem/cdirectory.hpp
- filesystem/library/src/cdirectory.cpp

6.2 lfc1::system::CErrorCategory Class Reference

This class is an error category for this library.

```
#include <lfc1/system/ceerrorcategory.hpp>
```

Inheritance diagram for lfc1::system::CErrorCategory:



Public Member Functions

- [CErrorCategory](#) (const [CErrorCategory](#) &)=delete
Not supported.
- [CErrorCategory](#) ([CErrorCategory](#) &&)=delete
Not supported.
- virtual [~CErrorCategory](#) () noexcept
This function destroys a [CErrorCategory](#) object.
- [CErrorCategory](#) & operator= (const [CErrorCategory](#) &)=delete
Not supported.
- [CErrorCategory](#) & operator= ([CErrorCategory](#) &&)=delete
Not supported.
- virtual const char * [name](#) () const noexcept
This function obtains the name of the error category.
- virtual std::string [message](#) (int ev) const
This function obtains the message equivalent to the given error code.

Static Public Member Functions

- static `std::error_category & smrGetErrorCategory ()` `noexcept`
This function obtains a reference to a single error category object.

Protected Member Functions

- `CErrorCategory ()` `noexcept`
This function creates a default `CErrorCategory` object.

6.2.1 Detailed Description

This class is an error category for this library.

Note

This class uses the Singleton design pattern thus only one instance of this class exists within the same process.

6.2.2 Member Function Documentation

6.2.2.1 `std::string lfc1::system::CErrorCategory::message (int ev) const` `[virtual]`

This function obtains the message equivalent to the given error code.

Parameters

<code>in</code>	<code>ev</code>	The error code.
-----------------	-----------------	-----------------

Returns

The message equivalent to the given error code.

6.2.2.2 `const char * lfc1::system::CErrorCategory::name () const` `[virtual]`, `[noexcept]`

This function obtains the name of the error category.

Returns

The name of the error category.

6.2.2.3 `std::error_category & lfc1::system::CErrorCategory::smrGetErrorCategory ()` `[static]`, `[noexcept]`

This function obtains a reference to a single error category object.

Returns

A reference to a single error category object.

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

- `include/lfc1/system/cerrorcategory.hpp`
- `system/library/src/cerrorcategory.cpp`

6.3 lfc1::filesystem::CRecDirectory Class Reference

This class wraps a recursive directory iterator so that directory entries could be loaded and sorted.

```
#include <lfc1/filesystem/crecdirectory.hpp>
```

Public Types

- typedef std::pair
< boost::filesystem::directory_entry,
int > TEntry
The type of a directory entry.
- typedef std::vector< TEntry > TEntries
The type of the directory entries container.
- typedef TEntries::const_iterator TIterator
The type of the directory entries container iterator.

Public Member Functions

- CRecDirectory (const boost::filesystem::path &arPath)
This function creates a default CRecDirectory object.
- CRecDirectory (const CRecDirectory &arRHS)
This function copy constructs a CRecDirectory object.
- CRecDirectory (CRecDirectory &&arRHS)
This function move constructs a CRecDirectory object.
- ~CRecDirectory () noexcept
This function destroys a CRecDirectory object.
- CRecDirectory & operator= (const CRecDirectory &arRHS)
This function assigns a CRecDirectory object to another CRecDirectory object.
- CRecDirectory & operator= (CRecDirectory &&arRHS) noexcept
This function moves a CRecDirectory object to another CRecDirectory object.
- int mvGetLevelCount () const noexcept
This function obtains the number of directory levels.
- boost::filesystem::path mvGetPath () const
This function obtains the name of the directory.
- unsigned mvGetNumEntries () const noexcept
This function obtains the number of directory entries.
- TIterator begin () const noexcept
This function obtains the beginning iterator to the list of directory entries.
- TIterator end () const noexcept
This function obtains the ending iterator to the list of directory entries.
- void swap (CRecDirectory &arRHS) noexcept
This function swaps a CRecDirectory object with another CRecDirectory object.
- void mvRefresh ()
This function loads and sorts the contents of a directory.

6.3.1 Detailed Description

This class wraps a recursive directory iterator so that directory entries could be loaded and sorted.

6.3.2 Constructor & Destructor Documentation

6.3.2.1 lfc1::filesystem::CRecDirectory::CRecDirectory (const boost::filesystem::path & *arPath*)

This function creates a default [CRecDirectory](#) object.

Parameters

<i>in</i>	<i>arPath</i>	The name of the directory.
-----------	---------------	----------------------------

6.3.2.2 lfc1::filesystem::CRecDirectory::CRecDirectory (const CRecDirectory & *arRHS*)

This function copy constructs a [CRecDirectory](#) object.

Parameters

<i>in</i>	<i>arRHS</i>	The CRecDirectory object to be copied.
-----------	--------------	--

6.3.2.3 lfc1::filesystem::CRecDirectory::CRecDirectory (CRecDirectory && *arRHS*)

This function move constructs a [CRecDirectory](#) object.

Parameters

<i>in</i>	<i>arRHS</i>	The CRecDirectory object to be moved.
-----------	--------------	---

6.3.3 Member Function Documentation

6.3.3.1 CRecDirectory::TIterator lfc1::filesystem::CRecDirectory::begin () const [noexcept]

This function obtains the beginning iterator to the list of directory entries.

Returns

The beginning iterator to the list of directory entries.

Note

This function does not follow this library's member function naming convention in order to support range-base for loops.

6.3.3.2 CRecDirectory::TIterator lfc1::filesystem::CRecDirectory::end () const [noexcept]

This function obtains the ending iterator to the list of directory entries.

Returns

The ending iterator to the list of directory entries.

Note

This function does not follow this library's member function naming convention in order to support range-base for loops.

6.3.3.3 `int lfc1::filesystem::CRecDirectory::mvGetLevelCount () const` `[noexcept]`

This function obtains the number of directory levels.

Returns

The number of directory levels.

6.3.3.4 `unsigned lfc1::filesystem::CRecDirectory::mvGetNumEntries () const` `[noexcept]`

This function obtains the number of directory entries.

Returns

The number of directory entries.

6.3.3.5 `boost::filesystem::path lfc1::filesystem::CRecDirectory::mvGetPath () const`

This function obtains the name of the directory.

Returns

The name of the directory.

6.3.3.6 `void lfc1::filesystem::CRecDirectory::mvRefresh ()`

This function loads and sorts the contents of a directory.

Exceptions

<code>std::system_error</code>	Indicates a directory open or read failure.
--------------------------------	---

6.3.3.7 `CRecDirectory & lfc1::filesystem::CRecDirectory::operator= (const CRecDirectory & arRHS)`

This function assigns a [CRecDirectory](#) object to another [CRecDirectory](#) object.

Parameters

<code>in</code>	<code>arRHS</code>	The CRecDirectory object to assign to another CRecDirectory object.
-----------------	--------------------	---

Returns

The [CRecDirectory](#) object assigned to.

6.3.3.8 `CRecDirectory & lfc1::filesystem::CRecDirectory::operator= (CRecDirectory && arRHS)` `[noexcept]`

This function moves a [CRecDirectory](#) object to another [CRecDirectory](#) object.

Parameters

<code>in</code>	<code>arRHS</code>	The CRecDirectory object to move to another CRecDirectory object.
-----------------	--------------------	---

Returns

The [CRecDirectory](#) object assigned to.

6.3.3.9 void lfc1::filesystem::CRecDirectory::swap (CRecDirectory & *arRHS*) [noexcept]

This function swaps a [CRecDirectory](#) object with another [CRecDirectory](#) object.

Parameters

<i>in, out</i>	<i>arRHS</i>	The CRecDirectory object to swap with.
----------------	--------------	--

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

- include/lfc1/filesystem/crecdirectory.hpp
- filesystem/library/src/crecdirectory.cpp

Index

begin
 Ifc1::filesystem::CDirectory, 14
 Ifc1::filesystem::CRecDirectory, 19

CDirectory
 Ifc1::filesystem::CDirectory, 14

CRecDirectory
 Ifc1::filesystem::CRecDirectory, 19

end
 Ifc1::filesystem::CDirectory, 14
 Ifc1::filesystem::CRecDirectory, 19

Filesystem class library, 11
 swap, 11

grCategory
 System class library, 10

Ifc1::filesystem::CDirectory, 13
 begin, 14
 CDirectory, 14
 end, 14
 mvGetNumEntries, 15
 mvGetPath, 15
 mvRefresh, 15
 operator=, 15
 swap, 16

Ifc1::filesystem::CRecDirectory, 18
 begin, 19
 CRecDirectory, 19
 end, 19
 mvGetLevelCount, 19
 mvGetNumEntries, 20
 mvGetPath, 20
 mvRefresh, 20
 operator=, 20
 swap, 21

Ifc1::system::CErrorCategory, 16
 message, 17
 name, 17
 smrGetErrorCategory, 17

message
 Ifc1::system::CErrorCategory, 17

mvGetLevelCount
 Ifc1::filesystem::CRecDirectory, 19

mvGetNumEntries
 Ifc1::filesystem::CDirectory, 15
 Ifc1::filesystem::CRecDirectory, 20

mvGetPath
 Ifc1::filesystem::CDirectory, 15
 Ifc1::filesystem::CRecDirectory, 20

mvRefresh
 Ifc1::filesystem::CDirectory, 15
 Ifc1::filesystem::CRecDirectory, 20

name
 Ifc1::system::CErrorCategory, 17

operator=
 Ifc1::filesystem::CDirectory, 15
 Ifc1::filesystem::CRecDirectory, 20

smrGetErrorCategory
 Ifc1::system::CErrorCategory, 17

swap
 Filesystem class library, 11
 Ifc1::filesystem::CDirectory, 16
 Ifc1::filesystem::CRecDirectory, 21

System class library, 9
 grCategory, 10