LFC1 Libraries

0.0.1

Generated by Doxygen 1.8.3.1

Mon Jun 24 2013 19:47:32

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

Hierarchical Index

3.1 Class Hierarchy

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

Chapter 4

Class Index

4.1 Class List

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

Chapter 5

Module Documentation

5.1 Date and time class library

Classes

- class lfc1::datetime::CDateDuration

  This class represents date durations.
- class lfc1::datetime::CDate

  This class represents a local date.
- class lfc1::datetime::CDateBase

  This class serves as the base class for date input/output facets.
- class lfc1::datetime::CDateGet

  This class is a CDate class input facet.
- class lfc1::datetime::CDatePut

  This class is a CDate class output facet.
- class lfc1::datetime::CDateManip

  This class is a helper class for the date manipulator. This class was derived from Sections 3.3.1.3 to 3.3.1.7 of the book Standard C++ IOStreams and Locales.
- class lfc1::datetime::CDateTimeDuration

  This class represents date and time durations.
- class lfc1::datetime::CDateTime

  This class represents the time since 01-Jan-1970.
- class lfc1::datetime::CDateTimeBase

  This class serves as the base class for date and time input/output facets.
- class lfc1::datetime::CDateTimeGet

  This class is a CDateTime class input facet.
- class lfc1::datetime::CDateTimePut

  This class is a CDateTime class output facet.
- class lfc1::datetime::CDateTimeManip

  This class is a helper class for the date and time manipulator. This class was derived from Section 3.3.1.3 to 3.3.1.7 of the book Standard C++ IOStreams and Locales.
- class lfc1::datetime::CFacetBase

  This class serves as the base class for date and time input/output facets.
- class lfc1::datetime::CLangInfo

  This class obtains locale specific date and time strings.
- class lfc1::datetime::CTimeDuration

  This class represents time durations.
- class lfc1::datetime::CTime

This class represents the time since midnight.

- class lfc1::datetime::CTimeBase

  This class serves as the base class for time input/output facets.
- class lfc1::datetime::CTimeGet

  This class is a CTime class input facet.
- class lfc1::datetime::CTimePut

  This class is a CTime class output facet.
- class lfc1::datetime::CTimeManip

  This class is a helper class for the time manipulator. This class was derived from Sections 3.3.1.3 to 3.3.1.7 of the book Standard C++ IOStreams and Locales.

Functions

- void lfc1::datetime::swap (CDate &arLHS, CDate &arRHS) noexcept

  This function overloads std::swap for CDate objects.
- std::istream & lfc1::datetime::operator>> (std::istream &arStream, CDate &arDate)

  This function receives a CDate object from an input stream.
- std::ostream & lfc1::datetime::operator<< (std::ostream &arStream, const CDate &arDate)

  This function sends a CDate object to an output stream.
- void lfc1::datetime::swap (CDateDuration &arLHS, CDateDuration &arRHS) noexcept

  This function overloads std::swap for CDateDuration objects.
- void lfc1::datetime::swap (CDateTime &arLHS, CDateTime &arRHS) noexcept

  This function overloads std::swap for CDateTime objects.
- std::istream & lfc1::datetime::operator>> (std::istream &arStream, CDateTime &arDateTime)

  This function receives a CDateTime object from an input stream.
- std::ostream & lfc1::datetime::operator<< (std::ostream &arStream, const CDateTime &arDateTime)

  This function sends a CDateTime object to an output stream.
- void lfc1::datetime::swap (CDateTimeDuration &arLHS, CDateTimeDuration &arRHS) noexcept

  This function overloads std::swap for CDateTimeDuration objects.
- void lfc1::datetime::swap (CTime &arLHS, CTime &arRHS) noexcept

  This function overloads std::swap for CTime objects.
- std::istream & lfc1::datetime::operator>> (std::istream &arStream, CTime &arTime)

  This function receives a CTime object from an input stream.
- std::ostream & lfc1::datetime::operator<< (std::ostream &arStream, const CTime &arTime)

  This function sends a CTime object to an output stream.
- void lfc1::datetime::swap (CTimeDuration &arLHS, CTimeDuration &arRHS) noexcept

  This function overloads std::swap for CTimeDuration objects.

5.1.1 Detailed Description

This library contains a set of date and time classes which provide millisecond precision.

5.1.2 Function Documentation

5.1.2.1 std::ostream & lfc1::datetime::operator<< (std::ostream & *arStream*, const CTime & *arTime*)

This function sends a CTime object to an output stream.

Parameters

in	*arStream*	The destination output stream.
in	*arTime*	The CTime object to send.

Returns

The destination output stream.

5.1.2.2 std::ostream & lfc1::datetime::operator<< (std::ostream & *arStream*, const CDateTime & *arDateTime*)

This function sends a CDateTime object to an output stream.

Parameters

in	*arStream*	The destination output stream.
in	*arDateTime*	The CDateTime object to send.

Returns

The destination output stream.

5.1.2.3 std::ostream & lfc1::datetime::operator<< (std::ostream & *arStream*, const CDate & *arDate*)

This function sends a CDate object to an output stream.

Parameters

in	*arStream*	The destination output stream.
in	*arDate*	The CDate object to send.

Returns

The destination output stream.

5.1.2.4 std::istream & lfc1::datetime::operator>> (std::istream & *arStream*, CTime & *arTime*)

This function receives a CTime object from an input stream.

Parameters

in	*arStream*	The source input stream.
out	*arTime*	The received CTime object.

Returns

The source input stream.

5.1.2.5 std::istream & lfc1::datetime::operator>> (std::istream & *arStream*, CDateTime & *arDateTime*)

This function receives a CDateTime object from an input stream.

Parameters

in	*arStream*	The source input stream.
out	*arDateTime*	The received CDateTime object.

Returns

The source input stream.

5.1.2.6 std::istream & lfc1::datetime::operator>> (std::istream & *arStream*, CDate & *arDate*)

This function receives a CDate object from an input stream.

Parameters

in	*arStream*	The source input stream.
out	*arDate*	The received CDate object.

Returns

The source input stream.

5.1.2.7 void lfc1::datetime::swap (CTimeDuration & *arLHS*, CTimeDuration & *arRHS*) [noexcept]

This function overloads std::swap for CTimeDuration objects.

Parameters

in,out	*arLHS*	The first object to swap.
in,out	*arRHS*	The second object to swap.

5.1.2.8 void lfc1::datetime::swap (CDateDuration & *arLHS*, CDateDuration & *arRHS*) [noexcept]

This function overloads std::swap for CDateDuration objects.

Parameters

in,out	*arLHS*	The first object to swap.
in,out	*arRHS*	The second object to swap.

5.1.2.9 void lfc1::datetime::swap (CDateTimeDuration & *arLHS*, CDateTimeDuration & *arRHS*) [noexcept]

This function overloads std::swap for CDateTimeDuration objects.

Parameters

in,out	*arLHS*	The first object to swap.
in,out	*arRHS*	The second object to swap.

5.1.2.10 void lfc1::datetime::swap (CTime & *arLHS*, CTime & *arRHS*) [noexcept]

This function overloads std::swap for CTime objects.

Parameters

in,out	*arLHS*	The first object to swap.
in,out	*arRHS*	The second object to swap.

5.1.2.11 void lfc1::datetime::swap (CDateTime & *arLHS*, CDateTime & *arRHS*) [noexcept]

This function overloads std::swap for CDateTime objects.

Parameters

in,out	*arLHS*	The first object to swap.
in,out	*arRHS*	The second object to swap.

5.1.2.12 void lfc1::datetime::swap (CDate & *arLHS*, CDate & *arRHS*) [noexcept]

This function overloads std::swap for CDate objects.

Parameters

in,out	*arLHS*	The first object to swap.
in,out	*arRHS*	The second object to swap.

Chapter 6

Class Documentation

6.1 lfc1::datetime::CDate Class Reference

This class represents a local date.

`#include <lfc1/datetime/cdate.hpp>`

Classes

- struct SDate

  A structure that holds the individual parts of a date.

Public Member Functions

- CDate () noexcept

  This function creates a CDate object based on the current system date.
- CDate (int avDateNo)

  This function creates a CDate object using the given Julian day number.
- CDate (int avYear, int avMonth, int avDay)

  This function creates a CDate object using the given year, month and day.
- CDate (const CDate &arRHS) noexcept

  This function copy constructs a CDate object.
- CDate (CDate &&arRHS) noexcept

  This function move constructs a CDate object.
- ~CDate () noexcept

  This function destroys a CDate object.
- CDate & operator= (int avDateNo)

  This function assigns the given Julian day number to the CDate object.
- CDate & operator= (const CDate &arRHS) noexcept

  This function assigns a CDate object to another CDate object.
- CDate & operator= (CDate &&arRHS) noexcept

  This function moves a CDate object to another CDate object.
- operator int () const noexcept

  This function obtains the equivalent Julian day number of a CDate object.
- void mvSetDate () noexcept

  This function sets a CDate object based on the current system date.
- void mvSetDate (int avYear, int avMonth, int avDay)

This function sets a CDate object based on the given year, month and day.

- SDate mvGetDate () const noexcept

  This function obtains the equivalent year, month and day of the CDate object.

- void swap (CDate &arRHS) noexcept

  This function swaps a CDate object with another CDate object.

- CDate & operator++ ()

  This function pre-increments a CDate object by one.

- CDate operator++ (int)

  This function post-increments a CDate object by one.

- CDate & operator-- ()

  This function pre-decrements a CDate object by one.

- CDate operator-- (int)

  This function post-decrements a CDate object by one.

- CDate & operator+= (const CDateDuration &arDateDuration)

  This function increments the CDate object by the given duration.

- CDate & operator-= (const CDateDuration &arDateDuration)

  This function decrements the CDate object by the given duration.

- std::ios_base::iostate mvInput (std::istream &arStream)

  This function receives a CDate object from the given input stream.

- std::ios_base::iostate mvOutput (std::ostream &arStream) const

  This function sends a CDate object to the given output stream.

Static Public Member Functions

- static int smvGetIndex () noexcept

  This function obtains the allocated iword/pword index.

- static bool smvIsLeapYear (int avYear) noexcept

  This function determines whether the given year is a leap year or not.

- static int smvGetMaxDay (int avYear, int avMonth) noexcept

  This function obtains the maximum number of days for a month of a given year.

- static void smvValidateDate (int avYear, int avMonth, int avDay)

  This function validates a date to be in the supported date range.

Static Public Attributes

- static const auto MIN_YEAR = 0

  The minimum year.

- static const auto MAX_YEAR = 9999

  The maximum year.

- static const auto MONTH_PER_YEAR = 12

  The number of months in a year.

- static const auto MIN_MONTH = 1

  The lowest month number in a year.

- static const auto MAX_MONTH = 12

  The highest month number in a year.

- static const auto DAY_PER_WEEK = 7

  The number of days per week.

- static const auto MIN_DAY = 1

  The lowest day number in a month.

- static const auto MAX_DAY = 31

The highest day number in a month.

- static const auto FIRST_DAY_MON = 0
- static const auto FIRST_DAY_SUN = 6
- static const auto MIN_DATENO = 2298874

  Julian day number for 01-Jan-1582.
- static const auto MAX_DATENO = 2817152

  Julian day number for 31-Dec-3000.
- static const auto DATENO_OFFSET_SUN = 1
- static const auto MIN_DATE_YEAR = 1582
- static const auto MAX_DATE_YEAR = 3000

  The highest supported year for 64-bit environment.

6.1.1 Detailed Description

This class represents a local date.

Note that this class has no timezone dependency.

This class supports the following operators:

```
binary arithmetic operators: +=, -=
unary arithmetic operators:  ++, --
```

6.1.2 Constructor & Destructor Documentation

6.1.2.1 lfc1::datetime::CDate::CDate (int *avDateNo*)

This function creates a CDate object using the given Julian day number.

Parameters

in	*avDateNo*	The Julian day number to assign to the CDate object.

Exceptions

std::out_of_range	Indicates that the given Julian day number is not within the valid range.

6.1.2.2 lfc1::datetime::CDate::CDate (int *avYear,* int *avMonth,* int *avDay*)

This function creates a CDate object using the given year, month and day.

Parameters

in	*avYear*	The given year.
in	*avMonth*	The given month.
in	*avDay*	The given day.

6.1.2.3 lfc1::datetime::CDate::CDate (const CDate & *arRHS*) [noexcept]

This function copy constructs a CDate object.

Parameters

in	*arRHS*	The CDate object to be copied.

6.1.2.4 lfc1::datetime::CDate::CDate (CDate && *arRHS*) [noexcept]

This function move constructs a CDate object.

Parameters

in	*arRHS*	The CDate object to be moved.

6.1.3 Member Function Documentation

6.1.3.1 CDate::SDate lfc1::datetime::CDate::mvGetDate ( ) const [noexcept]

This function obtains the equivalent year, month and day of the CDate object.

Returns

The equivalent year, month and day of the CDate object.

6.1.3.2 std::ios_base::iostate lfc1::datetime::CDate::mvInput (std::istream & *arStream*)

This function receives a CDate object from the given input stream.

Parameters

in	*arStream*	The source stream.

Returns

The resulting stream state.

6.1.3.3 std::ios_base::iostate lfc1::datetime::CDate::mvOutput (std::ostream & *arStream*) const

This function sends a CDate object to the given output stream.

Parameters

in	*arStream*	The destination stream.

Returns

The resulting stream state.

6.1.3.4 void lfc1::datetime::CDate::mvSetDate (int *avYear*, int *avMonth*, int *avDay*)

This function sets a CDate object based on the given year, month and day.

Parameters

in	*avYear*	The given year.
in	*avMonth*	The given month.
in	*avDay*	The given day.

Exceptions

std::out_of_range	Indicates the given values are not within the valid ranges.

6.1.3.5 lfc1::datetime::CDate::operator int ( ) const [noexcept]

This function obtains the equivalent Julian day number of a CDate object.

Returns

The equivalent Julian day number of a CDate object.

6.1.3.6 CDate & lfc1::datetime::CDate::operator++ ( )

This function pre-increments a CDate object by one.

Returns

The CDate object.

Exceptions

std::out_of_range	Indicates incrementing the CDate object will make the Julian day number go above the maximum supported date.

6.1.3.7 CDate lfc1::datetime::CDate::operator++ (int)

This function post-increments a CDate object by one.

Returns

The CDate object before the increment.

6.1.3.8 CDate & lfc1::datetime::CDate::operator+= (const CDateDuration & *arDateDuration*)

This function increments the CDate object by the given duration.

Parameters

in	*arDateDuration*	The duration to add to the date.

Returns

The CDate object.

Exceptions

std::out_of_range	Indicates incrementing the CDate object will make the Julian day number go above the maximum supported date.

6.1.3.9 CDate & lfc1::datetime::CDate::operator-- ( )

This function pre-decrements a CDate object by one.

Returns

The CDate object.

Exceptions

std::out_of_range	Indicates decrementing the CDate object will make the Julian day number go below the minimum supported date.

6.1.3.10 CDate lfc1::datetime::CDate::operator-- (int)

This function post-decrements a CDate object by one.

Returns

The CDate object before the decrement.

6.1.3.11 CDate & lfc1::datetime::CDate::operator-= (const CDateDuration & *arDateDuration*)

This function decrements the CDate object by the given duration.

Parameters

in	*arDateDuration*	The duration to subtract from the date.

Returns

The CDate object.

Exceptions

std::out_of_range	Indicates decrementing the CDate object will make the Julian day number go below the minimum supported date.

6.1.3.12 CDate & lfc1::datetime::CDate::operator= (int *avDateNo*)

This function assigns the given Julian day number to the CDate object.

Parameters

in	*avDateNo*	The Julian day number to assign to the CDate object.

Returns

The CDate object assigned to.

6.1.3.13 CDate & lfc1::datetime::CDate::operator= (const CDate & *arRHS*) [noexcept]

This function assigns a CDate object to another CDate object.

Parameters

in	*arRHS*	The CDate object to assign to another CDate object.

Returns

The CDate object assigned to.

6.1.3.14 CDate & lfc1::datetime::CDate::operator= (CDate && *arRHS*) [noexcept]

This function moves a CDate object to another CDate object.

Parameters

in	*arRHS*	The CDate object to move to another CDate object.

Returns

The CDate object assigned to.

6.1.3.15 int lfc1::datetime::CDate::smvGetIndex ( ) [static], [noexcept]

This function obtains the allocated iword/pword index.

Returns

The allocated iword/pword index.

6.1.3.16 int lfc1::datetime::CDate::smvGetMaxDay (int *avYear*, int *avMonth*) [static], [noexcept]

This function obtains the maximum number of days for a month of a given year.

Parameters

in	*avYear*	The year.
in	*avMonth*	The month.

Returns

The maximum number of days for the given year and month.

6.1.3.17 bool lfc1::datetime::CDate::smvIsLeapYear (int *avYear*) [static], [noexcept]

This function determines whether the given year is a leap year or not.

Parameters

in	*avYear*	The year.

Return values

true	The year is a leap year.
false	The year is not a leap year.

6.1.3.18 void lfc1::datetime::CDate::smvValidateDate (int *avYear,* int *avMonth,* int *avDay*) [static]

This function validates a date to be in the supported date range.

Parameters

in	*avYear*	The year of a date. (0 - 9999)
in	*avMonth*	The month of a date. (1 - 12)
in	*avDay*	The day of a date. (1 - 31)

Exceptions

std::out_of_range	Indicates either the year, month or day are not within valid ranges.

6.1.3.19 void lfc1::datetime::CDate::swap (CDate & *arRHS*) [noexcept]

This function swaps a CDate object with another CDate object.

Parameters

in,out	*arRHS*	The CDate object to swap with.

6.1.4 Member Data Documentation

6.1.4.1 const auto lfc1::datetime::CDate::DATENO_OFFSET_SUN = 1 [static]

The number to add to a Julian day number such that

when the resulting number is divided by 7, the remainder is the week day number with Sunday being 0.

6.1.4.2 const auto lfc1::datetime::CDate::FIRST_DAY_MON = 0 [static]

The number used to indicate that the first day of the

week is Monday.

6.1.4.3 const auto lfc1::datetime::CDate::FIRST_DAY_SUN = 6 [static]

The number used to indicate that the first day of the

week is Sunday.

6.1.4.4 const auto lfc1::datetime::CDate::MIN_DATE_YEAR = 1582 [static]

The lowest supported year. This is the year that the

Gregorian calendar started.

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

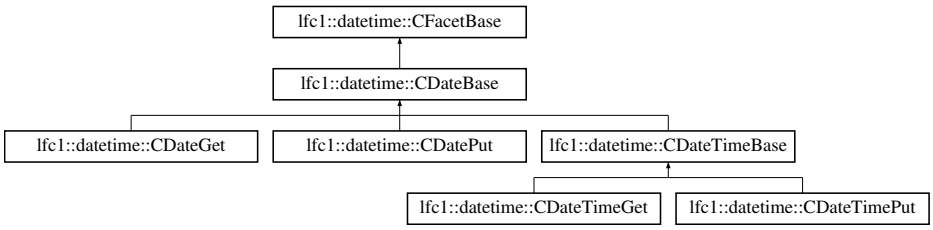
- include/lfc1/datetime/cdate.hpp
- datetime/library/src/cdate.cpp

6.2 lfc1::datetime::CDateBase Class Reference

This class serves as the base class for date input/output facets.

`#include <lfc1/datetime/cdate.hpp>`

Inheritance diagram for lfc1::datetime::CDateBase:



Public Types

- typedef
  std::istreambuf_iterator< char > TInIt

  This type is the input iterator.
- typedef
  std::ostreambuf_iterator< char > TOutIt

  This type is the output iterator.

Protected Member Functions

- CDateBase () noexcept

  This function creates a default CDateBase object.
- CDateBase (const CDateBase &)=delete

  Not supported.
- CDateBase (CDateBase &&)=delete

  Not supported.
- ~CDateBase () noexceptoverride

  This function destroys a CDateBase object.
- CDateBase & operator= (const CDateBase &)=delete

  Not supported.
- CDateBase & operator= (CDateBase &&)=delete

  Not supported.

Static Protected Member Functions

- static std::string smvExpandFormat (std::string avFormat)

  This function expands the date format by replacing format specifiers with their equivalent format specifiers.
- static TInIt smvGetWeekDayName (TInIt avNext, TInIt avEnd, std::ios_base &arIosBase, std::ios_base-::iostate &arIoState, int &arWeekDayNo, bool avIsShort)

  This function parses an input iterator for a week day name.
- static TInIt smvGetMonthName (TInIt avNext, TInIt avEnd, std::ios_base &arIosBase, std::ios_base::iostate &arIoState, int &arMonthNo, bool avIsShort)

  This function parses an input iterator for a month name.
- static TOutIt smvPutWeekDayName (TOutIt avOut, int avWeekDayNo, bool avIsShort)

  This function sends the week day name of a given week day to an output iterator.

- static TOutIt smvPutMonthName (TOutIt avOut, int avMonthNo, bool avIsShort)

  This function sends the month name of a given month to an output iterator.
- static void smvCalcWeekBasedDetails (const CDate &arDate, int &arWeekBasedYear, int *apWeekBasedWeekNo) noexcept

  This function determines the week-based year and week-based week number of a given date.
- static int smvCalcWeekNo (const CDate &arDate, int avYear, int avDay) noexcept

  This function determines the week number of a given date within a given year.
- static TInIt smvIgnoreChar (TInIt avNext, TInIt avEnd, std::ios_base::iostate &arIoState, char avExpectedChar) noexcept

  This function parses an input iterator for a specific character. This function moves the input iterator forward one character if the expected character is found.
- static TInIt smvGetNumber (TInIt avNext, TInIt avEnd, std::ios_base &arIosBase, std::ios_base::iostate &arIoState, int *apNumber, int avDigits, char avPadding) noexcept

  This function parses an input iterator for an integer value.
- static TOutIt smvPutNumber1 (TOutIt avOut, int avNumber) noexcept

  This function sends a 1-digit number to an output iterator.
- static TOutIt smvPutNumber2Zero (TOutIt avOut, int avNumber) noexcept

  This function sends a 2-digit zero padded number to an output iterator.
- static TOutIt smvPutNumber2Blank (TOutIt avOut, int avNumber) noexcept

  This function sends a 2-digit blank padded number to an output iterator.
- static TOutIt smvPutNumber3Zero (TOutIt avOut, int avNumber) noexcept

  This function sends a 3-digit zero padded number to an output iterator.
- static TOutIt smvPutNumber4Zero (TOutIt avOut, int avNumber) noexcept

  This function sends a 4-digit zero padded number to an output iterator.

6.2.1 Detailed Description

This class serves as the base class for date input/output facets.

6.2.2 Member Function Documentation

6.2.2.1 void lfc1::datetime::CDateBase::smvCalcWeekBasedDetails (const CDate & *arDate*, int & *arWeekBasedYear*, int * *apWeekBasedWeekNo*) [static], [protected], [noexcept]

This function determines the week-based year and week-based week number of a given date.

Parameters

in	*arDate*	The given date expressed as a Julian day number.
in,out	*arWeekBasedYear*	Set to the starting week-based year on input and contains the resulting week-based year on output.
out	*apWeekBasedWeekNo*	The resulting week-based week number of the given date. If null, week-based week number is not stored.

6.2.2.2 int lfc1::datetime::CDateBase::smvCalcWeekNo (const CDate & *arDate*, int *avYear*, int *avDay*) [static], [protected], [noexcept]

This function determines the week number of a given date within a given year.

Parameters

in	*arDate*	The given date expressed as a Julian day number.
in	*avYear*	The year of the given date.

in	*avDay*	The starting day of the week. (0 or 6 = Monday or Sunday)

Returns

The week number of a given date within a given year.

6.2.2.3 std::string lfc1::datetime::CDateBase::smvExpandFormat (std::string *avFormat*) [static], [protected]

This function expands the date format by replacing format specifiers with their equivalent format specifiers.

Parameters

in	*avFormat*	The date format to be expanded.

Returns

The expanded format string.

6.2.2.4 CDateBase::TInIt lfc1::datetime::CDateBase::smvGetMonthName (TInIt *avNext,* TInIt *avEnd,* std::ios_base & *arIosBase,* std::ios_base::iostate & *arIoState,* int & *arMonthNo,* bool *avIsShort*) [static], [protected]

This function parses an input iterator for a month name.

Parameters

in	*avNext*	The input iterator.
in	*avEnd*	The input end iterator.
in	*arIosBase*	The stream formatting information.
out	*arIoState*	The stream state.
out	*arMonthNo*	The storage for the resulting month number.
in	*avIsShort*	Short (true) or long (false) month name indicator.

Returns

The input iterator.

6.2.2.5 CFacetBase::TInIt lfc1::datetime::CFacetBase::smvGetNumber (TInIt *avNext,* TInIt *avEnd,* std::ios_base & *arIosBase,* std::ios_base::iostate & *arIoState,* int * *apNumber,* int *avDigits,* char *avPadding*) [static], [protected], [noexcept], [inherited]

This function parses an input iterator for an integer value.

Parameters

in	*avNext*	The input iterator.
in	*avEnd*	The input end iterator.
in	*arIosBase*	The stream formatting information.
out	*arIoState*	The stream state.
out	*apNumber*	The storage for the resulting integer.
in	*avDigits*	The maximum number of digits allowed for the integer.
in	*avPadding*	The padding character allowed for the integer value.

Return values

ios_base::goodbit	A valid integer was found.
ios_base::failbit	A valid integer was not found.

6.2.2.6 CDateBase::TInIt lfc1::datetime::CDateBase::smvGetWeekDayName (TInIt *avNext,* TInIt *avEnd,* std::ios_base & *arIosBase,* std::ios_base::iostate & *arIoState,* int & *arWeekDayNo,* bool *avIsShort*) [static],[protected]

This function parses an input iterator for a week day name.

Parameters

in	*avNext*	The input iterator.
in	*avEnd*	The input end iterator.
in	*arIosBase*	The stream formatting information.
out	*arIoState*	The stream state.
out	*arWeekDayNo*	The storage for the resulting day of the week. (0 = Sunday)
in	*avIsShort*	Short (true) or long (false) week day name indicator.

Returns

The input iterator.

6.2.2.7 CFacetBase::TInIt lfc1::datetime::CFacetBase::smvIgnoreChar (TInIt *avNext,* TInIt *avEnd,* std::ios_base::iostate & *arIoState,* char *avExpectedChar*) [static],[protected],[noexcept],[inherited]

This function parses an input iterator for a specific character. This function moves the input iterator forward one character if the expected character is found.

Parameters

in	*avNext*	The input iterator.
in	*avEnd*	The input end iterator.
out	*arIoState*	The stream state.
in	*avExpectedChar*	The next character expected from the input iterator.

Return values

ios_base::goodbit	The expected character was found.
ios_base::failbit	The expected character was not found.

6.2.2.8 CDateBase::TOutIt lfc1::datetime::CDateBase::smvPutMonthName (TOutIt *avOut,* int *avMonthNo,* bool *avIsShort*) [static],[protected]

This function sends the month name of a given month to an output iterator.

Parameters

in	*avOut*	The output iterator.
in	*avMonthNo*	The number of the month.
in	*avIsShort*	Short (true) or long (false) month name indicator.

Returns

The output iterator.

6.2.2.9 CFacetBase::TOutIt lfc1::datetime::CFacetBase::smvPutNumber1 (TOutIt *avOut,* int *avNumber*) [static], [protected], [noexcept], [inherited]

This function sends a 1-digit number to an output iterator.

Parameters

in	*avOut*	The output iterator.
in	*avNumber*	The 1-digit number.

Returns

The output iterator.

6.2.2.10 CFacetBase::TOutIt lfc1::datetime::CFacetBase::smvPutNumber2Blank (TOutIt *avOut,* int *avNumber*) [static], [protected], [noexcept], [inherited]

This function sends a 2-digit blank padded number to an output iterator.

Parameters

in	*avOut*	The output iterator.
in	*avNumber*	The 2-digit number.

Returns

The output iterator.

6.2.2.11 CFacetBase::TOutIt lfc1::datetime::CFacetBase::smvPutNumber2Zero (TOutIt *avOut,* int *avNumber*) [static], [protected], [noexcept], [inherited]

This function sends a 2-digit zero padded number to an output iterator.

Parameters

in	*avOut*	The output iterator.
in	*avNumber*	The 2-digit number.

Returns

The output iterator.

6.2.2.12 CFacetBase::TOutIt lfc1::datetime::CFacetBase::smvPutNumber3Zero (TOutIt *avOut,* int *avNumber*) [static], [protected], [noexcept], [inherited]

This function sends a 3-digit zero padded number to an output iterator.

Parameters

in	*avOut*	The output iterator.
in	*avNumber*	The 3-digit number.

Returns

The output iterator.

6.2.2.13 CFacetBase::TOutIt lfc1::datetime::CFacetBase::smvPutNumber4Zero (TOutIt *avOut*, int *avNumber*) [static], [protected], [noexcept], [inherited]

This function sends a 4-digit zero padded number to an output iterator.

Parameters

in	*avOut*	The output iterator.
in	*avNumber*	The 4-digit number.

Returns

The output iterator.

6.2.2.14 CDateBase::TOutIt lfc1::datetime::CDateBase::smvPutWeekDayName (TOutIt *avOut*, int *avWeekDayNo*, bool *avIsShort*) [static], [protected]

This function sends the week day name of a given week day to an output iterator.

Parameters

in	*avOut*	The output iterator.
in	*avWeekDayNo*	The number of the week day.
in	*avIsShort*	Short (true) or long (false) week day name indicator.

Returns

The output iterator.

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

- include/lfc1/datetime/cdate.hpp
- datetime/library/src/cdatebase.cpp

6.3 lfc1::datetime::CDateDuration Class Reference

This class represents date durations.

`#include <lfc1/datetime/cdate.hpp>`

Public Types

- enum EUnits {
  E_U_YEAR,
  E_U_MONTH,
  E_U_WEEK,
  E_U_DAY }

  < *The list of date durations.*

Public Member Functions

- CDateDuration (int avDuration, EUnits avUnit=E_U_DAY) noexcept

  This function creates a CDateDuration object.
- CDateDuration (const CDateDuration &arRHS) noexcept

  This function copy constructs a CDateDuration object.
- CDateDuration (CDateDuration &&arRHS) noexcept

  This function move constructs a CDateDuration object.
- ~CDateDuration () noexcept

  This function destroys a CDateDuration object.
- CDateDuration & operator= (const CDateDuration &arRHS) noexcept

  This function assigns a CDateDuration object to another CDateDuration object.
- CDateDuration & operator= (CDateDuration &&arRHS) noexcept

  This function moves a CDateDuration object to another CDateDuration object.
- int mvGetDuration () const noexcept

  This function obtains the date duration.
- EUnits mvGetUnit () const noexcept

  This function obtains the unit of the date duration.
- void swap (CDateDuration &arRHS) noexcept

  This function swaps a CDateDuration object with another CDateDuration object.

6.3.1 Detailed Description

This class represents date durations.

The purpose of this class is to support the increment and decrement operators of the CDate class.

6.3.2 Constructor & Destructor Documentation

6.3.2.1 lfc1::datetime::CDateDuration::CDateDuration (int *avDuration*, EUnits *avUnit* = E_U_DAY) [noexcept]

This function creates a CDateDuration object.

Parameters

in	*avDuration*	The date duration.
in	*avUnit*	The unit of measure of the date duration.

6.3.2.2 lfc1::datetime::CDateDuration::CDateDuration (const CDateDuration & *arRHS*) [noexcept]

This function copy constructs a CDateDuration object.

Parameters

in	*arRHS*	The CDateDuration object to be copied.

6.3.2.3 lfc1::datetime::CDateDuration::CDateDuration (CDateDuration && *arRHS*) [noexcept]

This function move constructs a CDateDuration object.

Parameters

in	*arRHS*	The CDateDuration object to be moved.

6.3.3 Member Function Documentation

6.3.3.1 int lfc1::datetime::CDateDuration::mvGetDuration ( ) const [noexcept]

This function obtains the date duration.

Returns

The date duration.

6.3.3.2 CDateDuration::EUnits lfc1::datetime::CDateDuration::mvGetUnit ( ) const [noexcept]

This function obtains the unit of the date duration.

Returns

The unit of the date duration.

6.3.3.3 CDateDuration & lfc1::datetime::CDateDuration::operator= (const CDateDuration & *arRHS*) [noexcept]

This function assigns a CDateDuration object to another CDateDuration object.

Parameters

in	*arRHS*	The CDateDuration object to assign to another CDateDuration object.

Returns

The CDateDuration object assigned to.

6.3.3.4 CDateDuration & lfc1::datetime::CDateDuration::operator= (CDateDuration && *arRHS*) [noexcept]

This function moves a CDateDuration object to another CDateDuration object.

Parameters

in	*arRHS*	The CDateDuration object to move to another CDateDuration object.

Returns

The CDateDuration object assigned to.

6.3.3.5 void lfc1::datetime::CDateDuration::swap (CDateDuration & *arRHS*) [noexcept]

This function swaps a CDateDuration object with another CDateDuration object.

Parameters

in,out	*arRHS*	The CDateDuration object to swap with.

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

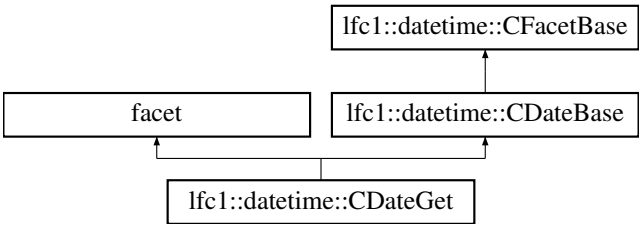
- include/lfc1/datetime/cdate.hpp
- datetime/library/src/cdateduration.cpp

6.4 lfc1::datetime::CDateGet Class Reference

This class is a CDate class input facet.

`#include <lfc1/datetime/cdate.hpp>`

Inheritance diagram for lfc1::datetime::CDateGet:



Public Types

- typedef
  std::istreambuf_iterator< char > TInIt
  This type is the input iterator.
- typedef
  std::ostreambuf_iterator< char > TOutIt
  This type is the output iterator.

Public Member Functions

- CDateGet (size_t avRelease=0)
  This function creates a CDateGet object.
- CDateGet (const CDateGet &)=delete
  Not supported.
- CDateGet (CDateGet &&)=delete
  Not supported.
- ~CDateGet () noexceptfinal
  This function destroys a CDateGet object.
- CDateGet & operator= (const CDateGet &)=delete
  Not supported.
- CDateGet & operator= (CDateGet &&)=delete
  Not supported.
- TInIt mvGet (TInIt avNext, TInIt avEnd, std::ios_base &arIosBase, std::ios_base::iostate &arIoState, CDate &arDate, const std::string &arFormat) const
  This function inputs a date from a stream. Duplication of date parts is not allowed.

Static Public Attributes

- static std::locale::id id
  The facet ID.

Protected Member Functions

- virtual TInIt mvDoGet (TInIt avNext, TInIt avEnd, std::ios_base &arIosBase, std::ios_base::iostate &arIoState, CDate &arDate, const std::string &arFormat) const

  This function implements the behavior of the mvGet() function.

Static Protected Member Functions

- static std::string smvExpandFormat (std::string avFormat)

  This function expands the date format by replacing format specifiers with their equivalent format specifiers.
- static TInIt smvGetWeekDayName (TInIt avNext, TInIt avEnd, std::ios_base &arIosBase, std::ios_base::iostate &arIoState, int &arWeekDayNo, bool avIsShort)

  This function parses an input iterator for a week day name.
- static TInIt smvGetMonthName (TInIt avNext, TInIt avEnd, std::ios_base &arIosBase, std::ios_base::iostate &arIoState, int &arMonthNo, bool avIsShort)

  This function parses an input iterator for a month name.
- static TOutIt smvPutWeekDayName (TOutIt avOut, int avWeekDayNo, bool avIsShort)

  This function sends the week day name of a given week day to an output iterator.
- static TOutIt smvPutMonthName (TOutIt avOut, int avMonthNo, bool avIsShort)

  This function sends the month name of a given month to an output iterator.
- static void smvCalcWeekBasedDetails (const CDate &arDate, int &arWeekBasedYear, int *apWeekBasedWeekNo) noexcept

  This function determines the week-based year and week-based week number of a given date.
- static int smvCalcWeekNo (const CDate &arDate, int avYear, int avDay) noexcept

  This function determines the week number of a given date within a given year.
- static TInIt smvIgnoreChar (TInIt avNext, TInIt avEnd, std::ios_base::iostate &arIoState, char avExpectedChar) noexcept

  This function parses an input iterator for a specific character. This function moves the input iterator forward one character if the expected character is found.
- static TInIt smvGetNumber (TInIt avNext, TInIt avEnd, std::ios_base &arIosBase, std::ios_base::iostate &arIoState, int *apNumber, int avDigits, char avPadding) noexcept

  This function parses an input iterator for an integer value.
- static TOutIt smvPutNumber1 (TOutIt avOut, int avNumber) noexcept

  This function sends a 1-digit number to an output iterator.
- static TOutIt smvPutNumber2Zero (TOutIt avOut, int avNumber) noexcept

  This function sends a 2-digit zero padded number to an output iterator.
- static TOutIt smvPutNumber2Blank (TOutIt avOut, int avNumber) noexcept

  This function sends a 2-digit blank padded number to an output iterator.
- static TOutIt smvPutNumber3Zero (TOutIt avOut, int avNumber) noexcept

  This function sends a 3-digit zero padded number to an output iterator.
- static TOutIt smvPutNumber4Zero (TOutIt avOut, int avNumber) noexcept

  This function sends a 4-digit zero padded number to an output iterator.

6.4.1 Detailed Description

This class is a CDate class input facet.

Note

This class uses the Template Method design pattern.

6.4.2 Constructor & Destructor Documentation

6.4.2.1 lfc1::datetime::CDateGet::CDateGet (size_t *avRelease* = 0) [explicit]

This function creates a CDateGet object.

Parameters

in	*avRelease*	Indicates who controls the lifetime of the facet. (0 means locale)

6.4.3 Member Function Documentation

6.4.3.1 CDateGet::TInIt lfc1::datetime::CDateGet::mvDoGet (TInIt *avNext*, TInIt *avEnd*, std::ios_base & *arIosBase*, std::ios_base::iostate & *arIoState*, CDate & *arDate*, const std::string & *arFormat*) const [protected], [virtual]

This function implements the behavior of the mvGet() function.

Parameters

in	*avNext*	The input iterator.
in	*avEnd*	The input end iterator.
in	*arIosBase*	The stream formatting information.
out	*arIoState*	The stream state.
out	*arDate*	The date retrieved from the input stream.
in	*arFormat*	The format of the date retrieved from the input stream.

Returns

The input iterator.

6.4.3.2 CDateGet::TInIt lfc1::datetime::CDateGet::mvGet (TInIt *avNext*, TInIt *avEnd*, std::ios_base & *arIosBase*, std::ios_base::iostate & *arIoState*, CDate & *arDate*, const std::string & *arFormat*) const

This function inputs a date from a stream. Duplication of date parts is not allowed.

This function recognizes the following format specifiers which are identical to the date specific format specifiers of the C language strftime() function:

```
%% - A percent character.
%a - Abbreviated weekday name.
%A - Full weekday name.
%b - Abbreviated month name.
%B - Full month name.
%C - Century [00-99].
%d - Day of the month zero padded [01-31].
%D - Equivalent to %m/%d/%y.
%e - Day of the month blank padded [ 1-31].
%F - Equivalent to %Y-%m-%d.
%g - Week-based year without century [00-99].
%G - Week-based year with century [0000-9999].
%h - Equivalent to %b.
%j - Day of the year [001-366].
%m - Month number [01-12].
%n - Newline character.
%t - Horizontal tab character.
%u - Day of the week [1-7], Monday is 1.
%U - Week of the year [00-53], the first Sunday starts week 1.
%V - Week number of a week-based year [01-53].
%w - Day of the week [0-6], Sunday is 0.
%W - Week of the year [00-53], the first Monday starts week 1.
%x - Default date format.
%y - Year without century [00-99].
%Y - Year with century [0000-9999].
```

Any other sequence of characters not included in the above list will be taken literally.

The following combinations of format specifiers results into a valid CDate object:

```
%D
%F
%x
(%Y or (%y [and %C])) and (%b or %B or %m) and (%d or %e)
(%Y or (%y [and %C])) and (%j)
(%Y or (%y [and %C])) and (%U or %W) and (%a or %A or %u or %w)
(%G or (%g [and %C])) and (%V) and (%a or %A or %u or %w)
```

Parameters

in	*avNext*	The input iterator.
in	*avEnd*	The input end iterator.
in	*arIosBase*	The stream formatting information.
out	*arIoState*	The stream state.
out	*arDate*	The date retrieved from the input stream.
in	*arFormat*	The format of the date retrieved from the input stream.

Returns

The input iterator.

6.4.3.3 void lfc1::datetime::CDateBase::smvCalcWeekBasedDetails (const CDate & *arDate*, int & *arWeekBasedYear*, int * *apWeekBasedWeekNo*) [static], [protected], [noexcept], [inherited]

This function determines the week-based year and week-based week number of a given date.

Parameters

in	*arDate*	The given date expressed as a Julian day number.
in,out	*arWeekBased-Year*	Set to the starting week-based year on input and contains the resulting week-based year on output.
out	*apWeekBased-WeekNo*	The resulting week-based week number of the given date. If null, week-based week number is not stored.

6.4.3.4 int lfc1::datetime::CDateBase::smvCalcWeekNo (const CDate & *arDate*, int *avYear*, int *avDay*) [static], [protected], [noexcept], [inherited]

This function determines the week number of a given date within a given year.

Parameters

in	*arDate*	The given date expressed as a Julian day number.
in	*avYear*	The year of the given date.
in	*avDay*	The starting day of the week. (0 or 6 = Monday or Sunday)

Returns

The week number of a given date within a given year.

6.4.3.5 std::string lfc1::datetime::CDateBase::smvExpandFormat (std::string *avFormat*) [static], [protected], [inherited]

This function expands the date format by replacing format specifiers with their equivalent format specifiers.

Parameters

in	*avFormat*	The date format to be expanded.

Returns

The expanded format string.

6.4.3.6 CDateBase::TInIt lfc1::datetime::CDateBase::smvGetMonthName (TInIt *avNext,* TInIt *avEnd,* std::ios_base & *arIosBase,* std::ios_base::iostate & *arIoState,* int & *arMonthNo,* bool *avIsShort*) [static], [protected], [inherited]

This function parses an input iterator for a month name.

Parameters

in	*avNext*	The input iterator.
in	*avEnd*	The input end iterator.
in	*arIosBase*	The stream formatting information.
out	*arIoState*	The stream state.
out	*arMonthNo*	The storage for the resulting month number.
in	*avIsShort*	Short (true) or long (false) month name indicator.

Returns

The input iterator.

6.4.3.7 CFacetBase::TInIt lfc1::datetime::CFacetBase::smvGetNumber (TInIt *avNext,* TInIt *avEnd,* std::ios_base & *arIosBase,* std::ios_base::iostate & *arIoState,* int * *apNumber,* int *avDigits,* char *avPadding*) [static], [protected], [noexcept], [inherited]

This function parses an input iterator for an integer value.

Parameters

in	*avNext*	The input iterator.
in	*avEnd*	The input end iterator.
in	*arIosBase*	The stream formatting information.
out	*arIoState*	The stream state.
out	*apNumber*	The storage for the resulting integer.
in	*avDigits*	The maximum number of digits allowed for the integer.
in	*avPadding*	The padding character allowed for the integer value.

Return values

ios_base::goodbit	A valid integer was found.
ios_base::failbit	A valid integer was not found.

6.4.3.8 CDateBase::TInIt lfc1::datetime::CDateBase::smvGetWeekDayName (TInIt *avNext,* TInIt *avEnd,* std::ios_base & *arIosBase,* std::ios_base::iostate & *arIoState,* int & *arWeekDayNo,* bool *avIsShort*) [static], [protected], [inherited]

This function parses an input iterator for a week day name.

Parameters

in	*avNext*	The input iterator.
in	*avEnd*	The input end iterator.
in	*arIosBase*	The stream formatting information.
out	*arIoState*	The stream state.
out	*arWeekDayNo*	The storage for the resulting day of the week. (0 = Sunday)
in	*avIsShort*	Short (true) or long (false) week day name indicator.

Returns

The input iterator.

6.4.3.9 CFacetBase::TInIt lfc1::datetime::CFacetBase::smvIgnoreChar (TInIt *avNext,* TInIt *avEnd,* std::ios_base::iostate & *arIoState,* char *avExpectedChar*) [static],[protected],[noexcept],[inherited]

This function parses an input iterator for a specific character. This function moves the input iterator forward one character if the expected character is found.

Parameters

in	*avNext*	The input iterator.
in	*avEnd*	The input end iterator.
out	*arIoState*	The stream state.
in	*avExpectedChar*	The next character expected from the input iterator.

Return values

ios_base::goodbit	The expected character was found.
ios_base::failbit	The expected character was not found.

6.4.3.10 CDateBase::TOutIt lfc1::datetime::CDateBase::smvPutMonthName (TOutIt *avOut,* int *avMonthNo,* bool *avIsShort*) [static],[protected],[inherited]

This function sends the month name of a given month to an output iterator.

Parameters

in	*avOut*	The output iterator.
in	*avMonthNo*	The number of the month.
in	*avIsShort*	Short (true) or long (false) month name indicator.

Returns

The output iterator.

6.4.3.11 CFacetBase::TOutIt lfc1::datetime::CFacetBase::smvPutNumber1 (TOutIt *avOut,* int *avNumber*) [static],[protected],[noexcept],[inherited]

This function sends a 1-digit number to an output iterator.

Parameters

in	*avOut*	The output iterator.
in	*avNumber*	The 1-digit number.

Returns

The output iterator.

6.4.3.12 CFacetBase::TOutIt lfc1::datetime::CFacetBase::smvPutNumber2Blank (TOutIt *avOut,* int *avNumber*) [static],[protected],[noexcept],[inherited]

This function sends a 2-digit blank padded number to an output iterator.

Parameters

in	*avOut*	The output iterator.
in	*avNumber*	The 2-digit number.

Returns

The output iterator.

6.4.3.13 CFacetBase::TOutIt lfc1::datetime::CFacetBase::smvPutNumber2Zero (TOutIt *avOut,* int *avNumber*) [static],[protected],[noexcept],[inherited]

This function sends a 2-digit zero padded number to an output iterator.

Parameters

in	*avOut*	The output iterator.
in	*avNumber*	The 2-digit number.

Returns

The output iterator.

6.4.3.14 CFacetBase::TOutIt lfc1::datetime::CFacetBase::smvPutNumber3Zero (TOutIt *avOut,* int *avNumber*) [static],[protected],[noexcept],[inherited]

This function sends a 3-digit zero padded number to an output iterator.

Parameters

in	*avOut*	The output iterator.
in	*avNumber*	The 3-digit number.

Returns

The output iterator.

6.4.3.15 CFacetBase::TOutIt lfc1::datetime::CFacetBase::smvPutNumber4Zero (TOutIt *avOut,* int *avNumber*) [static],[protected],[noexcept],[inherited]

This function sends a 4-digit zero padded number to an output iterator.

Parameters

in	*avOut*	The output iterator.
in	*avNumber*	The 4-digit number.

Returns

The output iterator.

6.4.3.16 CDateBase::TOutIt lfc1::datetime::CDateBase::smvPutWeekDayName (TOutIt *avOut*, int *avWeekDayNo*, bool *avIsShort*) [static], [protected], [inherited]

This function sends the week day name of a given week day to an output iterator.

Parameters

in	*avOut*	The output iterator.
in	*avWeekDayNo*	The number of the week day.
in	*avIsShort*	Short (true) or long (false) week day name indicator.

Returns

The output iterator.

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

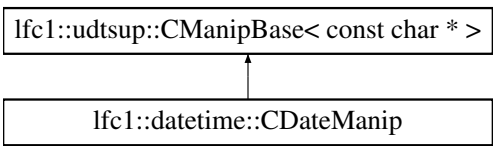
- include/lfc1/datetime/cdate.hpp
- datetime/library/src/cdateget.cpp

6.5 lfc1::datetime::CDateManip Class Reference

This class is a helper class for the date manipulator. This class was derived from Sections 3.3.1.3 to 3.3.1.7 of the book Standard C++ IOStreams and Locales.

`#include <lfc1/datetime/cdate.hpp>`

Inheritance diagram for lfc1::datetime::CDateManip:



Public Member Functions

- CDateManip (const char *apFormat)
  This function creates a CDateManip object.
- CDateManip (const CDateManip &)=default
  Uses default implementation.
- CDateManip & operator= (const CDateManip &)=default
  Uses default implementation.

Static Public Member Functions

- static std::string smvGetFormat (std::ios_base &arIosBase)
  This function obtains the date format from a stream.
- static std::ios_base::iostate smvCopyFmtErr (std::basic_ios< char > &arIos)
  This function obtains the error information relating to date format copying.

6.5.1 Detailed Description

This class is a helper class for the date manipulator. This class was derived from Sections 3.3.1.3 to 3.3.1.7 of the book Standard C++ IOStreams and Locales.

6.5.2 Constructor & Destructor Documentation

6.5.2.1 lfc1::datetime::CDateManip::CDateManip (const char * *apFormat*)

This function creates a CDateManip object.

Parameters

in	*apFormat*	The desired date format.

6.5.3 Member Function Documentation

6.5.3.1 std::ios_base::iostate lfc1::datetime::CDateManip::smvCopyFmtErr (std::basic_ios< char > & *arlos*) [static]

This function obtains the error information relating to date format copying.

Parameters

in	*arlos*	The stream containing the error information.

Returns

The error information relating to date format copying.

6.5.3.2 std::string lfc1::datetime::CDateManip::smvGetFormat (std::ios_base & *arlosBase*) [static]

This function obtains the date format from a stream.

Parameters

in	*arlosBase*	The stream containing the date format.

Returns

The date format.

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

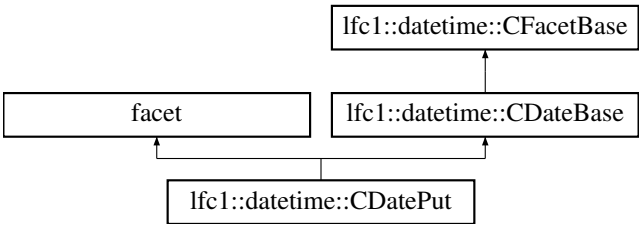
- include/lfc1/datetime/cdate.hpp
- datetime/library/src/cdatemanip.cpp

6.6 lfc1::datetime::CDatePut Class Reference

This class is a CDate class output facet.

`#include <lfc1/datetime/cdate.hpp>`

Inheritance diagram for lfc1::datetime::CDatePut:



Public Types

- typedef
  std::istreambuf_iterator< char > TInIt

  This type is the input iterator.
- typedef
  std::ostreambuf_iterator< char > TOutIt

  This type is the output iterator.

Public Member Functions

- CDatePut (size_t avRelease=0)

  This function creates a CDatePut object.
- CDatePut (const CDatePut &)=delete

  Not supported.
- CDatePut (CDatePut &&)=delete

  Not supported.
- ~CDatePut () noexceptfinal

  This function destroys a CDatePut object.
- CDatePut & operator= (const CDatePut &)=delete

  Not supported.
- CDatePut & operator= (CDatePut &&)=delete

  Not supported.
- TOutIt mvPut (TOutIt avOut, const CDate &arDate, const std::string &arFormat) const

  This function outputs a date to a stream.

Static Public Attributes

- static std::locale::id id

  The facet ID.

Protected Member Functions

- virtual TOutIt mvDoPut (TOutIt avOut, const CDate &arDate, const std::string &arFormat) const

  This function implements the behavior of the mvPut() function.

Static Protected Member Functions

- static std::string smvExpandFormat (std::string avFormat)

  This function expands the date format by replacing format specifiers with their equivalent format specifiers.
- static TInIt smvGetWeekDayName (TInIt avNext, TInIt avEnd, std::ios_base &arIosBase, std::ios_base-::iostate &arIoState, int &arWeekDayNo, bool avIsShort)

This function parses an input iterator for a week day name.

- static TInIt smvGetMonthName (TInIt avNext, TInIt avEnd, std::ios_base &arIosBase, std::ios_base::iostate &arIoState, int &arMonthNo, bool avIsShort)

  This function parses an input iterator for a month name.
- static TOutIt smvPutWeekDayName (TOutIt avOut, int avWeekDayNo, bool avIsShort)

  This function sends the week day name of a given week day to an output iterator.
- static TOutIt smvPutMonthName (TOutIt avOut, int avMonthNo, bool avIsShort)

  This function sends the month name of a given month to an output iterator.
- static void smvCalcWeekBasedDetails (const CDate &arDate, int &arWeekBasedYear, int *apWeekBasedWeekNo) noexcept

  This function determines the week-based year and week-based week number of a given date.
- static int smvCalcWeekNo (const CDate &arDate, int avYear, int avDay) noexcept

  This function determines the week number of a given date within a given year.
- static TInIt smvIgnoreChar (TInIt avNext, TInIt avEnd, std::ios_base::iostate &arIoState, char avExpectedChar) noexcept

  This function parses an input iterator for a specific character. This function moves the input iterator forward one character if the expected character is found.
- static TInIt smvGetNumber (TInIt avNext, TInIt avEnd, std::ios_base &arIosBase, std::ios_base::iostate &arIoState, int *apNumber, int avDigits, char avPadding) noexcept

  This function parses an input iterator for an integer value.
- static TOutIt smvPutNumber1 (TOutIt avOut, int avNumber) noexcept

  This function sends a 1-digit number to an output iterator.
- static TOutIt smvPutNumber2Zero (TOutIt avOut, int avNumber) noexcept

  This function sends a 2-digit zero padded number to an output iterator.
- static TOutIt smvPutNumber2Blank (TOutIt avOut, int avNumber) noexcept

  This function sends a 2-digit blank padded number to an output iterator.
- static TOutIt smvPutNumber3Zero (TOutIt avOut, int avNumber) noexcept

  This function sends a 3-digit zero padded number to an output iterator.
- static TOutIt smvPutNumber4Zero (TOutIt avOut, int avNumber) noexcept

  This function sends a 4-digit zero padded number to an output iterator.

6.6.1 Detailed Description

This class is a CDate class output facet.

Note

This class uses the Template Method design pattern.

6.6.2 Constructor & Destructor Documentation

6.6.2.1 lfc1::datetime::CDatePut::CDatePut (size_t *avRelease* = 0) [explicit]

This function creates a CDatePut object.

Parameters

in	*avRelease*	Indicates who controls the lifetime of the facet. (0 means locale)

6.6.3 Member Function Documentation

6.6.3.1 CDatePut::TOutIt lfc1::datetime::CDatePut::mvDoPut (TOutIt *avOut,* const CDate & *arDate,* const std::string & *arFormat*) const [protected], [virtual]

This function implements the behavior of the mvPut() function.

Parameters

in	*avOut*	The output iterator.
in	*arDate*	The date to be sent to the output stream.
in	*arFormat*	The format of the date to be sent to the output stream.

Returns

The output iterator.

6.6.3.2 CDatePut::TOutIt lfc1::datetime::CDatePut::mvPut (TOutIt *avOut,* const CDate & *arDate,* const std::string & *arFormat*) const

This function outputs a date to a stream.

This function recognizes the following format specifiers which are identical to the date specific format specifiers of the C language strftime() function:

```
%% - A percent character.
%a - Abbreviated weekday name.
%A - Full weekday name.
%b - Abbreviated month name.
%B - Full month name.
%C - Century [00-99].
%d - Day of the month zero padded [01-31].
%D - Equivalent to %m/%d/%y.
%e - Day of the month blank padded [ 1-31].
%F - Equivalent to %Y-%m-%d.
%g - Week-based year without century [00-99].   (See week-based year explanation below.)
%G - Week-based year with century [0000-9999].  (See week-based year explanation below.)
%h - Equivalent to %b.
%j - Day of the year [001-366].
%m - Month number [01-12].
%n - Newline character.
%t - Horizontal tab character.
%u - Day of the week [1-7], Monday is 1.
%U - Week of the year [00-53], the first Sunday starts week 1.
%V - Week number of a week-based year [01-53].  (See week-based year explanation below.)
%w - Day of the week [0-6], Sunday is 0.
%W - Week of the year [00-53], the first Monday starts week 1.
%x - Default date format.
%y - Year without century [00-99].
%Y - Year with century [0000-9999].
```

Any other sequence of characters not included in the above list will be taken literally.

The week-based year is the ISO 8601 week-based year. In this system, weeks begin on a Monday and week 1 of the year is the week that includes January 4th, which is also the week that includes the first Thursday of the year and is also the first week that contains at least four days in the year. If the first Monday of January is the 2nd, 3rd or 4th, the preceding days are part of the last week of the preceding year; thus, for Saturday 2nd January 1999, G is replaced by 1998 and V is replaced by 53. If December 29th, 30th or 31st is a Monday, it and any following days are part of week 1 of the following year. Thus for Tuesday 30th December 1997, G is replaced by 1998 and V is replaced by 01.

Parameters

in	*avOut*	The output iterator.
in	*arDate*	The date to be sent to the output stream.
in	*arFormat*	The format of the date to be sent to the output stream.

Returns

The output iterator.

6.6.3.3 void lfc1::datetime::CDateBase::smvCalcWeekBasedDetails (const CDate & *arDate,* int & *arWeekBasedYear,* int * *apWeekBasedWeekNo*) [static],[protected],[noexcept],[inherited]

This function determines the week-based year and week-based week number of a given date.

Parameters

in	*arDate*	The given date expressed as a Julian day number.
in,out	*arWeekBased-Year*	Set to the starting week-based year on input and contains the resulting week-based year on output.
out	*apWeekBased-WeekNo*	The resulting week-based week number of the given date. If null, week-based week number is not stored.

6.6.3.4 int lfc1::datetime::CDateBase::smvCalcWeekNo (const CDate & *arDate,* int *avYear,* int *avDay*) [static],[protected],[noexcept],[inherited]

This function determines the week number of a given date within a given year.

Parameters

in	*arDate*	The given date expressed as a Julian day number.
in	*avYear*	The year of the given date.
in	*avDay*	The starting day of the week. (0 or 6 = Monday or Sunday)

Returns

The week number of a given date within a given year.

6.6.3.5 std::string lfc1::datetime::CDateBase::smvExpandFormat (std::string *avFormat*) [static],[protected],[inherited]

This function expands the date format by replacing format specifiers with their equivalent format specifiers.

Parameters

in	*avFormat*	The date format to be expanded.

Returns

The expanded format string.

6.6.3.6 CDateBase::TInIt lfc1::datetime::CDateBase::smvGetMonthName (TInIt *avNext,* TInIt *avEnd,* std::ios_base & *arIosBase,* std::ios_base::iostate & *arIoState,* int & *arMonthNo,* bool *avIsShort*) [static],[protected],[inherited]

This function parses an input iterator for a month name.

Parameters

in	*avNext*	The input iterator.
in	*avEnd*	The input end iterator.
in	*arIosBase*	The stream formatting information.
out	*arIoState*	The stream state.
out	*arMonthNo*	The storage for the resulting month number.
in	*avIsShort*	Short (true) or long (false) month name indicator.

Returns

The input iterator.

6.6.3.7 CFacetBase::TInIt lfc1::datetime::CFacetBase::smvGetNumber (TInIt *avNext*, TInIt *avEnd*, std::ios_base & *arIosBase*, std::ios_base::iostate & *arIoState*, int * *apNumber*, int *avDigits*, char *avPadding*) [static], [protected], [noexcept], [inherited]

This function parses an input iterator for an integer value.

Parameters

in	*avNext*	The input iterator.
in	*avEnd*	The input end iterator.
in	*arIosBase*	The stream formatting information.
out	*arIoState*	The stream state.
out	*apNumber*	The storage for the resulting integer.
in	*avDigits*	The maximum number of digits allowed for the integer.
in	*avPadding*	The padding character allowed for the integer value.

Return values

ios_base::goodbit	A valid integer was found.
ios_base::failbit	A valid integer was not found.

6.6.3.8 CDateBase::TInIt lfc1::datetime::CDateBase::smvGetWeekDayName (TInIt *avNext*, TInIt *avEnd*, std::ios_base & *arIosBase*, std::ios_base::iostate & *arIoState*, int & *arWeekDayNo*, bool *avIsShort*) [static], [protected], [inherited]

This function parses an input iterator for a week day name.

Parameters

in	*avNext*	The input iterator.
in	*avEnd*	The input end iterator.
in	*arIosBase*	The stream formatting information.
out	*arIoState*	The stream state.
out	*arWeekDayNo*	The storage for the resulting day of the week. (0 = Sunday)
in	*avIsShort*	Short (true) or long (false) week day name indicator.

Returns

The input iterator.

6.6.3.9 CFacetBase::TInIt lfc1::datetime::CFacetBase::smvIgnoreChar (TInIt *avNext,* TInIt *avEnd,* std::ios_base::iostate & *arIoState,* char *avExpectedChar*) [static],[protected],[noexcept],[inherited]

This function parses an input iterator for a specific character. This function moves the input iterator forward one character if the expected character is found.

Parameters

in	*avNext*	The input iterator.
in	*avEnd*	The input end iterator.
out	*arIoState*	The stream state.
in	*avExpectedChar*	The next character expected from the input iterator.

Return values

ios_base::goodbit	The expected character was found.
ios_base::failbit	The expected character was not found.

6.6.3.10 CDateBase::TOutIt lfc1::datetime::CDateBase::smvPutMonthName (TOutIt *avOut,* int *avMonthNo,* bool *avIsShort*) [static],[protected],[inherited]

This function sends the month name of a given month to an output iterator.

Parameters

in	*avOut*	The output iterator.
in	*avMonthNo*	The number of the month.
in	*avIsShort*	Short (true) or long (false) month name indicator.

Returns

The output iterator.

6.6.3.11 CFacetBase::TOutIt lfc1::datetime::CFacetBase::smvPutNumber1 (TOutIt *avOut,* int *avNumber*) [static],[protected],[noexcept],[inherited]

This function sends a 1-digit number to an output iterator.

Parameters

in	*avOut*	The output iterator.
in	*avNumber*	The 1-digit number.

Returns

The output iterator.

6.6.3.12 CFacetBase::TOutIt lfc1::datetime::CFacetBase::smvPutNumber2Blank (TOutIt *avOut,* int *avNumber*) [static],[protected],[noexcept],[inherited]

This function sends a 2-digit blank padded number to an output iterator.

Parameters

in	*avOut*	The output iterator.
in	*avNumber*	The 2-digit number.

Returns

The output iterator.

6.6.3.13 CFacetBase::TOutIt lfc1::datetime::CFacetBase::smvPutNumber2Zero (TOutIt *avOut,* int *avNumber*) [static],[protected],[noexcept],[inherited]

This function sends a 2-digit zero padded number to an output iterator.

Parameters

in	*avOut*	The output iterator.
in	*avNumber*	The 2-digit number.

Returns

The output iterator.

6.6.3.14 CFacetBase::TOutIt lfc1::datetime::CFacetBase::smvPutNumber3Zero (TOutIt *avOut,* int *avNumber*) [static],[protected],[noexcept],[inherited]

This function sends a 3-digit zero padded number to an output iterator.

Parameters

in	*avOut*	The output iterator.
in	*avNumber*	The 3-digit number.

Returns

The output iterator.

6.6.3.15 CFacetBase::TOutIt lfc1::datetime::CFacetBase::smvPutNumber4Zero (TOutIt *avOut,* int *avNumber*) [static],[protected],[noexcept],[inherited]

This function sends a 4-digit zero padded number to an output iterator.

Parameters

in	*avOut*	The output iterator.
in	*avNumber*	The 4-digit number.

Returns

The output iterator.

6.6.3.16 CDateBase::TOutIt lfc1::datetime::CDateBase::smvPutWeekDayName (TOutIt *avOut,* int *avWeekDayNo,* bool *avIsShort*) [static],[protected],[inherited]

This function sends the week day name of a given week day to an output iterator.

Parameters

in	*avOut*	The output iterator.
in	*avWeekDayNo*	The number of the week day.
in	*avIsShort*	Short (true) or long (false) week day name indicator.

Returns

The output iterator.

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

- include/lfc1/datetime/cdate.hpp
- datetime/library/src/cdateput.cpp

6.7 lfc1::datetime::CDateTime Class Reference

This class represents the time since 01-Jan-1970.

`#include <lfc1/datetime/cdatetime.hpp>`

Classes

- struct SDateTime

  A structure that holds the individual parts of a date and time.

Public Member Functions

- CDateTime () noexcept

  This function creates a default CDateTime object.
- CDateTime (long long avDateTimeNo)

  This function creates a CDateTime object using the given number of milliseconds since 01-Jan-1970 00:00:00.000.
- CDateTime (int avYear, int avMonth, int avDay, int avHour, int avMinute, int avSecond, int avMillisecond, int avDst=-1)

  This function creates a CDateTime object using the given year, month, day, hour, minute, second, millisecond and daylight savings indicator.
- CDateTime (const CDateTime &arRHS) noexcept

  This function copy constructs a CDateTime object.
- CDateTime (CDateTime &&arRHS) noexcept

  This function move constructs a CDateTime object.
- ~CDateTime () noexcept

  This function destroys a CDateTime object.
- CDateTime & operator= (long long avDateTimeNo)

This function assigns the given number of milliseconds since 01-Jan-1970 00:00:00.000 to a CDateTime object.

- CDateTime & operator= (const CDateTime &arRHS) noexcept

  This function assigns a CDateTime object to another CDateTime object.

- CDateTime & operator= (CDateTime &&arRHS) noexcept

  This function moves a CDateTime object to another CDateTime object.

- operator long long () const noexcept

  This function obtains the number of milliseconds since 01-Jan-1970 00:00:00.000.

- void mvSetDateTime () noexcept

  This function assigns the current system time to a CDateTime object.

- void mvSetDateTime (int avYear, int avMonth, int avDay, int avHour, int avMinute, int avSecond, int avMillisecond, int avDst=-1)

  This function sets a CDateTime object based on the given year, month, day, hour, minute, second, millisecond and daylight savings indicator.

- SDateTime mvGetDateTime () const noexcept

  This function obtains the equivalent local year, month, day, hour, minute, second, millisecond and daylight saving time indicator from a CDateTimeNo object.

- void swap (CDateTime &arRHS) noexcept

  This function swaps a CDateTime object with another CDateTime object.

- CDateTime & operator++ ()

  This function pre-increments a CDateTime object by one.

- CDateTime operator++ (int)

  This function post-increments a CDateTime object by one.

- CDateTime & operator-- ()

  This function pre-decrements a CDateTime object by one.

- CDateTime operator-- (int)

  This function post-decrements a CDateTime object by one.

- CDateTime & operator+= (const CDateTimeDuration &arDateTimeDuration)

  This function iIncrements the CDateTime object by the given duration.

- CDateTime & operator-= (const CDateTimeDuration &arDateTimeDuration)

  This function decrements the CDateTime object by the given duration.

- std::ios_base::iostate mvInput (std::istream &arStream)

  This function receives a CDateTime object from the given input stream.

- std::ios_base::iostate mvOutput (std::ostream &arStream) const

  This function sends a CDateTime object to the given output stream.

Static Public Member Functions

- static int smvGetIndex () noexcept

  This function obtains an iword/pword index.

- static void smvValidateDst (int avDst)

  This function validates a daylight savings indicator.

Static Public Attributes

- static const auto MIN_DATETIMENO = 0LL

  DateTimeNo for 01-Jan-1970 00:00:00.000 UTC.

- static const auto MAX_DATETIMENO = 32535215999999LL

  DateTimeNo for 31-Dec-3000 23:59:59.999 UTC.

6.7.1 Detailed Description

This class represents the time since 01-Jan-1970.

Note that this class has timezone dependency.

This class supports the following operators:

```
binary arithmetic operators: +=, -=
unary arithmetic operators:  ++, --
```

6.7.2 Constructor & Destructor Documentation

6.7.2.1 lfc1::datetime::CDateTime::CDateTime (long long *avDateTimeNo*)

This function creates a CDateTime object using the given number of milliseconds since 01-Jan-1970 00:00:00.000.

Parameters

in	*avDateTimeNo*	The number of milliseconds since 01-Jan-1970 00:00:00.000.

Exceptions

std::out_of_range	Indicates that the local date and time represented by the number of milliseconds since 01-Jan-1970 00:00:00.000 UTC is not within valid range.

6.7.2.2 lfc1::datetime::CDateTime::CDateTime (int *avYear,* int *avMonth,* int *avDay,* int *avHour,* int *avMinute,* int *avSecond,* int *avMillisecond,* int *avDst* = `-1`)

This function creates a CDateTime object using the given year, month, day, hour, minute, second, millisecond and daylight savings indicator.

Parameters

in	*avYear*	The given year.
in	*avMonth*	The given month.
in	*avDay*	The given day.
in	*avHour*	The given hour.
in	*avMinute*	The given minute.
in	*avSecond*	The given second.
in	*avMillisecond*	The given millisecond.
in	*avDst*	The given daylight savings indicator. (-1 means daylight state unknown, 0 means date and time is a standard date and time and 1 means date and time is a daylight date and time.

6.7.2.3 lfc1::datetime::CDateTime::CDateTime (const CDateTime & *arRHS*) `[noexcept]`

This function copy constructs a CDateTime object.

Parameters

in	*arRHS*	The CDateTime object to be copied.

6.7.2.4 lfc1::datetime::CDateTime::CDateTime (CDateTime && *arRHS*) [noexcept]

This function move constructs a CDateTime object.

Parameters

in	*arRHS*	The CDateTime object to be moved.

6.7.3 Member Function Documentation

6.7.3.1 CDateTime::SDateTime lfc1::datetime::CDateTime::mvGetDateTime ( ) const [noexcept]

This function obtains the equivalent local year, month, day, hour, minute, second, millisecond and daylight saving time indicator from a CDateTimeNo object.

Returns

The equivalent local year, month, day, hour, minute, second, millisecond and daylight saving time indicator from a CDateTimeNo object.

6.7.3.2 std::ios_base::iostate lfc1::datetime::CDateTime::mvInput (std::istream & *arStream*)

This function receives a CDateTime object from the given input stream.

Parameters

in	*arStream*	The source stream.

Returns

The resulting stream state.

6.7.3.3 std::ios_base::iostate lfc1::datetime::CDateTime::mvOutput (std::ostream & *arStream*) const

This function sends a CDateTime object to the given output stream.

Parameters

in	*arStream*	The destination stream.

Returns

The resulting stream state.

6.7.3.4 void lfc1::datetime::CDateTime::mvSetDateTime (int *avYear,* int *avMonth,* int *avDay,* int *avHour,* int *avMinute,* int *avSecond,* int *avMillisecond,* int *avDst* = −1)

This function sets a CDateTime object based on the given year, month, day, hour, minute, second, millisecond and daylight savings indicator.

Parameters

in	*avYear*	The given year.
in	*avMonth*	The given month.

in	*avDay*	The given day.
in	*avHour*	The given hour.
in	*avMinute*	The given minute.
in	*avSecond*	The given second.
in	*avMillisecond*	The given millisecond.
in	*avDst*	The given daylight savings indicator. (-1 means daylight state unknown, 0 means date and time is a standard date and time and 1 means date and time is a daylight date and time.

Exceptions

std::system_error	Indicates that conversion to a time_t value failed.
std::out_of_range	Indicates that the number of milliseconds since 01-Jan-1970 00:00:00.000 is out-of-range.

6.7.3.5 lfc1::datetime::CDateTime::operator long long ( ) const [noexcept]

This function obtains the number of milliseconds since 01-Jan-1970 00:00:00.000.

Returns

The number of milliseconds since 01-Jan-1970 00:00:00.000.

6.7.3.6 CDateTime & lfc1::datetime::CDateTime::operator++ ( )

This function pre-increments a CDateTime object by one.

Returns

The CDateTime object.

Exceptions

std::out_of_range	Indicates incrementing the number of milliseconds since 01-Jan-1970 00:00:00.-000 UTC will make the date go above the maximum supported date.

6.7.3.7 CDateTime lfc1::datetime::CDateTime::operator++ (int)

This function post-increments a CDateTime object by one.

Returns

The CDateTime object before the increment.

6.7.3.8 CDateTime & lfc1::datetime::CDateTime::operator+= (const CDateTimeDuration & *arDateTimeDuration*)

This function iIncrements the CDateTime object by the given duration.

Parameters

in	*arDateTime-Duration*	The duration to add to the date and time.

Returns

The CDateTime object.

Exceptions

std::out_of_range	Indicates incrementing the date and time will make the date and time go above the maximum supported date and time.

6.7.3.9 CDateTime & lfc1::datetime::CDateTime::operator-- ( )

This function pre-decrements a CDateTime object by one.

Returns

The CDateTime object.

Exceptions

std::out_of_range	Indicates decrementing the number of milliseconds since 01-Jan-1970 00:00:00.-000 UTC will make the date go below the minimum supported date.

6.7.3.10 CDateTime lfc1::datetime::CDateTime::operator-- (int)

This function post-decrements a CDateTime object by one.

Returns

The CDateTime object before the decrement.

6.7.3.11 CDateTime & lfc1::datetime::CDateTime::operator-= (const CDateTimeDuration & *arDateTimeDuration*)

This function decrements the CDateTime object by the given duration.

Parameters

in	*arDateTime-Duration*	The duration to subtract from the date and time.

Returns

The CDateTime object.

Exceptions

std::out_of_range	Indicates decrementing the date and time will make the date and time go below the minimum supported date and time.

6.7.3.12 CDateTime & lfc1::datetime::CDateTime::operator= (long long *avDateTimeNo*)

This function assigns the given number of milliseconds since 01-Jan-1970 00:00:00.000 to a CDateTime object.

Parameters

in	*avDateTimeNo*	The given number of milliseconds since 01-Jan-1970 00:00:00.000.

Returns

The CDateTime object assigned to.

6.7.3.13 CDateTime & lfc1::datetime::CDateTime::operator= (const CDateTime & *arRHS*) [noexcept]

This function assigns a CDateTime object to another CDateTime object.

Parameters

in	*arRHS*	The CDateTime object to assign to another CDateTime object.

Returns

The CDateTime object assigned to.

6.7.3.14 CDateTime & lfc1::datetime::CDateTime::operator= (CDateTime && *arRHS*) [noexcept]

This function moves a CDateTime object to another CDateTime object.

Parameters

in	*arRHS*	The CDateTime object to move to another CDateTime object.

Returns

The CDateTime object assigned to.

6.7.3.15 int lfc1::datetime::CDateTime::smvGetIndex ( ) [static], [noexcept]

This function obtains an iword/pword index.

Returns

The iword/pword index.

6.7.3.16 void lfc1::datetime::CDateTime::smvValidateDst (int *avDst*) [static]

This function validates a daylight savings indicator.

Parameters

in	*avDst*	The daylight savings indicator (-1, 0 or 1).

Exceptions

std::out_of_range	Indicates that the daylight savings indicator has an invalid value.

6.7.3.17 void lfc1::datetime::CDateTime::swap (CDateTime & *arRHS*) [noexcept]

This function swaps a CDateTime object with another CDateTime object.

Parameters

in,out	*arRHS*	The CDateTime object to swap with.

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

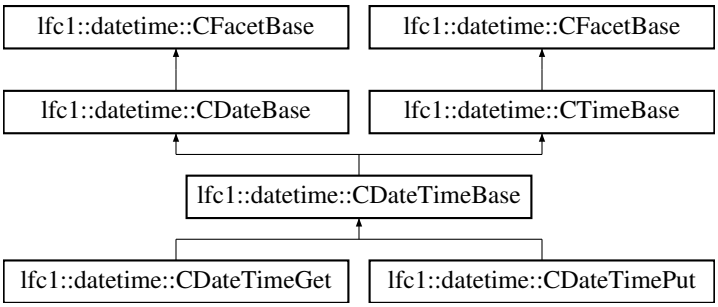
- include/lfc1/datetime/cdatetime.hpp
- datetime/library/src/cdatetime.cpp

6.8 lfc1::datetime::CDateTimeBase Class Reference

This class serves as the base class for date and time input/output facets.

`#include <lfc1/datetime/cdatetime.hpp>`

Inheritance diagram for lfc1::datetime::CDateTimeBase:



Public Types

- typedef
  std::istreambuf_iterator< char > TInIt

  This type is the input iterator.
- typedef
  std::ostreambuf_iterator< char > TOutIt

  This type is the output iterator.
- typedef
  std::istreambuf_iterator< char > TInIt

  This type is the input iterator.
- typedef
  std::ostreambuf_iterator< char > TOutIt

  This type is the output iterator.

Protected Member Functions

- CDateTimeBase () noexcept

  Creates a default CDateTimeBase object.
- CDateTimeBase (const CDateTimeBase &)=delete

  Not supported.
- CDateTimeBase (CDateTimeBase &&)=delete

Not supported.

- ~CDateTimeBase () noexceptoverride

  Destroys a CDateTimeBase object.

- CDateTimeBase & operator= (const CDateTimeBase &)=delete

  Not supported.

- CDateTimeBase & operator= (CDateTimeBase &&)=delete

  Not supported.

Static Protected Member Functions

- static std::string smvExpandFormat (std::string avFormat)

  This function expands the date and time format by replacing format specifiers with their equivalent format specifiers.

- static TInIt smvGetTzName (TInIt arNext, TInIt arEnd, std::ios_base::iostate &arIoState, int &arDst)

  This function parses an input iterator for a timezone name.

- static TInIt smvGetTzOffset (TInIt arNext, TInIt arEnd, std::ios_base &arIosBase, std::ios_base::iostate &arIoState, int &arDst)

  This function parses an input iterator for a timezone name.

- static TOutIt smvPutTzName (TOutIt avOut, int avDst)

  This function sends the timezone name to an output iterator.

- static TOutIt smvPutTzOffset (TOutIt avOut, int avDst) noexcept

  This function sends the timezone offset to an output iterator.

- static TInIt smvGetWeekDayName (TInIt avNext, TInIt avEnd, std::ios_base &arIosBase, std::ios_base::iostate &arIoState, int &arWeekDayNo, bool avIsShort)

  This function parses an input iterator for a week day name.

- static TInIt smvGetMonthName (TInIt avNext, TInIt avEnd, std::ios_base &arIosBase, std::ios_base::iostate &arIoState, int &arMonthNo, bool avIsShort)

  This function parses an input iterator for a month name.

- static TOutIt smvPutWeekDayName (TOutIt avOut, int avWeekDayNo, bool avIsShort)

  This function sends the week day name of a given week day to an output iterator.

- static TOutIt smvPutMonthName (TOutIt avOut, int avMonthNo, bool avIsShort)

  This function sends the month name of a given month to an output iterator.

- static void smvCalcWeekBasedDetails (const CDate &arDate, int &arWeekBasedYear, int *apWeekBasedWeekNo) noexcept

  This function determines the week-based year and week-based week number of a given date.

- static int smvCalcWeekNo (const CDate &arDate, int avYear, int avDay) noexcept

  This function determines the week number of a given date within a given year.

- static TInIt smvIgnoreChar (TInIt avNext, TInIt avEnd, std::ios_base::iostate &arIoState, char avExpectedChar) noexcept

  This function parses an input iterator for a specific character. This function moves the input iterator forward one character if the expected character is found.

- static TInIt smvGetNumber (TInIt avNext, TInIt avEnd, std::ios_base &arIosBase, std::ios_base::iostate &arIoState, int *apNumber, int avDigits, char avPadding) noexcept

  This function parses an input iterator for an integer value.

- static TOutIt smvPutNumber1 (TOutIt avOut, int avNumber) noexcept

  This function sends a 1-digit number to an output iterator.

- static TOutIt smvPutNumber2Zero (TOutIt avOut, int avNumber) noexcept

  This function sends a 2-digit zero padded number to an output iterator.

- static TOutIt smvPutNumber2Blank (TOutIt avOut, int avNumber) noexcept

  This function sends a 2-digit blank padded number to an output iterator.

- static TOutIt smvPutNumber3Zero (TOutIt avOut, int avNumber) noexcept

  This function sends a 3-digit zero padded number to an output iterator.

- static TOutIt smvPutNumber4Zero (TOutIt avOut, int avNumber) noexcept

  This function sends a 4-digit zero padded number to an output iterator.
- static TInIt smvGetAmPm (TInIt avNext, TInIt avEnd, std::ios_base &arIosBase, std::ios_base::iostate &arIoState, bool &arHasAmPm, bool &arIsPm)

  This function parses an input iterator for an AM/PM indicator.
- static TOutIt smvPutAmPm (TOutIt avOut, int avHour)

  This function sends AM or PM to an output iterator based on the given hour.
- static TInIt smvIgnoreChar (TInIt avNext, TInIt avEnd, std::ios_base::iostate &arIoState, char avExpectedChar) noexcept

  This function parses an input iterator for a specific character. This function moves the input iterator forward one character if the expected character is found.
- static TInIt smvGetNumber (TInIt avNext, TInIt avEnd, std::ios_base &arIosBase, std::ios_base::iostate &arIoState, int *apNumber, int avDigits, char avPadding) noexcept

  This function parses an input iterator for an integer value.
- static TOutIt smvPutNumber1 (TOutIt avOut, int avNumber) noexcept

  This function sends a 1-digit number to an output iterator.
- static TOutIt smvPutNumber2Zero (TOutIt avOut, int avNumber) noexcept

  This function sends a 2-digit zero padded number to an output iterator.
- static TOutIt smvPutNumber2Blank (TOutIt avOut, int avNumber) noexcept

  This function sends a 2-digit blank padded number to an output iterator.
- static TOutIt smvPutNumber3Zero (TOutIt avOut, int avNumber) noexcept

  This function sends a 3-digit zero padded number to an output iterator.
- static TOutIt smvPutNumber4Zero (TOutIt avOut, int avNumber) noexcept

  This function sends a 4-digit zero padded number to an output iterator.

6.8.1 Detailed Description

This class serves as the base class for date and time input/output facets.

6.8.2 Member Function Documentation

6.8.2.1 void lfc1::datetime::CDateBase::smvCalcWeekBasedDetails (const CDate & *arDate*, int & *arWeekBasedYear*, int * *apWeekBasedWeekNo*) [static], [protected], [noexcept], [inherited]

This function determines the week-based year and week-based week number of a given date.

Parameters

in	*arDate*	The given date expressed as a Julian day number.
in, out	*arWeekBasedYear*	Set to the starting week-based year on input and contains the resulting week-based year on output.
out	*apWeekBasedWeekNo*	The resulting week-based week number of the given date. If null, week-based week number is not stored.

6.8.2.2 int lfc1::datetime::CDateBase::smvCalcWeekNo (const CDate & *arDate*, int *avYear*, int *avDay*) [static], [protected], [noexcept], [inherited]

This function determines the week number of a given date within a given year.

Parameters

in	*arDate*	The given date expressed as a Julian day number.
in	*avYear*	The year of the given date.

in	*avDay*	The starting day of the week. (0 or 6 = Monday or Sunday)

Returns

The week number of a given date within a given year.

6.8.2.3 std::string lfc1::datetime::CDateTimeBase::smvExpandFormat (std::string *avFormat*) [static], [protected]

This function expands the date and time format by replacing format specifiers with their equivalent format specifiers.

Parameters

in	*avFormat*	The date and time format to be expanded.

Returns

The expanded format string.

6.8.2.4 CTimeBase::TInIt lfc1::datetime::CTimeBase::smvGetAmPm (TInIt *avNext,* TInIt *avEnd,* std::ios_base & *arIosBase,* std::ios_base::iostate & *arIoState,* bool & *arHasAmPm,* bool & *arIsPm*) [static], [protected], [inherited]

This function parses an input iterator for an AM/PM indicator.

Parameters

in	*avNext*	The input iterator.
in	*avEnd*	The input end iterator.
in	*arIosBase*	The stream formatting information.
out	*arIoState*	The stream state.
out	*arHasAmPm*	The storage for the resulting has AM/PM indicator (true) or not (false).
out	*arIsPm*	The storage for the resulting PM (true) or AM (false) indicator.

Returns

The input iterator.

6.8.2.5 CDateBase::TInIt lfc1::datetime::CDateBase::smvGetMonthName (TInIt *avNext,* TInIt *avEnd,* std::ios_base & *arIosBase,* std::ios_base::iostate & *arIoState,* int & *arMonthNo,* bool *avIsShort*) [static], [protected], [inherited]

This function parses an input iterator for a month name.

Parameters

in	*avNext*	The input iterator.
in	*avEnd*	The input end iterator.
in	*arIosBase*	The stream formatting information.
out	*arIoState*	The stream state.
out	*arMonthNo*	The storage for the resulting month number.
in	*avIsShort*	Short (true) or long (false) month name indicator.

Returns

The input iterator.

6.8.2.6 CFacetBase::TInIt lfc1::datetime::CFacetBase::smvGetNumber (TInIt *avNext*, TInIt *avEnd*, std::ios_base & *arIosBase*, std::ios_base::iostate & *arIoState*, int * *apNumber*, int *avDigits*, char *avPadding*) [static], [protected], [noexcept], [inherited]

This function parses an input iterator for an integer value.

Parameters

in	*avNext*	The input iterator.
in	*avEnd*	The input end iterator.
in	*arIosBase*	The stream formatting information.
out	*arIoState*	The stream state.
out	*apNumber*	The storage for the resulting integer.
in	*avDigits*	The maximum number of digits allowed for the integer.
in	*avPadding*	The padding character allowed for the integer value.

Return values

ios_base::goodbit	A valid integer was found.
ios_base::failbit	A valid integer was not found.

6.8.2.7 CFacetBase::TInIt lfc1::datetime::CFacetBase::smvGetNumber (TInIt *avNext*, TInIt *avEnd*, std::ios_base & *arIosBase*, std::ios_base::iostate & *arIoState*, int * *apNumber*, int *avDigits*, char *avPadding*) [static], [protected], [noexcept], [inherited]

This function parses an input iterator for an integer value.

Parameters

in	*avNext*	The input iterator.
in	*avEnd*	The input end iterator.
in	*arIosBase*	The stream formatting information.
out	*arIoState*	The stream state.
out	*apNumber*	The storage for the resulting integer.
in	*avDigits*	The maximum number of digits allowed for the integer.
in	*avPadding*	The padding character allowed for the integer value.

Return values

ios_base::goodbit	A valid integer was found.
ios_base::failbit	A valid integer was not found.

6.8.2.8 CDateTimeBase::TInIt lfc1::datetime::CDateTimeBase::smvGetTzName (TInIt *avNext*, TInIt *avEnd*, std::ios_base::iostate & *arIoState*, int & *arDst*) [static], [protected]

This function parses an input iterator for a timezone name.

Parameters

in	*avNext*	The input iterator.
in	*avEnd*	The input end iterator.

out	*arIoState*	The stream state.
out	*arDst*	The storage for the resulting daylight saving time indicator.

Returns

The input iterator.

6.8.2.9 CDateTimeBase::TInIt lfc1::datetime::CDateTimeBase::smvGetTzOffset (TInIt *avNext,* TInIt *avEnd,* std::ios_base & *arIosBase,* std::ios_base::iostate & *arIoState,* int & *arDst*) [static], [protected]

This function parses an input iterator for a timezone name.

Parameters

in	*avNext*	The input iterator.
in	*avEnd*	The input end iterator.
in	*arIosBase*	The stream formatting information.
out	*arIoState*	The stream state.
out	*arDst*	The storage for the resulting daylight saving time indicator.

Returns

The input iterator.

6.8.2.10 CDateBase::TInIt lfc1::datetime::CDateBase::smvGetWeekDayName (TInIt *avNext,* TInIt *avEnd,* std::ios_base & *arIosBase,* std::ios_base::iostate & *arIoState,* int & *arWeekDayNo,* bool *avIsShort*) [static], [protected], [inherited]

This function parses an input iterator for a week day name.

Parameters

in	*avNext*	The input iterator.
in	*avEnd*	The input end iterator.
in	*arIosBase*	The stream formatting information.
out	*arIoState*	The stream state.
out	*arWeekDayNo*	The storage for the resulting day of the week. (0 = Sunday)
in	*avIsShort*	Short (true) or long (false) week day name indicator.

Returns

The input iterator.

6.8.2.11 CFacetBase::TInIt lfc1::datetime::CFacetBase::smvIgnoreChar (TInIt *avNext,* TInIt *avEnd,* std::ios_base::iostate & *arIoState,* char *avExpectedChar*) [static], [protected], [noexcept], [inherited]

This function parses an input iterator for a specific character. This function moves the input iterator forward one character if the expected character is found.

Parameters

in	*avNext*	The input iterator.
in	*avEnd*	The input end iterator.

out	*arIoState*	The stream state.
in	*avExpectedChar*	The next character expected from the input iterator.

Return values

ios_base::goodbit	The expected character was found.
ios_base::failbit	The expected character was not found.

6.8.2.12 CFacetBase::TInIt lfc1::datetime::CFacetBase::smvIgnoreChar (TInIt *avNext,* TInIt *avEnd,* std::ios_base::iostate & *arIoState,* char *avExpectedChar*) [static], [protected], [noexcept], [inherited]

This function parses an input iterator for a specific character. This function moves the input iterator forward one character if the expected character is found.

Parameters

in	*avNext*	The input iterator.
in	*avEnd*	The input end iterator.
out	*arIoState*	The stream state.
in	*avExpectedChar*	The next character expected from the input iterator.

Return values

ios_base::goodbit	The expected character was found.
ios_base::failbit	The expected character was not found.

6.8.2.13 CTimeBase::TOutIt lfc1::datetime::CTimeBase::smvPutAmPm (TOutIt *avOut,* int *avHour*) [static], [protected], [inherited]

This function sends AM or PM to an output iterator based on the given hour.

Parameters

in	*avOut*	The output iterator.
in	*avHour*	The given hour.

Returns

The output iterator.

6.8.2.14 CDateBase::TOutIt lfc1::datetime::CDateBase::smvPutMonthName (TOutIt *avOut,* int *avMonthNo,* bool *avIsShort*) [static], [protected], [inherited]

This function sends the month name of a given month to an output iterator.

Parameters

in	*avOut*	The output iterator.
in	*avMonthNo*	The number of the month.
in	*avIsShort*	Short (true) or long (false) month name indicator.

Returns

The output iterator.

6.8.2.15 CFacetBase::TOutIt lfc1::datetime::CFacetBase::smvPutNumber1 (TOutIt *avOut,* int *avNumber*) [static], [protected], [noexcept], [inherited]

This function sends a 1-digit number to an output iterator.

Parameters

in	*avOut*	The output iterator.
in	*avNumber*	The 1-digit number.

Returns

The output iterator.

6.8.2.16 CFacetBase::TOutIt lfc1::datetime::CFacetBase::smvPutNumber1 (TOutIt *avOut,* int *avNumber*) [static], [protected], [noexcept], [inherited]

This function sends a 1-digit number to an output iterator.

Parameters

in	*avOut*	The output iterator.
in	*avNumber*	The 1-digit number.

Returns

The output iterator.

6.8.2.17 CFacetBase::TOutIt lfc1::datetime::CFacetBase::smvPutNumber2Blank (TOutIt *avOut,* int *avNumber*) [static], [protected], [noexcept], [inherited]

This function sends a 2-digit blank padded number to an output iterator.

Parameters

in	*avOut*	The output iterator.
in	*avNumber*	The 2-digit number.

Returns

The output iterator.

6.8.2.18 CFacetBase::TOutIt lfc1::datetime::CFacetBase::smvPutNumber2Blank (TOutIt *avOut,* int *avNumber*) [static], [protected], [noexcept], [inherited]

This function sends a 2-digit blank padded number to an output iterator.

Parameters

in	*avOut*	The output iterator.
in	*avNumber*	The 2-digit number.

Returns

The output iterator.

6.8.2.19 CFacetBase::TOutIt lfc1::datetime::CFacetBase::smvPutNumber2Zero (TOutIt *avOut,* int *avNumber*) [static],[protected],[noexcept],[inherited]

This function sends a 2-digit zero padded number to an output iterator.

Parameters

in	*avOut*	The output iterator.
in	*avNumber*	The 2-digit number.

Returns

The output iterator.

6.8.2.20 CFacetBase::TOutIt lfc1::datetime::CFacetBase::smvPutNumber2Zero (TOutIt *avOut,* int *avNumber*) [static],[protected],[noexcept],[inherited]

This function sends a 2-digit zero padded number to an output iterator.

Parameters

in	*avOut*	The output iterator.
in	*avNumber*	The 2-digit number.

Returns

The output iterator.

6.8.2.21 CFacetBase::TOutIt lfc1::datetime::CFacetBase::smvPutNumber3Zero (TOutIt *avOut,* int *avNumber*) [static],[protected],[noexcept],[inherited]

This function sends a 3-digit zero padded number to an output iterator.

Parameters

in	*avOut*	The output iterator.
in	*avNumber*	The 3-digit number.

Returns

The output iterator.

6.8.2.22 CFacetBase::TOutIt lfc1::datetime::CFacetBase::smvPutNumber3Zero (TOutIt *avOut,* int *avNumber*) [static],[protected],[noexcept],[inherited]

This function sends a 3-digit zero padded number to an output iterator.

Parameters

in	*avOut*	The output iterator.
in	*avNumber*	The 3-digit number.

Returns

The output iterator.

6.8.2.23 CFacetBase::TOutIt lfc1::datetime::CFacetBase::smvPutNumber4Zero (TOutIt *avOut,* int *avNumber*)

`[static],[protected],[noexcept],[inherited]`

This function sends a 4-digit zero padded number to an output iterator.

Parameters

in	*avOut*	The output iterator.
in	*avNumber*	The 4-digit number.

Returns

The output iterator.

6.8.2.24 CFacetBase::TOutIt lfc1::datetime::CFacetBase::smvPutNumber4Zero (TOutIt *avOut,* int *avNumber*)

`[static],[protected],[noexcept],[inherited]`

This function sends a 4-digit zero padded number to an output iterator.

Parameters

in	*avOut*	The output iterator.
in	*avNumber*	The 4-digit number.

Returns

The output iterator.

6.8.2.25 CDateTimeBase::TOutIt lfc1::datetime::CDateTimeBase::smvPutTzName (TOutIt *avOut,* int *avDst*)

`[static],[protected]`

This function sends the timezone name to an output iterator.

Parameters

in	*avOut*	The output iterator.
in	*avDst*	The daylight saving time indicator.

Returns

The output iterator.

6.8.2.26 CDateTimeBase::TOutIt lfc1::datetime::CDateTimeBase::smvPutTzOffset (TOutIt *avOut,* int *avDst*)

`[static],[protected],[noexcept]`

This function sends the timezone offset to an output iterator.

Parameters

in	*avOut*	The output iterator.
in	*avDst*	The daylight saving time indicator.

Returns

The output iterator.

6.8.2.27 CDateBase::TOutIt lfc1::datetime::CDateBase::smvPutWeekDayName (TOutIt *avOut*, int *avWeekDayNo*, bool *avIsShort*) [static],[protected],[inherited]

This function sends the week day name of a given week day to an output iterator.

Parameters

in	*avOut*	The output iterator.
in	*avWeekDayNo*	The number of the week day.
in	*avIsShort*	Short (true) or long (false) week day name indicator.

Returns

The output iterator.

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

- include/lfc1/datetime/cdatetime.hpp
- datetime/library/src/cdatetimebase.cpp

6.9 lfc1::datetime::CDateTimeDuration Class Reference

This class represents date and time durations.

`#include <lfc1/datetime/cdatetime.hpp>`

Public Types

- enum EUnits {
  E_U_YEAR,
  E_U_MONTH,
  E_U_WEEK,
  E_U_DAY,
  E_U_HOUR,
  E_U_MINUTE,
  E_U_SECOND,
  E_U_MILLISECOND }

  The list of date and time durations.

Public Member Functions

- CDateTimeDuration (int avDuration, EUnits avUnit=E_U_MILLISECOND) noexcept

  This function creates a CDateTimeDuration object.
- CDateTimeDuration (const CDateTimeDuration &arRHS) noexcept

  This function copy constructs a CDateTimeDuration object.
- CDateTimeDuration (CDateTimeDuration &&arRHS) noexcept

  This function move constructs a CDateTimeDuration object.
- ~CDateTimeDuration () noexcept

  This function destroys a CDateTimeDuration object.

- CDateTimeDuration & operator= (const CDateTimeDuration &arRHS) noexcept

  This function assigns a CDateTimeDuration object to another CDateTimeDuration object.
- CDateTimeDuration & operator= (CDateTimeDuration &&arRHS) noexcept

  This function moves a CDateTimeDuration object to another CDateTimeDuration object.
- int mvGetDuration () const noexcept

  This function obtains the date duration.
- EUnits mvGetUnit () const noexcept

  This function obtains the unit of the date duration.
- void swap (CDateTimeDuration &arRHS) noexcept

  This function swaps a CDateTimeDuration object with another CDateTimeDuration object.

6.9.1 Detailed Description

This class represents date and time durations.

The purpose of this class is to support the increment and decrement operators of the CDateTime class.

6.9.2 Constructor & Destructor Documentation

6.9.2.1 lfc1::datetime::CDateTimeDuration::CDateTimeDuration (int *avDuration*, EUnits *avUnit* = E_U_MILLISECOND) [noexcept]

This function creates a CDateTimeDuration object.

Parameters

in	*avDuration*	The date duration.
in	*avUnit*	The unit of measure of the date duration.

6.9.2.2 lfc1::datetime::CDateTimeDuration::CDateTimeDuration (const CDateTimeDuration & *arRHS*) [noexcept]

This function copy constructs a CDateTimeDuration object.

Parameters

in	*arRHS*	The CDateTimeDuration object to be copied.

6.9.2.3 lfc1::datetime::CDateTimeDuration::CDateTimeDuration (CDateTimeDuration && *arRHS*) [noexcept]

This function move constructs a CDateTimeDuration object.

Parameters

in	*arRHS*	The CDateTimeDuration object to be moved.

6.9.3 Member Function Documentation

6.9.3.1 int lfc1::datetime::CDateTimeDuration::mvGetDuration ( ) const [noexcept]

This function obtains the date duration.

Returns

The date duration.

6.9.3.2 CDateTimeDuration::EUnits lfc1::datetime::CDateTimeDuration::mvGetUnit ( ) const [noexcept]

This function obtains the unit of the date duration.

Returns

The unit of the date duration.

6.9.3.3 CDateTimeDuration & lfc1::datetime::CDateTimeDuration::operator= (const CDateTimeDuration & *arRHS*) [noexcept]

This function assigns a CDateTimeDuration object to another CDateTimeDuration object.

Parameters

in	*arRHS*	The CDateTimeDuration object to assign to another CDateTimeDuration object.

Returns

The CDateTimeDuration object assigned to.

6.9.3.4 CDateTimeDuration & lfc1::datetime::CDateTimeDuration::operator= (CDateTimeDuration && *arRHS*) [noexcept]

This function moves a CDateTimeDuration object to another CDateTimeDuration object.

Parameters

in	*arRHS*	The CDateTimeDuration object to move to another CDateTimeDuration object.

Returns

The CDateTimeDuration object assigned to.

6.9.3.5 void lfc1::datetime::CDateTimeDuration::swap (CDateTimeDuration & *arRHS*) [noexcept]

This function swaps a CDateTimeDuration object with another CDateTimeDuration object.

Parameters

in,out	*arRHS*	The CDateTimeDuration object to swap with.

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

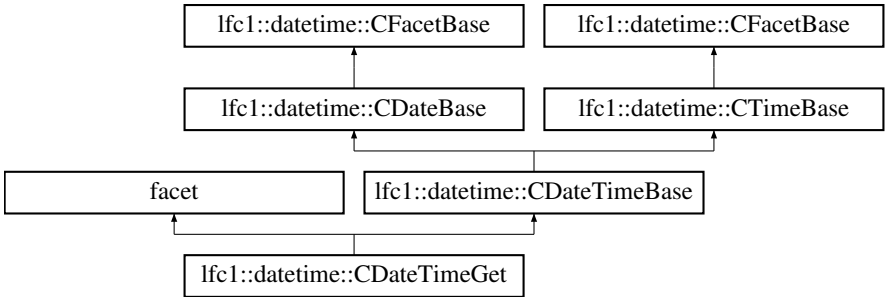
- include/lfc1/datetime/cdatetime.hpp
- datetime/library/src/cdatetimeduration.cpp

6.10 lfc1::datetime::CDateTimeGet Class Reference

This class is a CDateTime class input facet.

`#include <lfc1/datetime/cdatetime.hpp>`

Inheritance diagram for lfc1::datetime::CDateTimeGet:



Public Types

- typedef
  std::istreambuf_iterator< char > TInIt

  This type is the input iterator.
- typedef
  std::istreambuf_iterator< char > TInIt

  This type is the input iterator.
- typedef
  std::ostreambuf_iterator< char > TOutIt

  This type is the output iterator.
- typedef
  std::ostreambuf_iterator< char > TOutIt

  This type is the output iterator.

Public Member Functions

- CDateTimeGet (size_t avRelease=0)

  This function creates a CDateTimeGet object.
- CDateTimeGet (const CDateTimeGet &)=delete

  Not supported.
- CDateTimeGet (CDateTimeGet &&)=delete

  Not supported.
- ~CDateTimeGet () noexceptfinal

  This function destroys a CDateTimeGet object.
- CDateTimeGet & operator= (const CDateTimeGet &)=delete

  Not supported.
- CDateTimeGet & operator= (CDateTimeGet &&)=delete

  Not supported.
- TInIt mvGet (TInIt avNext, TInIt avEnd, std::ios_base &arIosBase, std::ios_base::iostate &arIoState, CDateTime &arDateTime, const std::string &arFormat) const

  This function inputs a date and time from a stream. Duplication of date or time parts is not allowed.

Static Public Attributes

- static std::locale::id id

  The facet ID.

Protected Member Functions

- virtual TInIt mvDoGet (TInIt avNext, TInIt avEnd, std::ios_base &arIosBase, std::ios_base::iostate &arIoState, CDateTime &arDateTime, const std::string &arFormat) const

  This function implements the behavior of the mvGet() function.

Static Protected Member Functions

- static std::string smvExpandFormat (std::string avFormat)

  This function expands the date and time format by replacing format specifiers with their equivalent format specifiers.
- static TInIt smvGetTzName (TInIt arNext, TInIt arEnd, std::ios_base::iostate &arIoState, int &arDst)

  This function parses an input iterator for a timezone name.
- static TInIt smvGetTzOffset (TInIt arNext, TInIt arEnd, std::ios_base &arIosBase, std::ios_base::iostate &arIoState, int &arDst)

  This function parses an input iterator for a timezone name.
- static TOutIt smvPutTzName (TOutIt avOut, int avDst)

  This function sends the timezone name to an output iterator.
- static TOutIt smvPutTzOffset (TOutIt avOut, int avDst) noexcept

  This function sends the timezone offset to an output iterator.
- static TInIt smvGetWeekDayName (TInIt avNext, TInIt avEnd, std::ios_base &arIosBase, std::ios_base::iostate &arIoState, int &arWeekDayNo, bool avIsShort)

  This function parses an input iterator for a week day name.
- static TInIt smvGetMonthName (TInIt avNext, TInIt avEnd, std::ios_base &arIosBase, std::ios_base::iostate &arIoState, int &arMonthNo, bool avIsShort)

  This function parses an input iterator for a month name.
- static TOutIt smvPutWeekDayName (TOutIt avOut, int avWeekDayNo, bool avIsShort)

  This function sends the week day name of a given week day to an output iterator.
- static TOutIt smvPutMonthName (TOutIt avOut, int avMonthNo, bool avIsShort)

  This function sends the month name of a given month to an output iterator.
- static void smvCalcWeekBasedDetails (const CDate &arDate, int &arWeekBasedYear, int *apWeekBasedWeekNo) noexcept

  This function determines the week-based year and week-based week number of a given date.
- static int smvCalcWeekNo (const CDate &arDate, int avYear, int avDay) noexcept

  This function determines the week number of a given date within a given year.
- static TInIt smvIgnoreChar (TInIt avNext, TInIt avEnd, std::ios_base::iostate &arIoState, char avExpectedChar) noexcept

  This function parses an input iterator for a specific character. This function moves the input iterator forward one character if the expected character is found.
- static TInIt smvIgnoreChar (TInIt avNext, TInIt avEnd, std::ios_base::iostate &arIoState, char avExpectedChar) noexcept

  This function parses an input iterator for a specific character. This function moves the input iterator forward one character if the expected character is found.
- static TInIt smvGetNumber (TInIt avNext, TInIt avEnd, std::ios_base &arIosBase, std::ios_base::iostate &arIoState, int *apNumber, int avDigits, char avPadding) noexcept

  This function parses an input iterator for an integer value.
- static TInIt smvGetNumber (TInIt avNext, TInIt avEnd, std::ios_base &arIosBase, std::ios_base::iostate &arIoState, int *apNumber, int avDigits, char avPadding) noexcept

This function parses an input iterator for an integer value.

- static TOutIt smvPutNumber1 (TOutIt avOut, int avNumber) noexcept

  This function sends a 1-digit number to an output iterator.
- static TOutIt smvPutNumber1 (TOutIt avOut, int avNumber) noexcept

  This function sends a 1-digit number to an output iterator.
- static TOutIt smvPutNumber2Zero (TOutIt avOut, int avNumber) noexcept

  This function sends a 2-digit zero padded number to an output iterator.
- static TOutIt smvPutNumber2Zero (TOutIt avOut, int avNumber) noexcept

  This function sends a 2-digit zero padded number to an output iterator.
- static TOutIt smvPutNumber2Blank (TOutIt avOut, int avNumber) noexcept

  This function sends a 2-digit blank padded number to an output iterator.
- static TOutIt smvPutNumber2Blank (TOutIt avOut, int avNumber) noexcept

  This function sends a 2-digit blank padded number to an output iterator.
- static TOutIt smvPutNumber3Zero (TOutIt avOut, int avNumber) noexcept

  This function sends a 3-digit zero padded number to an output iterator.
- static TOutIt smvPutNumber3Zero (TOutIt avOut, int avNumber) noexcept

  This function sends a 3-digit zero padded number to an output iterator.
- static TOutIt smvPutNumber4Zero (TOutIt avOut, int avNumber) noexcept

  This function sends a 4-digit zero padded number to an output iterator.
- static TOutIt smvPutNumber4Zero (TOutIt avOut, int avNumber) noexcept

  This function sends a 4-digit zero padded number to an output iterator.
- static TInIt smvGetAmPm (TInIt avNext, TInIt avEnd, std::ios_base &arIosBase, std::ios_base::iostate &arIoState, bool &arHasAmPm, bool &arIsPm)

  This function parses an input iterator for an AM/PM indicator.
- static TOutIt smvPutAmPm (TOutIt avOut, int avHour)

  This function sends AM or PM to an output iterator based on the given hour.

6.10.1 Detailed Description

This class is a CDateTime class input facet.

Note

This class uses the Template Method design pattern.

6.10.2 Constructor & Destructor Documentation

6.10.2.1 lfc1::datetime::CDateTimeGet::CDateTimeGet (size_t *avRelease* = `0`) `[explicit]`

This function creates a CDateTimeGet object.

Parameters

in	*avRelease*	Indicates who controls the lifetime of the facet. (0 means locale)

6.10.3 Member Function Documentation

6.10.3.1 CDateTimeGet::TInIt lfc1::datetime::CDateTimeGet::mvDoGet (TInIt *avNext*, TInIt *avEnd*, std::ios_base & *arIosBase*, std::ios_base::iostate & *arIoState*, CDateTime & *arDateTime*, const std::string & *arFormat*) const `[protected], [virtual]`

This function implements the behavior of the mvGet() function.

Parameters

in	*avNext*	The input iterator.
in	*avEnd*	The input end iterator.
in	*arIosBase*	The stream formatting information.
out	*arIoState*	The stream state.
out	*arDateTime*	The date and time retrieved from the input stream.
in	*arFormat*	The format of the date and time retrieved from the input stream.

Returns

The input iterator.

6.10.3.2 CDateTimeGet::TInIt lfc1::datetime::CDateTimeGet::mvGet (TInIt *avNext*, TInIt *avEnd*, std::ios_base & *arIosBase*, std::ios_base::iostate & *arIoState*, CDateTime & *arDateTime*, const std::string & *arFormat*) const

This function inputs a date and time from a stream. Duplication of date or time parts is not allowed.

This function recognizes the following format specifiers which are identical to the time specific format specifiers of the C language strftime() function except for N:

```
%% - A percent character.
%a - Abbreviated weekday name.
%A - Full weekday name.
%b - Abbreviated month name.
%B - Full month name.
%c - Default date and time format.
%C - Century [00-99].
%d - Day of the month zero padded [01-31].
%D - Equivalent to %m/%d/%y.
%e - Day of the month blank padded [ 1-31].
%F - Equivalent to %Y-%m-%d.
%g - Week-based year without century [00-99].
%G - Week-based year with century [0000-9999].
%h - Equivalent to %b.
%H - Hour in 24-hour format [00-23] zero padded.
%I - Hour in 12-hour format [01-12] zero padded.
%j - Day of the year [001-366].
%k - Hour in 24-hour format [00-23] blank padded.
%l - Hour in 12-hour format [01-12] blank padded.
%m - Month number [01-12].
%M - Minute [00-59].
%n - Newline character.
%N - Millisecond [000-999].
%p - AM/PM designation.
%r - Default 12-hour time format.
%R - Equivalent to %H:%M.
%S - Second [00-59].
%t - Horizontal tab character.
%T - Equivalent to %H:%M:%S.
%u - Day of the week [1-7], Monday is 1.
%U - Week of the year [00-53], the first Sunday starts week 1.
%V - Week number of a week-based year [01-53].
%w - Day of the week [0-6], Sunday is 0.
%W - Week of the year [00-53], the first Monday starts week 1.
%x - Default date format.
%X - Default time format.
%y - Year without century [00-99].
%Y - Year with century [0000-9999].
%z - Time zone offset.  (Accepted but not used.)
%Z - Time zone name.    (Accepted but not used.)
```

Any other sequence of characters not included in the above list will be taken literally.

The following combinations of format specifiers results into a valid date for a CDateTime object:

```
%D
%F
%x
(%Y or (%y [and %C])) and (%b or %B or %m) and (%d or %e)
(%Y or (%y [and %C])) and (%j)
(%Y or (%y [and %C])) and (%U or %W) and (%a or %A or %u or %w)
(%G or (%g [and %C])) and (%V) and (%a or %A or %u or %w)
```

The following combinations of format specifiers results into a valid time for a CDateTime object:

```
[%R]
[%T]
[%X]
[(%H or %I or %k or %l) [and (%M) [and (%S) [and (%N) [and (%z or %Z)]]]]]
```

Parameters

in	*avNext*	The input iterator.
in	*avEnd*	The input end iterator.
in	*arIosBase*	The stream formatting information.
out	*arIoState*	The stream state.
out	*arDateTime*	The date and time retrieved from the input stream.
in	*arFormat*	The format of the date and time retrieved from the input stream.

Returns

The input iterator.

6.10.3.3 void lfc1::datetime::CDateBase::smvCalcWeekBasedDetails (const CDate & *arDate*, int & *arWeekBasedYear*, int * *apWeekBasedWeekNo*) [static], [protected], [noexcept], [inherited]

This function determines the week-based year and week-based week number of a given date.

Parameters

in	*arDate*	The given date expressed as a Julian day number.
in,out	*arWeekBased-Year*	Set to the starting week-based year on input and contains the resulting week-based year on output.
out	*apWeekBased-WeekNo*	The resulting week-based week number of the given date. If null, week-based week number is not stored.

6.10.3.4 int lfc1::datetime::CDateBase::smvCalcWeekNo (const CDate & *arDate*, int *avYear*, int *avDay*) [static], [protected], [noexcept], [inherited]

This function determines the week number of a given date within a given year.

Parameters

in	*arDate*	The given date expressed as a Julian day number.
in	*avYear*	The year of the given date.
in	*avDay*	The starting day of the week. (0 or 6 = Monday or Sunday)

Returns

The week number of a given date within a given year.

6.10.3.5 std::string lfc1::datetime::CDateTimeBase::smvExpandFormat (std::string *avFormat*) [static], [protected], [inherited]

This function expands the date and time format by replacing format specifiers with their equivalent format specifiers.

Parameters

in	*avFormat*	The date and time format to be expanded.

Returns

The expanded format string.

6.10.3.6 CTimeBase::TInIt lfc1::datetime::CTimeBase::smvGetAmPm (TInIt *avNext,* TInIt *avEnd,* std::ios_base & *arIosBase,* std::ios_base::iostate & *arIoState,* bool & *arHasAmPm,* bool & *arIsPm*) [static], [protected], [inherited]

This function parses an input iterator for an AM/PM indicator.

Parameters

in	*avNext*	The input iterator.
in	*avEnd*	The input end iterator.
in	*arIosBase*	The stream formatting information.
out	*arIoState*	The stream state.
out	*arHasAmPm*	The storage for the resulting has AM/PM indicator (true) or not (false).
out	*arIsPm*	The storage for the resulting PM (true) or AM (false) indicator.

Returns

The input iterator.

6.10.3.7 CDateBase::TInIt lfc1::datetime::CDateBase::smvGetMonthName (TInIt *avNext,* TInIt *avEnd,* std::ios_base & *arIosBase,* std::ios_base::iostate & *arIoState,* int & *arMonthNo,* bool *avIsShort*) [static], [protected], [inherited]

This function parses an input iterator for a month name.

Parameters

in	*avNext*	The input iterator.
in	*avEnd*	The input end iterator.
in	*arIosBase*	The stream formatting information.
out	*arIoState*	The stream state.
out	*arMonthNo*	The storage for the resulting month number.
in	*avIsShort*	Short (true) or long (false) month name indicator.

Returns

The input iterator.

6.10.3.8 CFacetBase::TInIt lfc1::datetime::CFacetBase::smvGetNumber (TInIt *avNext,* TInIt *avEnd,* std::ios_base & *arIosBase,* std::ios_base::iostate & *arIoState,* int * *apNumber,* int *avDigits,* char *avPadding*) [static], [protected], [noexcept], [inherited]

This function parses an input iterator for an integer value.

Parameters

in	*avNext*	The input iterator.
in	*avEnd*	The input end iterator.
in	*arIosBase*	The stream formatting information.
out	*arIoState*	The stream state.
out	*apNumber*	The storage for the resulting integer.
in	*avDigits*	The maximum number of digits allowed for the integer.
in	*avPadding*	The padding character allowed for the integer value.

Return values

ios_base::goodbit	A valid integer was found.
ios_base::failbit	A valid integer was not found.

6.10.3.9 CFacetBase::TInIt lfc1::datetime::CFacetBase::smvGetNumber (TInIt *avNext,* TInIt *avEnd,* std::ios_base & *arIosBase,* std::ios_base::iostate & *arIoState,* int * *apNumber,* int *avDigits,* char *avPadding*) [static], [protected], [noexcept], [inherited]

This function parses an input iterator for an integer value.

Parameters

in	*avNext*	The input iterator.
in	*avEnd*	The input end iterator.
in	*arIosBase*	The stream formatting information.
out	*arIoState*	The stream state.
out	*apNumber*	The storage for the resulting integer.
in	*avDigits*	The maximum number of digits allowed for the integer.
in	*avPadding*	The padding character allowed for the integer value.

Return values

ios_base::goodbit	A valid integer was found.
ios_base::failbit	A valid integer was not found.

6.10.3.10 CDateTimeBase::TInIt lfc1::datetime::CDateTimeBase::smvGetTzName (TInIt *avNext,* TInIt *avEnd,* std::ios_base::iostate & *arIoState,* int & *arDst*) [static], [protected], [inherited]

This function parses an input iterator for a timezone name.

Parameters

in	*avNext*	The input iterator.
in	*avEnd*	The input end iterator.
out	*arIoState*	The stream state.
out	*arDst*	The storage for the resulting daylight saving time indicator.

Returns

The input iterator.

6.10.3.11 CDateTimeBase::TInIt lfc1::datetime::CDateTimeBase::smvGetTzOffset (TInIt *avNext,* TInIt *avEnd,* std::ios_base & *arIosBase,* std::ios_base::iostate & *arIoState,* int & *arDst*) [static],[protected],[inherited]

This function parses an input iterator for a timezone name.

Parameters

in	*avNext*	The input iterator.
in	*avEnd*	The input end iterator.
in	*arIosBase*	The stream formatting information.
out	*arIoState*	The stream state.
out	*arDst*	The storage for the resulting daylight saving time indicator.

Returns

The input iterator.

6.10.3.12 CDateBase::TInIt lfc1::datetime::CDateBase::smvGetWeekDayName (TInIt *avNext,* TInIt *avEnd,* std::ios_base & *arIosBase,* std::ios_base::iostate & *arIoState,* int & *arWeekDayNo,* bool *avIsShort*) [static],[protected],[inherited]

This function parses an input iterator for a week day name.

Parameters

in	*avNext*	The input iterator.
in	*avEnd*	The input end iterator.
in	*arIosBase*	The stream formatting information.
out	*arIoState*	The stream state.
out	*arWeekDayNo*	The storage for the resulting day of the week. (0 = Sunday)
in	*avIsShort*	Short (true) or long (false) week day name indicator.

Returns

The input iterator.

6.10.3.13 CFacetBase::TInIt lfc1::datetime::CFacetBase::smvIgnoreChar (TInIt *avNext,* TInIt *avEnd,* std::ios_base::iostate & *arIoState,* char *avExpectedChar*) [static],[protected],[noexcept],[inherited]

This function parses an input iterator for a specific character. This function moves the input iterator forward one character if the expected character is found.

Parameters

in	*avNext*	The input iterator.
in	*avEnd*	The input end iterator.
out	*arIoState*	The stream state.
in	*avExpectedChar*	The next character expected from the input iterator.

Return values

ios_base::goodbit	The expected character was found.
ios_base::failbit	The expected character was not found.

6.10.3.14 CFacetBase::TInIt lfc1::datetime::CFacetBase::smvIgnoreChar (TInIt *avNext,* TInIt *avEnd,* std::ios_base::iostate & *arIoState,* char *avExpectedChar*) [static],[protected],[noexcept],[inherited]

This function parses an input iterator for a specific character. This function moves the input iterator forward one character if the expected character is found.

Parameters

in	*avNext*	The input iterator.
in	*avEnd*	The input end iterator.
out	*arIoState*	The stream state.
in	*avExpectedChar*	The next character expected from the input iterator.

Return values

ios_base::goodbit	The expected character was found.
ios_base::failbit	The expected character was not found.

6.10.3.15 CTimeBase::TOutIt lfc1::datetime::CTimeBase::smvPutAmPm (TOutIt *avOut,* int *avHour*) [static], [protected],[inherited]

This function sends AM or PM to an output iterator based on the given hour.

Parameters

in	*avOut*	The output iterator.
in	*avHour*	The given hour.

Returns

The output iterator.

6.10.3.16 CDateBase::TOutIt lfc1::datetime::CDateBase::smvPutMonthName (TOutIt *avOut,* int *avMonthNo,* bool *avIsShort*) [static],[protected],[inherited]

This function sends the month name of a given month to an output iterator.

Parameters

in	*avOut*	The output iterator.
in	*avMonthNo*	The number of the month.
in	*avIsShort*	Short (true) or long (false) month name indicator.

Returns

The output iterator.

6.10.3.17 CFacetBase::TOutIt lfc1::datetime::CFacetBase::smvPutNumber1 (TOutIt *avOut,* int *avNumber*)

[static],[protected],[noexcept],[inherited]

This function sends a 1-digit number to an output iterator.

Parameters

in	*avOut*	The output iterator.
in	*avNumber*	The 1-digit number.

Returns

The output iterator.

6.10.3.18 CFacetBase::TOutIt lfc1::datetime::CFacetBase::smvPutNumber1 (TOutIt *avOut,* int *avNumber*)

[static],[protected],[noexcept],[inherited]

This function sends a 1-digit number to an output iterator.

Parameters

in	*avOut*	The output iterator.
in	*avNumber*	The 1-digit number.

Returns

The output iterator.

6.10.3.19 CFacetBase::TOutIt lfc1::datetime::CFacetBase::smvPutNumber2Blank (TOutIt *avOut,* int *avNumber*)

[static],[protected],[noexcept],[inherited]

This function sends a 2-digit blank padded number to an output iterator.

Parameters

in	*avOut*	The output iterator.
in	*avNumber*	The 2-digit number.

Returns

The output iterator.

6.10.3.20 CFacetBase::TOutIt lfc1::datetime::CFacetBase::smvPutNumber2Blank (TOutIt *avOut,* int *avNumber*)

[static],[protected],[noexcept],[inherited]

This function sends a 2-digit blank padded number to an output iterator.

Parameters

in	*avOut*	The output iterator.
in	*avNumber*	The 2-digit number.

Returns

The output iterator.

6.10.3.21 CFacetBase::TOutIt lfc1::datetime::CFacetBase::smvPutNumber2Zero (TOutIt *avOut*, int *avNumber*) [static], [protected], [noexcept], [inherited]

This function sends a 2-digit zero padded number to an output iterator.

Parameters

in	*avOut*	The output iterator.
in	*avNumber*	The 2-digit number.

Returns

The output iterator.

6.10.3.22 CFacetBase::TOutIt lfc1::datetime::CFacetBase::smvPutNumber2Zero (TOutIt *avOut*, int *avNumber*) [static], [protected], [noexcept], [inherited]

This function sends a 2-digit zero padded number to an output iterator.

Parameters

in	*avOut*	The output iterator.
in	*avNumber*	The 2-digit number.

Returns

The output iterator.

6.10.3.23 CFacetBase::TOutIt lfc1::datetime::CFacetBase::smvPutNumber3Zero (TOutIt *avOut*, int *avNumber*) [static], [protected], [noexcept], [inherited]

This function sends a 3-digit zero padded number to an output iterator.

Parameters

in	*avOut*	The output iterator.
in	*avNumber*	The 3-digit number.

Returns

The output iterator.

6.10.3.24 CFacetBase::TOutIt lfc1::datetime::CFacetBase::smvPutNumber3Zero (TOutIt *avOut*, int *avNumber*) [static], [protected], [noexcept], [inherited]

This function sends a 3-digit zero padded number to an output iterator.

Parameters

in	*avOut*	The output iterator.
in	*avNumber*	The 3-digit number.

Returns

The output iterator.

6.10.3.25 CFacetBase::TOutIt lfc1::datetime::CFacetBase::smvPutNumber4Zero (TOutIt *avOut,* int *avNumber*) [static],[protected],[noexcept],[inherited]

This function sends a 4-digit zero padded number to an output iterator.

Parameters

in	*avOut*	The output iterator.
in	*avNumber*	The 4-digit number.

Returns

The output iterator.

6.10.3.26 CFacetBase::TOutIt lfc1::datetime::CFacetBase::smvPutNumber4Zero (TOutIt *avOut,* int *avNumber*) [static],[protected],[noexcept],[inherited]

This function sends a 4-digit zero padded number to an output iterator.

Parameters

in	*avOut*	The output iterator.
in	*avNumber*	The 4-digit number.

Returns

The output iterator.

6.10.3.27 CDateTimeBase::TOutIt lfc1::datetime::CDateTimeBase::smvPutTzName (TOutIt *avOut,* int *avDst*) [static],[protected],[inherited]

This function sends the timezone name to an output iterator.

Parameters

in	*avOut*	The output iterator.
in	*avDst*	The daylight saving time indicator.

Returns

The output iterator.

6.10.3.28 CDateTimeBase::TOutIt lfc1::datetime::CDateTimeBase::smvPutTzOffset (TOutIt *avOut,* int *avDst*) [static],[protected],[noexcept],[inherited]

This function sends the timezone offset to an output iterator.

Parameters

in	*avOut*	The output iterator.
in	*avDst*	The daylight saving time indicator.

Returns

The output iterator.

6.10.3.29 CDateBase::TOutIt lfc1::datetime::CDateBase::smvPutWeekDayName (TOutIt *avOut,* int *avWeekDayNo,* bool *avIsShort*) [static],[protected],[inherited]

This function sends the week day name of a given week day to an output iterator.

Parameters

in	*avOut*	The output iterator.
in	*avWeekDayNo*	The number of the week day.
in	*avIsShort*	Short (true) or long (false) week day name indicator.

Returns

The output iterator.

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

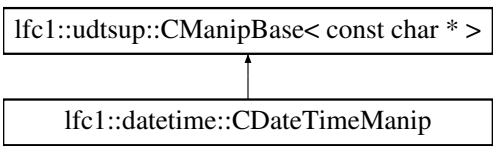
- include/lfc1/datetime/cdatetime.hpp
- datetime/library/src/cdatetimeget.cpp

6.11 lfc1::datetime::CDateTimeManip Class Reference

This class is a helper class for the date and time manipulator. This class was derived from Section 3.3.1.3 to 3.3.1.7 of the book Standard C++ IOStreams and Locales.

`#include <lfc1/datetime/cdatetime.hpp>`

Inheritance diagram for lfc1::datetime::CDateTimeManip:



Public Member Functions

- CDateTimeManip (const char *apFormat)

  This function creates a CDateTimeManip object.
- CDateTimeManip (const CDateTimeManip &)=default

  Uses default implementation.
- CDateTimeManip & operator= (const CDateTimeManip &)=default

  Uses default implementation.

Static Public Member Functions

- static std::string smvGetFormat (std::ios_base &arIosBase)

  This function obtains the date and time format from a stream.
- static std::ios_base::iostate smvCopyFmtErr (std::basic_ios< char > &arIos)

  This function obtains the error information relating to date and time format copying.

6.11.1 Detailed Description

This class is a helper class for the date and time manipulator. This class was derived from Section 3.3.1.3 to 3.3.1.7 of the book Standard C++ IOStreams and Locales.

6.11.2 Constructor & Destructor Documentation

6.11.2.1 lfc1::datetime::CDateTimeManip::CDateTimeManip (const char * *apFormat*)

This function creates a CDateTimeManip object.

Parameters

in	*apFormat*	The desired date and time format.

6.11.3 Member Function Documentation

6.11.3.1 std::ios_base::iostate lfc1::datetime::CDateTimeManip::smvCopyFmtErr (std::basic_ios< char > & *arlos*) [static]

This function obtains the error information relating to date and time format copying.

Parameters

in	*arlos*	The stream containing the error information.

Returns

The error information relating to date and time format copying.

6.11.3.2 std::string lfc1::datetime::CDateTimeManip::smvGetFormat (std::ios_base & *arlosBase*) [static]

This function obtains the date and time format from a stream.

Parameters

in	*arlosBase*	The stream containing the date and time format.

Returns

The date and time format.

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

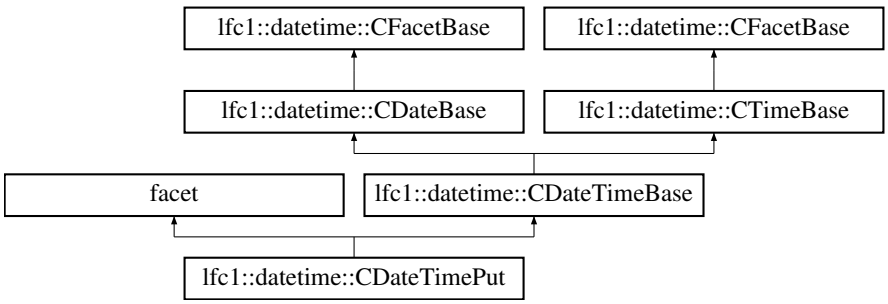
- include/lfc1/datetime/cdatetime.hpp
- datetime/library/src/cdatetimemanip.cpp

6.12 lfc1::datetime::CDateTimePut Class Reference

This class is a CDateTime class output facet.

`#include <lfc1/datetime/cdatetime.hpp>`

Inheritance diagram for lfc1::datetime::CDateTimePut:



Public Types

- typedef
  std::istreambuf_iterator< char > TInIt
  This type is the input iterator.
- typedef
  std::istreambuf_iterator< char > TInIt
  This type is the input iterator.
- typedef
  std::ostreambuf_iterator< char > TOutIt
  This type is the output iterator.
- typedef
  std::ostreambuf_iterator< char > TOutIt
  This type is the output iterator.

Public Member Functions

- CDateTimePut (size_t avRelease=0)
  This function creates a CDateTimePut object.
- CDateTimePut (const CDateTimePut &)=delete
  Not supported.
- CDateTimePut (CDateTimePut &&)=delete
  Not supported.
- ~CDateTimePut () noexceptfinal
  This function destroys a CDateTimePut object.
- CDateTimePut & operator= (const CDateTimePut &)=delete
  Not supported.
- CDateTimePut & operator= (CDateTimePut &&)=delete
  Not supported.
- TOutIt mvPut (TOutIt avOut, const CDateTime &arDateTime, const std::string &arFormat) const
  This function outputs a date and time to a stream.

Static Public Attributes

- static std::locale::id id
  The facet ID.

Protected Member Functions

- virtual TOutIt mvDoPut (TOutIt avOut, const CDateTime &arDateTime, const std::string &arFormat) const
  This function implements the behavior of the mvPut() function.

Static Protected Member Functions

- static std::string smvExpandFormat (std::string avFormat)

  This function expands the date and time format by replacing format specifiers with their equivalent format specifiers.
- static TInIt smvGetTzName (TInIt arNext, TInIt arEnd, std::ios_base::iostate &arIoState, int &arDst)

  This function parses an input iterator for a timezone name.
- static TInIt smvGetTzOffset (TInIt arNext, TInIt arEnd, std::ios_base &arIosBase, std::ios_base::iostate &arIoState, int &arDst)

  This function parses an input iterator for a timezone name.
- static TOutIt smvPutTzName (TOutIt avOut, int avDst)

  This function sends the timezone name to an output iterator.
- static TOutIt smvPutTzOffset (TOutIt avOut, int avDst) noexcept

  This function sends the timezone offset to an output iterator.
- static TInIt smvGetWeekDayName (TInIt avNext, TInIt avEnd, std::ios_base &arIosBase, std::ios_base::iostate &arIoState, int &arWeekDayNo, bool avIsShort)

  This function parses an input iterator for a week day name.
- static TInIt smvGetMonthName (TInIt avNext, TInIt avEnd, std::ios_base &arIosBase, std::ios_base::iostate &arIoState, int &arMonthNo, bool avIsShort)

  This function parses an input iterator for a month name.
- static TOutIt smvPutWeekDayName (TOutIt avOut, int avWeekDayNo, bool avIsShort)

  This function sends the week day name of a given week day to an output iterator.
- static TOutIt smvPutMonthName (TOutIt avOut, int avMonthNo, bool avIsShort)

  This function sends the month name of a given month to an output iterator.
- static void smvCalcWeekBasedDetails (const CDate &arDate, int &arWeekBasedYear, int *apWeekBasedWeekNo) noexcept

  This function determines the week-based year and week-based week number of a given date.
- static int smvCalcWeekNo (const CDate &arDate, int avYear, int avDay) noexcept

  This function determines the week number of a given date within a given year.
- static TInIt smvIgnoreChar (TInIt avNext, TInIt avEnd, std::ios_base::iostate &arIoState, char avExpectedChar) noexcept

  This function parses an input iterator for a specific character. This function moves the input iterator forward one character if the expected character is found.
- static TInIt smvIgnoreChar (TInIt avNext, TInIt avEnd, std::ios_base::iostate &arIoState, char avExpectedChar) noexcept

  This function parses an input iterator for a specific character. This function moves the input iterator forward one character if the expected character is found.
- static TInIt smvGetNumber (TInIt avNext, TInIt avEnd, std::ios_base &arIosBase, std::ios_base::iostate &arIoState, int *apNumber, int avDigits, char avPadding) noexcept

  This function parses an input iterator for an integer value.
- static TInIt smvGetNumber (TInIt avNext, TInIt avEnd, std::ios_base &arIosBase, std::ios_base::iostate &arIoState, int *apNumber, int avDigits, char avPadding) noexcept

  This function parses an input iterator for an integer value.
- static TOutIt smvPutNumber1 (TOutIt avOut, int avNumber) noexcept

  This function sends a 1-digit number to an output iterator.
- static TOutIt smvPutNumber1 (TOutIt avOut, int avNumber) noexcept

  This function sends a 1-digit number to an output iterator.
- static TOutIt smvPutNumber2Zero (TOutIt avOut, int avNumber) noexcept

  This function sends a 2-digit zero padded number to an output iterator.
- static TOutIt smvPutNumber2Zero (TOutIt avOut, int avNumber) noexcept

  This function sends a 2-digit zero padded number to an output iterator.
- static TOutIt smvPutNumber2Blank (TOutIt avOut, int avNumber) noexcept

  This function sends a 2-digit blank padded number to an output iterator.

- static TOutIt smvPutNumber2Blank (TOutIt avOut, int avNumber) noexcept

  This function sends a 2-digit blank padded number to an output iterator.
- static TOutIt smvPutNumber3Zero (TOutIt avOut, int avNumber) noexcept

  This function sends a 3-digit zero padded number to an output iterator.
- static TOutIt smvPutNumber3Zero (TOutIt avOut, int avNumber) noexcept

  This function sends a 3-digit zero padded number to an output iterator.
- static TOutIt smvPutNumber4Zero (TOutIt avOut, int avNumber) noexcept

  This function sends a 4-digit zero padded number to an output iterator.
- static TOutIt smvPutNumber4Zero (TOutIt avOut, int avNumber) noexcept

  This function sends a 4-digit zero padded number to an output iterator.
- static TInIt smvGetAmPm (TInIt avNext, TInIt avEnd, std::ios_base &arIosBase, std::ios_base::iostate &arIoState, bool &arHasAmPm, bool &arIsPm)

  This function parses an input iterator for an AM/PM indicator.
- static TOutIt smvPutAmPm (TOutIt avOut, int avHour)

  This function sends AM or PM to an output iterator based on the given hour.

6.12.1 Detailed Description

This class is a CDateTime class output facet.

Note

This class uses the Template Method design pattern.

6.12.2 Constructor & Destructor Documentation

6.12.2.1 lfc1::datetime::CDateTimePut::CDateTimePut (size_t *avRelease* = `0`) `[explicit]`

This function creates a CDateTimePut object.

Parameters

in	*avRelease*	Indicates who controls the lifetime of the facet. (0 means locale)

6.12.3 Member Function Documentation

6.12.3.1 CDateTimePut::TOutIt lfc1::datetime::CDateTimePut::mvDoPut (TOutIt *avOut*, const CDateTime & *arDateTime*, const std::string & *arFormat*) const `[protected]`, `[virtual]`

This function implements the behavior of the mvPut() function.

Parameters

in	*avOut*	The output iterator.
in	*arDateTime*	The date and time to be sent to the output stream.
in	*arFormat*	The format of the date and time to be sent to the output stream.

Returns

The output iterator.

6.12.3.2 CDateTimePut::TOutIt lfc1::datetime::CDateTimePut::mvPut (TOutIt *avOut,* const CDateTime & *arDateTime,* const std::string & *arFormat*) const

This function outputs a date and time to a stream.

This function recognizes the following format specifiers which are identical to the date specific format specifiers of the C language strftime() function except for N:

```
%% - A percent character.
%a - Abbreviated weekday name.
%A - Full weekday name.
%b - Abbreviated month name.
%B - Full month name.
%c - Default date and time format.
%C - Century [00-99].
%d - Day of the month zero padded [01-31].
%D - Equivalent to %m/%d/%y.
%e - Day of the month blank padded [ 1-31].
%F - Equivalent to %Y-%m-%d.
%g - Week-based year without century [00-99].   (See week-based year explanation below.)
%G - Week-based year with century [0000-9999].  (See week-based year explanation below.)
%h - Equivalent to %b.
%H - Hour in 24-hour format [00-23] zero padded.
%I - Hour in 12-hour format [01-12] zero padded.
%j - Day of the year [001-366].
%k - Hour in 24-hour format [00-23] blank padded.
%l - Hour in 12-hour format [01-12] blank padded.
%m - Month number [01-12].
%M - Minute [00-59].
%n - Newline character.
%N - Millisecond [000-999].
%p - AM/PM designation.
%r - Default 12-hour time format.
%R - Equivalent to %H:%M.
%S - Second [00-59].
%t - Horizontal tab character.
%T - Equivalent to %H:%M:%S.
%u - Day of the week [1-7], Monday is 1.
%U - Week of the year [00-53], the first Sunday starts week 1.
%V - Week number of a week-based year [01-53].  (See week-based year explanation below.)
%w - Day of the week [0-6], Sunday is 0.
%W - Week of the year [00-53], the first Monday starts week 1.
%x - Default date format.
%X - Default time format.
%y - Year without century [00-99].
%Y - Year with century [0000-9999].
%z - Time zone offset.
%Z - Time zone name.
```

Any other sequence of characters not included in the above list will be taken literally.

For an explanation of the week-based year, see CDatePut.

Parameters

in	*avOut*	The output iterator.
in	*arDateTime*	The date and time to be sent to the output stream.
in	*arFormat*	The format of the date and time to be sent to the output stream.

Returns

The output iterator.

6.12.3.3 void lfc1::datetime::CDateBase::smvCalcWeekBasedDetails (const CDate & *arDate,* int & *arWeekBasedYear,* int * *apWeekBasedWeekNo*) [static], [protected], [noexcept], [inherited]

This function determines the week-based year and week-based week number of a given date.

Parameters

in	*arDate*	The given date expressed as a Julian day number.
in,out	*arWeekBased-Year*	Set to the starting week-based year on input and contains the resulting week-based year on output.
out	*apWeekBased-WeekNo*	The resulting week-based week number of the given date. If null, week-based week number is not stored.

6.12.3.4 int lfc1::datetime::CDateBase::smvCalcWeekNo (const CDate & *arDate,* int *avYear,* int *avDay*) [static], [protected], [noexcept], [inherited]

This function determines the week number of a given date within a given year.

Parameters

in	*arDate*	The given date expressed as a Julian day number.
in	*avYear*	The year of the given date.
in	*avDay*	The starting day of the week. (0 or 6 = Monday or Sunday)

Returns

The week number of a given date within a given year.

6.12.3.5 std::string lfc1::datetime::CDateTimeBase::smvExpandFormat (std::string *avFormat*) [static], [protected], [inherited]

This function expands the date and time format by replacing format specifiers with their equivalent format specifiers.

Parameters

in	*avFormat*	The date and time format to be expanded.

Returns

The expanded format string.

6.12.3.6 CTimeBase::TInIt lfc1::datetime::CTimeBase::smvGetAmPm (TInIt *avNext,* TInIt *avEnd,* std::ios_base & *arIosBase,* std::ios_base::iostate & *arIoState,* bool & *arHasAmPm,* bool & *arIsPm*) [static], [protected], [inherited]

This function parses an input iterator for an AM/PM indicator.

Parameters

in	*avNext*	The input iterator.
in	*avEnd*	The input end iterator.
in	*arIosBase*	The stream formatting information.
out	*arIoState*	The stream state.
out	*arHasAmPm*	The storage for the resulting has AM/PM indicator (true) or not (false).
out	*arIsPm*	The storage for the resulting PM (true) or AM (false) indicator.

Returns

The input iterator.

6.12.3.7 CDateBase::TInIt lfc1::datetime::CDateBase::smvGetMonthName (TInIt *avNext,* TInIt *avEnd,* std::ios_base & *arIosBase,* std::ios_base::iostate & *arIoState,* int & *arMonthNo,* bool *avIsShort*) [static], [protected], [inherited]

This function parses an input iterator for a month name.

Parameters

in	*avNext*	The input iterator.
in	*avEnd*	The input end iterator.
in	*arIosBase*	The stream formatting information.
out	*arIoState*	The stream state.
out	*arMonthNo*	The storage for the resulting month number.
in	*avIsShort*	Short (true) or long (false) month name indicator.

Returns

The input iterator.

6.12.3.8 CFacetBase::TInIt lfc1::datetime::CFacetBase::smvGetNumber (TInIt *avNext,* TInIt *avEnd,* std::ios_base & *arIosBase,* std::ios_base::iostate & *arIoState,* int * *apNumber,* int *avDigits,* char *avPadding*) [static], [protected], [noexcept], [inherited]

This function parses an input iterator for an integer value.

Parameters

in	*avNext*	The input iterator.
in	*avEnd*	The input end iterator.
in	*arIosBase*	The stream formatting information.
out	*arIoState*	The stream state.
out	*apNumber*	The storage for the resulting integer.
in	*avDigits*	The maximum number of digits allowed for the integer.
in	*avPadding*	The padding character allowed for the integer value.

Return values

ios_base::goodbit	A valid integer was found.
ios_base::failbit	A valid integer was not found.

6.12.3.9 CFacetBase::TInIt lfc1::datetime::CFacetBase::smvGetNumber (TInIt *avNext,* TInIt *avEnd,* std::ios_base & *arIosBase,* std::ios_base::iostate & *arIoState,* int * *apNumber,* int *avDigits,* char *avPadding*) [static], [protected], [noexcept], [inherited]

This function parses an input iterator for an integer value.

Parameters

in	*avNext*	The input iterator.
in	*avEnd*	The input end iterator.
in	*arIosBase*	The stream formatting information.
out	*arIoState*	The stream state.
out	*apNumber*	The storage for the resulting integer.
in	*avDigits*	The maximum number of digits allowed for the integer.
in	*avPadding*	The padding character allowed for the integer value.

Return values

ios_base::goodbit	A valid integer was found.
ios_base::failbit	A valid integer was not found.

6.12.3.10 CDateTimeBase::TInIt lfc1::datetime::CDateTimeBase::smvGetTzName (TInIt *avNext,* TInIt *avEnd,* std::ios_base::iostate & *arIoState,* int & *arDst*) [static],[protected],[inherited]

This function parses an input iterator for a timezone name.

Parameters

in	*avNext*	The input iterator.
in	*avEnd*	The input end iterator.
out	*arIoState*	The stream state.
out	*arDst*	The storage for the resulting daylight saving time indicator.

Returns

The input iterator.

6.12.3.11 CDateTimeBase::TInIt lfc1::datetime::CDateTimeBase::smvGetTzOffset (TInIt *avNext,* TInIt *avEnd,* std::ios_base & *arIosBase,* std::ios_base::iostate & *arIoState,* int & *arDst*) [static],[protected],[inherited]

This function parses an input iterator for a timezone name.

Parameters

in	*avNext*	The input iterator.
in	*avEnd*	The input end iterator.
in	*arIosBase*	The stream formatting information.
out	*arIoState*	The stream state.
out	*arDst*	The storage for the resulting daylight saving time indicator.

Returns

The input iterator.

6.12.3.12 CDateBase::TInIt lfc1::datetime::CDateBase::smvGetWeekDayName (TInIt *avNext,* TInIt *avEnd,* std::ios_base & *arIosBase,* std::ios_base::iostate & *arIoState,* int & *arWeekDayNo,* bool *avIsShort*) [static],[protected],[inherited]

This function parses an input iterator for a week day name.

Parameters

in	*avNext*	The input iterator.
in	*avEnd*	The input end iterator.
in	*arIosBase*	The stream formatting information.
out	*arIoState*	The stream state.
out	*arWeekDayNo*	The storage for the resulting day of the week. (0 = Sunday)
in	*avIsShort*	Short (true) or long (false) week day name indicator.

Returns

The input iterator.

6.12.3.13 CFacetBase::TInIt lfc1::datetime::CFacetBase::smvIgnoreChar (TInIt *avNext,* TInIt *avEnd,* std::ios_base::iostate & *arIoState,* char *avExpectedChar*) [static],[protected],[noexcept],[inherited]

This function parses an input iterator for a specific character. This function moves the input iterator forward one character if the expected character is found.

Parameters

in	*avNext*	The input iterator.
in	*avEnd*	The input end iterator.
out	*arIoState*	The stream state.
in	*avExpectedChar*	The next character expected from the input iterator.

Return values

ios_base::goodbit	The expected character was found.
ios_base::failbit	The expected character was not found.

6.12.3.14 CFacetBase::TInIt lfc1::datetime::CFacetBase::smvIgnoreChar (TInIt *avNext,* TInIt *avEnd,* std::ios_base::iostate & *arIoState,* char *avExpectedChar*) [static],[protected],[noexcept],[inherited]

This function parses an input iterator for a specific character. This function moves the input iterator forward one character if the expected character is found.

Parameters

in	*avNext*	The input iterator.
in	*avEnd*	The input end iterator.
out	*arIoState*	The stream state.
in	*avExpectedChar*	The next character expected from the input iterator.

Return values

ios_base::goodbit	The expected character was found.
ios_base::failbit	The expected character was not found.

6.12.3.15 CTimeBase::TOutIt lfc1::datetime::CTimeBase::smvPutAmPm (TOutIt *avOut,* int *avHour*) [static],[protected],[inherited]

This function sends AM or PM to an output iterator based on the given hour.

Parameters

in	*avOut*	The output iterator.
in	*avHour*	The given hour.

Returns

The output iterator.

6.12.3.16 CDateBase::TOutIt lfc1::datetime::CDateBase::smvPutMonthName (TOutIt *avOut,* int *avMonthNo,* bool *avIsShort*) [static],[protected],[inherited]

This function sends the month name of a given month to an output iterator.

Parameters

in	*avOut*	The output iterator.
in	*avMonthNo*	The number of the month.
in	*avIsShort*	Short (true) or long (false) month name indicator.

Returns

The output iterator.

6.12.3.17 CFacetBase::TOutIt lfc1::datetime::CFacetBase::smvPutNumber1 (TOutIt *avOut,* int *avNumber*) [static],[protected],[noexcept],[inherited]

This function sends a 1-digit number to an output iterator.

Parameters

in	*avOut*	The output iterator.
in	*avNumber*	The 1-digit number.

Returns

The output iterator.

6.12.3.18 CFacetBase::TOutIt lfc1::datetime::CFacetBase::smvPutNumber1 (TOutIt *avOut,* int *avNumber*) [static],[protected],[noexcept],[inherited]

This function sends a 1-digit number to an output iterator.

Parameters

in	*avOut*	The output iterator.
in	*avNumber*	The 1-digit number.

Returns

The output iterator.

6.12.3.19 CFacetBase::TOutIt lfc1::datetime::CFacetBase::smvPutNumber2Blank (TOutIt *avOut,* int *avNumber*) [static],[protected],[noexcept],[inherited]

This function sends a 2-digit blank padded number to an output iterator.

Parameters

in	*avOut*	The output iterator.
in	*avNumber*	The 2-digit number.

Returns

The output iterator.

6.12.3.20 CFacetBase::TOutIt lfc1::datetime::CFacetBase::smvPutNumber2Blank (TOutIt *avOut,* int *avNumber*) [static],[protected],[noexcept],[inherited]

This function sends a 2-digit blank padded number to an output iterator.

Parameters

in	*avOut*	The output iterator.
in	*avNumber*	The 2-digit number.

Returns

The output iterator.

6.12.3.21 CFacetBase::TOutIt lfc1::datetime::CFacetBase::smvPutNumber2Zero (TOutIt *avOut,* int *avNumber*) [static],[protected],[noexcept],[inherited]

This function sends a 2-digit zero padded number to an output iterator.

Parameters

in	*avOut*	The output iterator.
in	*avNumber*	The 2-digit number.

Returns

The output iterator.

6.12.3.22 CFacetBase::TOutIt lfc1::datetime::CFacetBase::smvPutNumber2Zero (TOutIt *avOut,* int *avNumber*) [static],[protected],[noexcept],[inherited]

This function sends a 2-digit zero padded number to an output iterator.

Parameters

in	*avOut*	The output iterator.
in	*avNumber*	The 2-digit number.

Returns

The output iterator.

6.12.3.23 CFacetBase::TOutIt lfc1::datetime::CFacetBase::smvPutNumber3Zero (TOutIt *avOut,* int *avNumber*) [static],[protected],[noexcept],[inherited]

This function sends a 3-digit zero padded number to an output iterator.

Parameters

in	*avOut*	The output iterator.
in	*avNumber*	The 3-digit number.

Returns

The output iterator.

6.12.3.24 CFacetBase::TOutIt lfc1::datetime::CFacetBase::smvPutNumber3Zero (TOutIt *avOut,* int *avNumber*)

`[static],[protected],[noexcept],[inherited]`

This function sends a 3-digit zero padded number to an output iterator.

Parameters

in	*avOut*	The output iterator.
in	*avNumber*	The 3-digit number.

Returns

The output iterator.

6.12.3.25 CFacetBase::TOutIt lfc1::datetime::CFacetBase::smvPutNumber4Zero (TOutIt *avOut,* int *avNumber*)

`[static],[protected],[noexcept],[inherited]`

This function sends a 4-digit zero padded number to an output iterator.

Parameters

in	*avOut*	The output iterator.
in	*avNumber*	The 4-digit number.

Returns

The output iterator.

6.12.3.26 CFacetBase::TOutIt lfc1::datetime::CFacetBase::smvPutNumber4Zero (TOutIt *avOut,* int *avNumber*)

`[static],[protected],[noexcept],[inherited]`

This function sends a 4-digit zero padded number to an output iterator.

Parameters

in	*avOut*	The output iterator.
in	*avNumber*	The 4-digit number.

Returns

The output iterator.

6.12.3.27 CDateTimeBase::TOutIt lfc1::datetime::CDateTimeBase::smvPutTzName (TOutIt *avOut,* int *avDst*)

`[static],[protected],[inherited]`

This function sends the timezone name to an output iterator.

Parameters

in	*avOut*	The output iterator.
in	*avDst*	The daylight saving time indicator.

Returns

The output iterator.

6.12.3.28 CDateTimeBase::TOutIt lfc1::datetime::CDateTimeBase::smvPutTzOffset (TOutIt *avOut,* int *avDst*) [static],[protected],[noexcept],[inherited]

This function sends the timezone offset to an output iterator.

Parameters

in	*avOut*	The output iterator.
in	*avDst*	The daylight saving time indicator.

Returns

The output iterator.

6.12.3.29 CDateBase::TOutIt lfc1::datetime::CDateBase::smvPutWeekDayName (TOutIt *avOut,* int *avWeekDayNo,* bool *avIsShort*) [static],[protected],[inherited]

This function sends the week day name of a given week day to an output iterator.

Parameters

in	*avOut*	The output iterator.
in	*avWeekDayNo*	The number of the week day.
in	*avIsShort*	Short (true) or long (false) week day name indicator.

Returns

The output iterator.

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

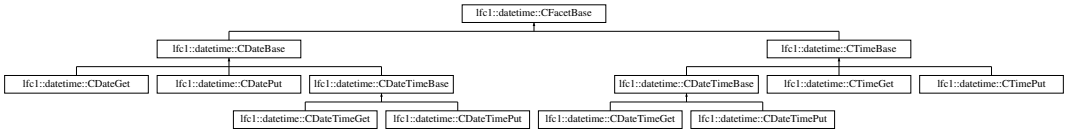
- include/lfc1/datetime/cdatetime.hpp
- datetime/library/src/cdatetimeput.cpp

6.13 lfc1::datetime::CFacetBase Class Reference

This class serves as the base class for date and time input/output facets.

`#include <lfc1/datetime/cfacetbase.hpp>`

Inheritance diagram for lfc1::datetime::CFacetBase:



Public Types

- typedef
  std::istreambuf_iterator< char > TInIt

This type is the input iterator.

- typedef
  std::ostreambuf_iterator< char > TOutIt

  This type is the output iterator.

Protected Member Functions

- CFacetBase () noexcept

  This function creates a default CFacetBase object.
- CFacetBase (const CFacetBase &)=delete

  Not supported.
- CFacetBase (CFacetBase &&)=delete

  Not supported.
- virtual ~CFacetBase () noexcept

  This function destroys a CFacetBase object.
- CFacetBase & operator= (const CFacetBase &)=delete

  Not supported.
- CFacetBase & operator= (CFacetBase &&)=delete

  Not supported.

Static Protected Member Functions

- static TInIt smvIgnoreChar (TInIt avNext, TInIt avEnd, std::ios_base::iostate &arIoState, char avExpectedChar) noexcept

  This function parses an input iterator for a specific character. This function moves the input iterator forward one character if the expected character is found.
- static TInIt smvGetNumber (TInIt avNext, TInIt avEnd, std::ios_base &arIosBase, std::ios_base::iostate &arIoState, int *apNumber, int avDigits, char avPadding) noexcept

  This function parses an input iterator for an integer value.
- static TOutIt smvPutNumber1 (TOutIt avOut, int avNumber) noexcept

  This function sends a 1-digit number to an output iterator.
- static TOutIt smvPutNumber2Zero (TOutIt avOut, int avNumber) noexcept

  This function sends a 2-digit zero padded number to an output iterator.
- static TOutIt smvPutNumber2Blank (TOutIt avOut, int avNumber) noexcept

  This function sends a 2-digit blank padded number to an output iterator.
- static TOutIt smvPutNumber3Zero (TOutIt avOut, int avNumber) noexcept

  This function sends a 3-digit zero padded number to an output iterator.
- static TOutIt smvPutNumber4Zero (TOutIt avOut, int avNumber) noexcept

  This function sends a 4-digit zero padded number to an output iterator.

6.13.1 Detailed Description

This class serves as the base class for date and time input/output facets.

6.13.2 Member Function Documentation

6.13.2.1 CFacetBase::TInIt lfc1::datetime::CFacetBase::smvGetNumber (TInIt *avNext*, TInIt *avEnd*, std::ios_base & *arIosBase*, std::ios_base::iostate & *arIoState*, int * *apNumber*, int *avDigits*, char *avPadding*) [static], [protected], [noexcept]

This function parses an input iterator for an integer value.

Parameters

in	*avNext*	The input iterator.
in	*avEnd*	The input end iterator.
in	*arIosBase*	The stream formatting information.
out	*arIoState*	The stream state.
out	*apNumber*	The storage for the resulting integer.
in	*avDigits*	The maximum number of digits allowed for the integer.
in	*avPadding*	The padding character allowed for the integer value.

Return values

ios_base::goodbit	A valid integer was found.
ios_base::failbit	A valid integer was not found.

6.13.2.2 CFacetBase::TInIt lfc1::datetime::CFacetBase::smvIgnoreChar (TInIt *avNext,* TInIt *avEnd,* std::ios_base::iostate & *arIoState,* char *avExpectedChar*) [static],[protected],[noexcept]

This function parses an input iterator for a specific character. This function moves the input iterator forward one character if the expected character is found.

Parameters

in	*avNext*	The input iterator.
in	*avEnd*	The input end iterator.
out	*arIoState*	The stream state.
in	*avExpectedChar*	The next character expected from the input iterator.

Return values

ios_base::goodbit	The expected character was found.
ios_base::failbit	The expected character was not found.

6.13.2.3 CFacetBase::TOutIt lfc1::datetime::CFacetBase::smvPutNumber1 (TOutIt *avOut,* int *avNumber*) [static],[protected],[noexcept]

This function sends a 1-digit number to an output iterator.

Parameters

in	*avOut*	The output iterator.
in	*avNumber*	The 1-digit number.

Returns

The output iterator.

6.13.2.4 CFacetBase::TOutIt lfc1::datetime::CFacetBase::smvPutNumber2Blank (TOutIt *avOut,* int *avNumber*) [static],[protected],[noexcept]

This function sends a 2-digit blank padded number to an output iterator.

Parameters

in	*avOut*	The output iterator.
in	*avNumber*	The 2-digit number.

Returns

The output iterator.

6.13.2.5 CFacetBase::TOutIt lfc1::datetime::CFacetBase::smvPutNumber2Zero (TOutIt *avOut,* int *avNumber*) [static],[protected],[noexcept]

This function sends a 2-digit zero padded number to an output iterator.

Parameters

in	*avOut*	The output iterator.
in	*avNumber*	The 2-digit number.

Returns

The output iterator.

6.13.2.6 CFacetBase::TOutIt lfc1::datetime::CFacetBase::smvPutNumber3Zero (TOutIt *avOut,* int *avNumber*) [static],[protected],[noexcept]

This function sends a 3-digit zero padded number to an output iterator.

Parameters

in	*avOut*	The output iterator.
in	*avNumber*	The 3-digit number.

Returns

The output iterator.

6.13.2.7 CFacetBase::TOutIt lfc1::datetime::CFacetBase::smvPutNumber4Zero (TOutIt *avOut,* int *avNumber*) [static],[protected],[noexcept]

This function sends a 4-digit zero padded number to an output iterator.

Parameters

in	*avOut*	The output iterator.
in	*avNumber*	The 4-digit number.

Returns

The output iterator.

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

- include/lfc1/datetime/cfacetbase.hpp
- datetime/library/src/cfacetbase.cpp

6.14 lfc1::datetime::CLangInfo Class Reference

This class obtains locale specific date and time strings.

```
#include <lfc1/datetime/clanginfo.hpp>
```

Public Member Functions

- CLangInfo ()=delete

  Not supported.
- CLangInfo (const CLangInfo &)=delete

  Not supported.
- CLangInfo (CLangInfo &&)=delete

  Not supported.
- ~CLangInfo ()=delete

  Not supported.
- CLangInfo & operator= (const CLangInfo &)=delete

  Not supported.
- CLangInfo & operator= (CLangInfo &&)=delete

  Not supported.

Static Public Member Functions

- static std::string smvGetDateFormat ()

  This function obtains the date format.
- static std::string smvGetTime24Format ()

  This function obtains the 24-hour time format.
- static std::string smvGetTime12Format ()

  This function obtains the 12-hour time format.
- static std::string smvGetDateTimeFormat ()

  This function obtains the date and time format.
- static std::string smvGetLongDayName (int avWeekDay)

  This function obtains the long name of the specified week day.
- static std::string smvGetShortDayName (int avWeekDay)

  This function obtains the short name of the specified week day.
- static std::string smvGetLongMonthName (int avMonth)

  This function obtains the long name of the specified month.
- static std::string smvGetShortMonthName (int avMonth)

  This function obtains the short name of the specified month.
- static std::string smvGetAMString ()

  This function obtains the string for AM.
- static std::string smvGetPMString ()

  This function obtains the string for PM.

6.14.1 Detailed Description

This class obtains locale specific date and time strings.

For the getter functions to work correctly for the desired locale, the desired locale must be selected as the global locale. This is not ideal because imbuing streams with a different locale whose facets uses this class will not work correctly. Unfortunately, that is how it is in a UNIX environment because locale specific information can only be accessed via C API functions and these functions require the desired locale to be set as the global locale.

6.14.2 Member Function Documentation

6.14.2.1 std::string lfc1::datetime::CLangInfo::smvGetAMString ( ) [static]

This function obtains the string for AM.

Returns

The string for AM.

6.14.2.2 std::string lfc1::datetime::CLangInfo::smvGetDateFormat ( ) [static]

This function obtains the date format.

Returns

The date format.

6.14.2.3 std::string lfc1::datetime::CLangInfo::smvGetDateTimeFormat ( ) [static]

This function obtains the date and time format.

Returns

The date and time format.

6.14.2.4 std::string lfc1::datetime::CLangInfo::smvGetLongDayName (int *avWeekDay*) [static]

This function obtains the long name of the specified week day.

Parameters

in	*avWeekDay*	The number of the week day in the range 0 to 6 with 0 being Sunday.

Returns

The long name of the specified week day.

Exceptions

std::out_of_range	Indicates the given week day is out-of-range.

6.14.2.5 std::string lfc1::datetime::CLangInfo::smvGetLongMonthName (int *avMonth*) [static]

This function obtains the long name of the specified month.

Parameters

in	*avMonth*	The number of the month in the range 1 to 12 with 1 being January.

Returns

The long name of the specified month.

Exceptions

std::out_of_range	Indicates the given month is out-of-range.

6.14.2.6 std::string lfc1::datetime::CLangInfo::smvGetPMString ( ) [static]

This function obtains the string for PM.

Returns

The string for PM.

6.14.2.7 std::string lfc1::datetime::CLangInfo::smvGetShortDayName (int *avWeekDay*) [static]

This function obtains the short name of the specified week day.

Parameters

in	*avWeekDay*	The number of the week day in the range 0 to 6 with 0 being Sunday.

Returns

The short name of the specified week day.

Exceptions

std::out_of_range	Indicates the given week day is out-of-range.

6.14.2.8 std::string lfc1::datetime::CLangInfo::smvGetShortMonthName (int *avMonth*) [static]

This function obtains the short name of the specified month.

Parameters

in	*avMonth*	The number of the month in the range 1 to 12 with 1 being January.

Returns

The short name of the specified month.

Exceptions

std::out_of_range	Indicates the given month is out-of-range.

6.14.2.9 std::string lfc1::datetime::CLangInfo::smvGetTime12Format ( ) [static]

This function obtains the 12-hour time format.

Returns

The 12-hour time format.

6.14.2.10 std::string lfc1::datetime::CLangInfo::smvGetTime24Format ( ) [static]

This function obtains the 24-hour time format.

Returns

The 24-hour time format.

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

- include/lfc1/datetime/clanginfo.hpp
- datetime/library/src/clanginfo.cpp

6.15 lfc1::datetime::CTime Class Reference

This class represents the time since midnight.

`#include <lfc1/datetime/ctime.hpp>`

Classes

- struct STime

  A structure that holds the individual parts of a time.

Public Member Functions

- CTime () noexcept

  This function creates a default CTime object based on the current system time.
- CTime (int avTimeNo)

  This function creates a CTime object using the given Julian day number.
- CTime (int avHour, int avMinute, int avSecond, int avMillisecond)

  This function creates a CTime object using the given hour, minute, second and millisecond.
- CTime (const CTime &arRHS) noexcept

  This function copy constructs a CTime object.
- CTime (CTime &&arRHS) noexcept

  This function move constructs a CTime object.
- ~CTime () noexcept

  This function destroys a CTime object.
- CTime & operator= (int avTimeNo)

  This function assigns the given Julian day number to the CTime object.
- CTime & operator= (const CTime &arRHS) noexcept

  This function assigns a CTime object to another CTime object.
- CTime & operator= (CTime &&arRHS) noexcept

  This function moves a CTime object to another CTime object.
- operator int () const noexcept

  This function obtains the equivalent milliseconds since midnight number of a CTime object.
- void mvSetTime () noexcept

  This function assigns the current system time to a CTime object.
- void mvSetTime (int avHour, int avMinute, int avSecond, int avMillisecond)

  This function sets a CTime object based on the given hour, minute, second and millisecond.
- STime mvGetTime () const noexcept

This function obtains the equivalent hour, minute, second and millisecond of the CTime object.

- void swap (CTime &arRHS) noexcept

  This function swaps a CTime object with another CTime object.
- CTime & operator++ ()

  This function pre-increments the CTime object by one millisecond.
- CTime operator++ (int)

  This function post-increments the CTime object by one millisecond.
- CTime & operator-- ()

  This function pre-decrements the CTime object by one millisecond.
- CTime operator-- (int)

  This function post-decrements the CTime object by one millisecond.
- CTime & operator+= (const CTimeDuration &arTimeDuration)

  This function increments the CTime object by the given duration.
- CTime & operator-= (const CTimeDuration &arTimeDuration)

  This function decrements the CTime object by the given duration.
- std::ios_base::iostate mvInput (std::istream &arStream)

  This function receives a CTime object from the given input stream.
- std::ios_base::iostate mvOutput (std::ostream &arStream) const

  This function sends a CTime object to the given output stream.

Static Public Member Functions

- static int smvGetIndex () noexcept

  This function obtains an iword/pword index.
- static void smvValidateTime (int avHour, int avMinute, int avSecond, int avMillisecond)

  This function validates a time to be in the range 00:00:00.000 to 23:59:59.999.

Static Public Attributes

- static const auto HR_PER_DAY = 24

  The number of hours in a day.
- static const auto MIN_HR = 0

  The lowest hour number in a day.
- static constexpr auto MAX_HR = HR_PER_DAY - 1

  The highest hour number in a day.
- static const auto MIN_PER_HR = 60

  The number of minutes in an hour.
- static const auto MIN_MIN = 0

  The lowest minute number in an hour.
- static constexpr auto MAX_MIN = MIN_PER_HR - 1

  The highest minute number in an hour.
- static const auto SEC_PER_MIN = 60

  The number of seconds in a minute.
- static const auto MIN_SEC = 0

  The lowest second number in a minute.
- static constexpr auto MAX_SEC = SEC_PER_MIN - 1

  The highest second number in a minute.
- static const auto MSEC_PER_SEC = 1000

  The number of milliseconds in a second.
- static const auto MIN_MSEC = 0

The lowest millisecond number in a second.

- static constexpr auto MAX_MSEC = MSEC_PER_SEC - 1

  The highest millisecond number in a second.
- static constexpr auto MSEC_PER_MIN = SEC_PER_MIN * MSEC_PER_SEC

  The no. of millisecs. per minute.
- static constexpr auto MSEC_PER_HR = MIN_PER_HR * MSEC_PER_MIN

  The no. of milliseconds per hour.
- static constexpr auto MSEC_PER_DAY = HR_PER_DAY * MSEC_PER_HR

  The no. of milliseconds per day.
- static constexpr auto SEC_PER_HR = MIN_PER_HR * SEC_PER_MIN

  The number of seconds per hour.
- static constexpr auto SEC_PER_DAY = HR_PER_DAY * SEC_PER_HR

  The number of seconds per day.
- static constexpr auto MIN_PER_DAY = HR_PER_DAY * MIN_PER_HR

  The number of minutes per day.
- static const auto MIN_TIMENO = 0

  The lowest millisecond of a day.
- static constexpr auto MAX_TIMENO = MSEC_PER_DAY - 1

  The highest millisecond of a day.

6.15.1 Detailed Description

This class represents the time since midnight.

This class supports the following operators:

```
binary arithmetic operators: +=, -=
unary arithmetic operators:  ++, --
```

6.15.2 Constructor & Destructor Documentation

6.15.2.1 lfc1::datetime::CTime::CTime (int *avTimeNo*)

This function creates a CTime object using the given Julian day number.

Parameters

in	*avTimeNo*	The milliseconds since midnight number to assign to the CTime object.

Exceptions

std::out_of_range	Indicates that the given milliseconds since midnight number is not within the valid range.

6.15.2.2 lfc1::datetime::CTime::CTime (int *avHour,* int *avMinute,* int *avSecond,* int *avMillisecond*)

This function creates a CTime object using the given hour, minute, second and millisecond.

Parameters

in	*avHour*	The given hour.
in	*avMinute*	The given minute.
in	*avSecond*	The given second.
in	*avMillisecond*	The given millisecond.

6.15.2.3 lfc1::datetime::CTime::CTime (const CTime & *arRHS*) [noexcept]

This function copy constructs a CTime object.

Parameters

in	*arRHS*	The CTime object to be copied.

6.15.2.4 lfc1::datetime::CTime::CTime (CTime && *arRHS*) [noexcept]

This function move constructs a CTime object.

Parameters

in	*arRHS*	The CTime object to be moved.

6.15.3 Member Function Documentation

6.15.3.1 CTime::STime lfc1::datetime::CTime::mvGetTime ( ) const [noexcept]

This function obtains the equivalent hour, minute, second and millisecond of the CTime object.

Returns

The equivalent hour, minute, second and millisecond of the CTime object.

6.15.3.2 std::ios_base::iostate lfc1::datetime::CTime::mvInput (std::istream & *arStream*)

This function receives a CTime object from the given input stream.

Parameters

in	*arStream*	The source stream.

Returns

The resulting stream state.

6.15.3.3 std::ios_base::iostate lfc1::datetime::CTime::mvOutput (std::ostream & *arStream*) const

This function sends a CTime object to the given output stream.

Parameters

in	*arStream*	The destination stream.

Returns

The resulting stream state.

6.15.3.4 void lfc1::datetime::CTime::mvSetTime (int *avHour,* int *avMinute,* int *avSecond,* int *avMillisecond*)

This function sets a CTime object based on the given hour, minute, second and millisecond.

Parameters

in	*avHour*	The given hour.
in	*avMinute*	The given minute.
in	*avSecond*	The given second.
in	*avMillisecond*	The given millisecond.

6.15.3.5 lfc1::datetime::CTime::operator int ( ) const [noexcept]

This function obtains the equivalent milliseconds since midnight number of a CTime object.

Returns

The equivalent milliseconds since midnight number of a CTime object.

6.15.3.6 CTime & lfc1::datetime::CTime::operator++ ( )

This function pre-increments the CTime object by one millisecond.

Returns

The CTime object.

Exceptions

std::out_of_range	Indicates incrementing the time will make the time go past the maximum supported time.

6.15.3.7 CTime lfc1::datetime::CTime::operator++ (int)

This function post-increments the CTime object by one millisecond.

Returns

The CTime object before the increment.

6.15.3.8 CTime & lfc1::datetime::CTime::operator+= (const CTimeDuration & *arTimeDuration*)

This function increments the CTime object by the given duration.

Parameters

in	*arTimeDuration*	The duration to add to the time.

Returns

The CTime object.

Exceptions

std::out_of_range	Indicates incrementing the time will make the time go past the maximum supported time.

6.15.3.9 CTime & lfc1::datetime::CTime::operator-- ( )

This function pre-decrements the CTime object by one millisecond.

Returns

The CTime object.

Exceptions

std::out_of_range	Indicates incrementing the time will make the time go below the minimum supported time.

6.15.3.10 CTime lfc1::datetime::CTime::operator-- (int)

This function post-decrements the CTime object by one millisecond.

Returns

The CTime object before the decrement.

6.15.3.11 CTime & lfc1::datetime::CTime::operator-= (const CTimeDuration & *arTimeDuration*)

This function decrements the CTime object by the given duration.

Parameters

in	*arTimeDuration*	The duration to subtract from the time.

Returns

The CTime object.

Exceptions

std::out_of_range	Indicates incrementing the time will make the time go below the minimum supported time.

6.15.3.12 CTime & lfc1::datetime::CTime::operator= (int *avTimeNo*)

This function assigns the given Julian day number to the CTime object.

Parameters

in	*avTimeNo*	The milliseconds since mignight number to assign to the CTime object.

Returns

The CTime object assigned to.

6.15.3.13 CTime & lfc1::datetime::CTime::operator= (const CTime & *arRHS*) [noexcept]

This function assigns a CTime object to another CTime object.

Parameters

in	*arRHS*	The CTime object to assign to another CTime object.

Returns

The CTime object assigned to.

6.15.3.14 CTime & lfc1::datetime::CTime::operator= (CTime && *arRHS*) [noexcept]

This function moves a CTime object to another CTime object.

Parameters

in	*arRHS*	The CTime object to move to another CTime object.

Returns

The CTime object assigned to.

6.15.3.15 int lfc1::datetime::CTime::smvGetIndex ( ) [static], [noexcept]

This function obtains an iword/pword index.

Returns

The iword/pword index.

6.15.3.16 void lfc1::datetime::CTime::smvValidateTime (int *avHour,* int *avMinute,* int *avSecond,* int *avMillisecond*) [static]

This function validates a time to be in the range 00:00:00.000 to 23:59:59.999.

Parameters

in	*avHour*	The hour of a time. (0 - 23)
in	*avMinute*	The minute of a time. (0 - 59)
in	*avSecond*	The second of a time. (0 - 59)
in	*avMillisecond*	The millisecond of a time. (0 - 999)

Exceptions

std::out_of_range	Indicates either the hour, minute, second or millisecond are not within valid ranges.

6.15.3.17 void lfc1::datetime::CTime::swap (CTime & *arRHS*) [noexcept]

This function swaps a CTime object with another CTime object.

Parameters

in, out	*arRHS*	The CTime object to swap with.

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

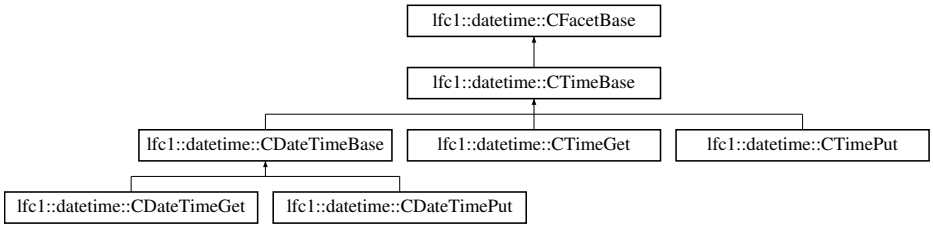
- include/lfc1/datetime/ctime.hpp
- datetime/library/src/ctime.cpp

6.16 lfc1::datetime::CTimeBase Class Reference

This class serves as the base class for time input/output facets.

`#include <lfc1/datetime/ctime.hpp>`

Inheritance diagram for lfc1::datetime::CTimeBase:



Public Types

- typedef
  std::istreambuf_iterator< char > TInIt

  This type is the input iterator.
- typedef
  std::ostreambuf_iterator< char > TOutIt

  This type is the output iterator.

Protected Member Functions

- CTimeBase () noexcept

  This function creates a default CTimeBase object.
- CTimeBase (const CTimeBase &)=delete

  Not supported.
- CTimeBase (CTimeBase &&)=delete

  Not supported.
- ~CTimeBase () noexceptoverride

  This function destroys a CTimeBase object.
- CTimeBase & operator= (const CTimeBase &)=delete

  Not supported.
- CTimeBase & operator= (CTimeBase &&)=delete

  Not supported.

Static Protected Member Functions

- static std::string smvExpandFormat (std::string avFormat)

  This function expands the time format by replacing format specifiers with their equivalent format specifiers.
- static TInIt smvGetAmPm (TInIt avNext, TInIt avEnd, std::ios_base &arIosBase, std::ios_base::iostate &arIoState, bool &arHasAmPm, bool &arIsPm)

This function parses an input iterator for an AM/PM indicator.

- static TOutIt smvPutAmPm (TOutIt avOut, int avHour)

  This function sends AM or PM to an output iterator based on the given hour.
- static TInIt smvIgnoreChar (TInIt avNext, TInIt avEnd, std::ios_base::iostate &arIoState, char avExpectedChar) noexcept

  This function parses an input iterator for a specific character. This function moves the input iterator forward one character if the expected character is found.
- static TInIt smvGetNumber (TInIt avNext, TInIt avEnd, std::ios_base &arIosBase, std::ios_base::iostate &arIoState, int *apNumber, int avDigits, char avPadding) noexcept

  This function parses an input iterator for an integer value.
- static TOutIt smvPutNumber1 (TOutIt avOut, int avNumber) noexcept

  This function sends a 1-digit number to an output iterator.
- static TOutIt smvPutNumber2Zero (TOutIt avOut, int avNumber) noexcept

  This function sends a 2-digit zero padded number to an output iterator.
- static TOutIt smvPutNumber2Blank (TOutIt avOut, int avNumber) noexcept

  This function sends a 2-digit blank padded number to an output iterator.
- static TOutIt smvPutNumber3Zero (TOutIt avOut, int avNumber) noexcept

  This function sends a 3-digit zero padded number to an output iterator.
- static TOutIt smvPutNumber4Zero (TOutIt avOut, int avNumber) noexcept

  This function sends a 4-digit zero padded number to an output iterator.

6.16.1 Detailed Description

This class serves as the base class for time input/output facets.

6.16.2 Member Function Documentation

6.16.2.1 std::string lfc1::datetime::CTimeBase::smvExpandFormat (std::string *avFormat*) [static], [protected]

This function expands the time format by replacing format specifiers with their equivalent format specifiers.

Parameters

in	*avFormat*	The time format to be expanded.

Returns

The expanded format string.

6.16.2.2 CTimeBase::TInIt lfc1::datetime::CTimeBase::smvGetAmPm (TInIt *avNext*, TInIt *avEnd*, std::ios_base & *arIosBase*, std::ios_base::iostate & *arIoState*, bool & *arHasAmPm*, bool & *arIsPm*) [static], [protected]

This function parses an input iterator for an AM/PM indicator.

Parameters

in	*avNext*	The input iterator.
in	*avEnd*	The input end iterator.
in	*arIosBase*	The stream formatting information.
out	*arIoState*	The stream state.
out	*arHasAmPm*	The storage for the resulting has AM/PM indicator (true) or not (false).
out	*arIsPm*	The storage for the resulting PM (true) or AM (false) indicator.

Returns

The input iterator.

6.16.2.3 CFacetBase::TInIt lfc1::datetime::CFacetBase::smvGetNumber (TInIt *avNext,* TInIt *avEnd,* std::ios_base & *arIosBase,* std::ios_base::iostate & *arIoState,* int * *apNumber,* int *avDigits,* char *avPadding*) [static], [protected], [noexcept], [inherited]

This function parses an input iterator for an integer value.

Parameters

in	*avNext*	The input iterator.
in	*avEnd*	The input end iterator.
in	*arIosBase*	The stream formatting information.
out	*arIoState*	The stream state.
out	*apNumber*	The storage for the resulting integer.
in	*avDigits*	The maximum number of digits allowed for the integer.
in	*avPadding*	The padding character allowed for the integer value.

Return values

ios_base::goodbit	A valid integer was found.
ios_base::failbit	A valid integer was not found.

6.16.2.4 CFacetBase::TInIt lfc1::datetime::CFacetBase::smvIgnoreChar (TInIt *avNext,* TInIt *avEnd,* std::ios_base::iostate & *arIoState,* char *avExpectedChar*) [static], [protected], [noexcept], [inherited]

This function parses an input iterator for a specific character. This function moves the input iterator forward one character if the expected character is found.

Parameters

in	*avNext*	The input iterator.
in	*avEnd*	The input end iterator.
out	*arIoState*	The stream state.
in	*avExpectedChar*	The next character expected from the input iterator.

Return values

ios_base::goodbit	The expected character was found.
ios_base::failbit	The expected character was not found.

6.16.2.5 CTimeBase::TOutIt lfc1::datetime::CTimeBase::smvPutAmPm (TOutIt *avOut,* int *avHour*) [static], [protected]

This function sends AM or PM to an output iterator based on the given hour.

Parameters

in	*avOut*	The output iterator.
in	*avHour*	The given hour.

Returns

The output iterator.

6.16.2.6 CFacetBase::TOutIt lfc1::datetime::CFacetBase::smvPutNumber1 (TOutIt *avOut,* int *avNumber*) [static], [protected], [noexcept], [inherited]

This function sends a 1-digit number to an output iterator.

Parameters

in	*avOut*	The output iterator.
in	*avNumber*	The 1-digit number.

Returns

The output iterator.

6.16.2.7 CFacetBase::TOutIt lfc1::datetime::CFacetBase::smvPutNumber2Blank (TOutIt *avOut,* int *avNumber*) [static], [protected], [noexcept], [inherited]

This function sends a 2-digit blank padded number to an output iterator.

Parameters

in	*avOut*	The output iterator.
in	*avNumber*	The 2-digit number.

Returns

The output iterator.

6.16.2.8 CFacetBase::TOutIt lfc1::datetime::CFacetBase::smvPutNumber2Zero (TOutIt *avOut,* int *avNumber*) [static], [protected], [noexcept], [inherited]

This function sends a 2-digit zero padded number to an output iterator.

Parameters

in	*avOut*	The output iterator.
in	*avNumber*	The 2-digit number.

Returns

The output iterator.

6.16.2.9 CFacetBase::TOutIt lfc1::datetime::CFacetBase::smvPutNumber3Zero (TOutIt *avOut,* int *avNumber*) [static], [protected], [noexcept], [inherited]

This function sends a 3-digit zero padded number to an output iterator.

Parameters

in	*avOut*	The output iterator.
in	*avNumber*	The 3-digit number.

Returns

The output iterator.

6.16.2.10 CFacetBase::TOutIt lfc1::datetime::CFacetBase::smvPutNumber4Zero (TOutIt *avOut*, int *avNumber*) [static],[protected],[noexcept],[inherited]

This function sends a 4-digit zero padded number to an output iterator.

Parameters

in	*avOut*	The output iterator.
in	*avNumber*	The 4-digit number.

Returns

The output iterator.

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

- include/lfc1/datetime/ctime.hpp
- datetime/library/src/ctimebase.cpp

6.17 lfc1::datetime::CTimeDuration Class Reference

This class represents time durations.

`#include <lfc1/datetime/ctime.hpp>`

Public Types

- enum EUnits {
  E_U_HOUR,
  E_U_MINUTE,
  E_U_SECOND,
  E_U_MILLISECOND }
  < *The list of time durations.*

Public Member Functions

- CTimeDuration (int avDuration, EUnits avUnit=E_U_MILLISECOND) noexcept
  This function creates a CTimeDuration object.
- CTimeDuration (const CTimeDuration &arRHS) noexcept
  This function copy constructs a CTimeDuration object.
- CTimeDuration (CTimeDuration &&arRHS) noexcept
  This function move constructs a CTimeDuration object.
- ~CTimeDuration () noexcept
  This function destroys a CTimeDuration object.
- CTimeDuration & operator= (const CTimeDuration &arRHS) noexcept
  This function assigns a CTimeDuration object to another CTimeDuration object.
- CTimeDuration & operator= (CTimeDuration &&arRHS) noexcept
  This function moves a CTimeDuration object to another CTimeDuration object.

- int mvGetDuration () const noexcept

  This function obtains the time duration.
- EUnits mvGetUnit () const noexcept

  This function obtains the unit of the time duration.
- void swap (CTimeDuration &arRHS) noexcept

  This function swaps a CTimeDuration object with another CTimeDuration object.

6.17.1 Detailed Description

This class represents time durations.

The purpose of this class is to support the increment and decrement operators of the CTime class.

6.17.2 Constructor & Destructor Documentation

6.17.2.1 lfc1::datetime::CTimeDuration::CTimeDuration (int *avDuration*, EUnits *avUnit* = E_U_MILLISECOND) [noexcept]

This function creates a CTimeDuration object.

Parameters

in	*avDuration*	The time duration.
in	*avUnit*	The unit of measure of the time duration.

6.17.2.2 lfc1::datetime::CTimeDuration::CTimeDuration (const CTimeDuration & *arRHS*) [noexcept]

This function copy constructs a CTimeDuration object.

Parameters

in	*arRHS*	The CTimeDuration object to be copied.

6.17.2.3 lfc1::datetime::CTimeDuration::CTimeDuration (CTimeDuration && *arRHS*) [noexcept]

This function move constructs a CTimeDuration object.

Parameters

in	*arRHS*	The CTimeDuration object to be moved.

6.17.3 Member Function Documentation

6.17.3.1 int lfc1::datetime::CTimeDuration::mvGetDuration ( ) const [noexcept]

This function obtains the time duration.

Returns

The time duration.

6.17.3.2 CTimeDuration::EUnits lfc1::datetime::CTimeDuration::mvGetUnit ( ) const [noexcept]

This function obtains the unit of the time duration.

Returns

The unit of the time duration.

6.17.3.3 CTimeDuration & lfc1::datetime::CTimeDuration::operator= (const CTimeDuration & *arRHS*) [noexcept]

This function assigns a CTimeDuration object to another CTimeDuration object.

Parameters

in	*arRHS*	The CTimeDuration object to assign to another CTimeDuration object.

Returns

The CTimeDuration object assigned to.

6.17.3.4 CTimeDuration & lfc1::datetime::CTimeDuration::operator= (CTimeDuration && *arRHS*) [noexcept]

This function moves a CTimeDuration object to another CTimeDuration object.

Parameters

in	*arRHS*	The CTimeDuration object to move to another CTimeDuration object.

Returns

The CTimeDuration object assigned to.

6.17.3.5 void lfc1::datetime::CTimeDuration::swap (CTimeDuration & *arRHS*) [noexcept]

This function swaps a CTimeDuration object with another CTimeDuration object.

Parameters

in,out	*arRHS*	The CTimeDuration object to swap with.

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

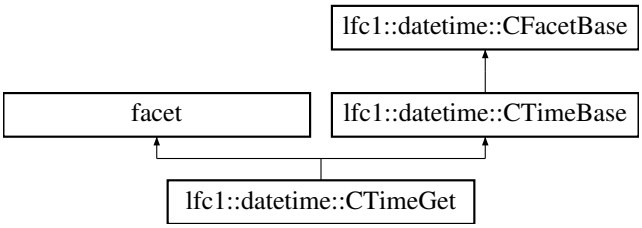
- include/lfc1/datetime/ctime.hpp
- datetime/library/src/ctimeduration.cpp

6.18 lfc1::datetime::CTimeGet Class Reference

This class is a CTime class input facet.

`#include <lfc1/datetime/ctime.hpp>`

Inheritance diagram for lfc1::datetime::CTimeGet:



Public Types

- typedef
  std::istreambuf_iterator< char > TInIt

  This type is the input iterator.
- typedef
  std::ostreambuf_iterator< char > TOutIt

  This type is the output iterator.

Public Member Functions

- CTimeGet (size_t avRelease=0)

  This function creates a CTimeGet object.
- CTimeGet (const CTimeGet &)=delete

  Not supported.
- CTimeGet (CTimeGet &&)=delete

  Not supported.
- ~CTimeGet () noexceptfinal

  This function destroys a CTimeGet object.
- CTimeGet & operator= (const CTimeGet &)=delete

  Not supported.
- CTimeGet & operator= (CTimeGet &&)=delete

  Not supported.
- TInIt mvGet (TInIt avNext, TInIt avEnd, std::ios_base &arIosBase, std::ios_base::iostate &arIoState, CTime &arTime, const std::string &arFormat) const

  This function inputs a time from a stream. Duplication of time parts is not allowed.

Static Public Attributes

- static std::locale::id id

  The facet ID.

Protected Member Functions

- virtual TInIt mvDoGet (TInIt avNext, TInIt avEnd, std::ios_base &arIosBase, std::ios_base::iostate &arIoState, CTime &arTime, const std::string &arFormat) const

  This function implements the behavior of the mvGet() function.

Static Protected Member Functions

- static std::string smvExpandFormat (std::string avFormat)

  This function expands the time format by replacing format specifiers with their equivalent format specifiers.
- static TInIt smvGetAmPm (TInIt avNext, TInIt avEnd, std::ios_base &arIosBase, std::ios_base::iostate &arIoState, bool &arHasAmPm, bool &arIsPm)

  This function parses an input iterator for an AM/PM indicator.
- static TOutIt smvPutAmPm (TOutIt avOut, int avHour)

  This function sends AM or PM to an output iterator based on the given hour.
- static TInIt smvIgnoreChar (TInIt avNext, TInIt avEnd, std::ios_base::iostate &arIoState, char avExpectedChar) noexcept

  This function parses an input iterator for a specific character. This function moves the input iterator forward one character if the expected character is found.
- static TInIt smvGetNumber (TInIt avNext, TInIt avEnd, std::ios_base &arIosBase, std::ios_base::iostate &arIoState, int *apNumber, int avDigits, char avPadding) noexcept

  This function parses an input iterator for an integer value.
- static TOutIt smvPutNumber1 (TOutIt avOut, int avNumber) noexcept

  This function sends a 1-digit number to an output iterator.
- static TOutIt smvPutNumber2Zero (TOutIt avOut, int avNumber) noexcept

  This function sends a 2-digit zero padded number to an output iterator.
- static TOutIt smvPutNumber2Blank (TOutIt avOut, int avNumber) noexcept

  This function sends a 2-digit blank padded number to an output iterator.
- static TOutIt smvPutNumber3Zero (TOutIt avOut, int avNumber) noexcept

  This function sends a 3-digit zero padded number to an output iterator.
- static TOutIt smvPutNumber4Zero (TOutIt avOut, int avNumber) noexcept

  This function sends a 4-digit zero padded number to an output iterator.

6.18.1 Detailed Description

This class is a CTime class input facet.

Note

This class uses the Template Method design pattern.

6.18.2 Constructor & Destructor Documentation

6.18.2.1 lfc1::datetime::CTimeGet::CTimeGet (size_t *avRelease* = `0`) [explicit]

This function creates a CTimeGet object.

Parameters

in	*avRelease*	Indicates who controls the lifetime of the facet. (0 means locale)

6.18.3 Member Function Documentation

6.18.3.1 CTimeGet::TInIt lfc1::datetime::CTimeGet::mvDoGet (TInIt *avNext*, TInIt *avEnd*, std::ios_base & *arIosBase*, std::ios_base::iostate & *arIoState*, CTime & *arTime*, const std::string & *arFormat*) const [protected], [virtual]

This function implements the behavior of the mvGet() function.

Parameters

in	*avNext*	The input iterator.
in	*avEnd*	The input end iterator.
in	*arIosBase*	The stream formatting information.
out	*arIoState*	The stream state.
out	*arTime*	The time retrieved from the input stream.
in	*arFormat*	The format of the time retrieved from the input stream.

Returns

The input iterator.

6.18.3.2 CTimeGet::TInIt lfc1::datetime::CTimeGet::mvGet (TInIt *avNext*, TInIt *avEnd*, std::ios_base & *arIosBase*, std::ios_base::iostate & *arIoState*, CTime & *arTime*, const std::string & *arFormat*) const

This function inputs a time from a stream. Duplication of time parts is not allowed.

This function recognizes the following format specifiers which are identical to the time specific format specifiers of the C language strftime() function except for N:

```
%% - A percent character.
%H - Hour in 24-hour format [00-23] zero padded.
%I - Hour in 12-hour format [01-12] zero padded.
%k - Hour in 24-hour format [00-23] blank padded.
%l - Hour in 12-hour format [01-12] blank padded.
%M - Minute [00-59].
%n - Newline character.
%N - Millisecond [000-999].
%p - AM/PM designation.
%r - Default 12-hour time format.
%R - Equivalent to %H:%M.
%S - Second [00-59].
%t - Horizontal tab character.
%T - Equivalent to %H:%M:%S.
%X - Default time format.
```

Any other sequence of characters not included in the above list will be taken literally.

The following combinations of format specifiers results into a valid CTime object:

```
%R
%T
%X
(%H or %I or %k or %l) [and (%M) [and (%S) [and (%N)]]]
```

Parameters

in	*avNext*	The input stream iterator.
in	*avEnd*	The input stream end iterator.
in	*arIosBase*	The stream formatting information.
out	*arIoState*	The stream state. Set to ios_base::failbit if parsing failed.
out	*arTime*	The time retrieved from the input stream.
in	*arFormat*	The format of the time retrieved from the input stream.

Returns

The input iterator.

6.18.3.3 std::string lfc1::datetime::CTimeBase::smvExpandFormat (std::string *avFormat*) [static], [protected], [inherited]

This function expands the time format by replacing format specifiers with their equivalent format specifiers.

Parameters

in	*avFormat*	The time format to be expanded.

Returns

The expanded format string.

6.18.3.4 CTimeBase::TInIt lfc1::datetime::CTimeBase::smvGetAmPm (TInIt *avNext,* TInIt *avEnd,* std::ios_base & *arIosBase,* std::ios_base::iostate & *arIoState,* bool & *arHasAmPm,* bool & *arIsPm*) [static], [protected], [inherited]

This function parses an input iterator for an AM/PM indicator.

Parameters

in	*avNext*	The input iterator.
in	*avEnd*	The input end iterator.
in	*arIosBase*	The stream formatting information.
out	*arIoState*	The stream state.
out	*arHasAmPm*	The storage for the resulting has AM/PM indicator (true) or not (false).
out	*arIsPm*	The storage for the resulting PM (true) or AM (false) indicator.

Returns

The input iterator.

6.18.3.5 CFacetBase::TInIt lfc1::datetime::CFacetBase::smvGetNumber (TInIt *avNext,* TInIt *avEnd,* std::ios_base & *arIosBase,* std::ios_base::iostate & *arIoState,* int * *apNumber,* int *avDigits,* char *avPadding*) [static], [protected], [noexcept], [inherited]

This function parses an input iterator for an integer value.

Parameters

in	*avNext*	The input iterator.
in	*avEnd*	The input end iterator.
in	*arIosBase*	The stream formatting information.
out	*arIoState*	The stream state.
out	*apNumber*	The storage for the resulting integer.
in	*avDigits*	The maximum number of digits allowed for the integer.
in	*avPadding*	The padding character allowed for the integer value.

Return values

ios_base::goodbit	A valid integer was found.
ios_base::failbit	A valid integer was not found.

6.18.3.6 CFacetBase::TInIt lfc1::datetime::CFacetBase::smvIgnoreChar (TInIt *avNext,* TInIt *avEnd,* std::ios_base::iostate & *arIoState,* char *avExpectedChar*) [static],[protected],[noexcept],[inherited]

This function parses an input iterator for a specific character. This function moves the input iterator forward one character if the expected character is found.

Parameters

in	*avNext*	The input iterator.
in	*avEnd*	The input end iterator.
out	*arIoState*	The stream state.
in	*avExpectedChar*	The next character expected from the input iterator.

Return values

ios_base::goodbit	The expected character was found.
ios_base::failbit	The expected character was not found.

6.18.3.7 CTimeBase::TOutIt lfc1::datetime::CTimeBase::smvPutAmPm (TOutIt *avOut,* int *avHour*) [static],[protected],[inherited]

This function sends AM or PM to an output iterator based on the given hour.

Parameters

in	*avOut*	The output iterator.
in	*avHour*	The given hour.

Returns

The output iterator.

6.18.3.8 CFacetBase::TOutIt lfc1::datetime::CFacetBase::smvPutNumber1 (TOutIt *avOut,* int *avNumber*) [static],[protected],[noexcept],[inherited]

This function sends a 1-digit number to an output iterator.

Parameters

in	*avOut*	The output iterator.
in	*avNumber*	The 1-digit number.

Returns

The output iterator.

6.18.3.9 CFacetBase::TOutIt lfc1::datetime::CFacetBase::smvPutNumber2Blank (TOutIt *avOut,* int *avNumber*) [static],[protected],[noexcept],[inherited]

This function sends a 2-digit blank padded number to an output iterator.

Parameters

in	*avOut*	The output iterator.
in	*avNumber*	The 2-digit number.

Returns

The output iterator.

6.18.3.10 CFacetBase::TOutIt lfc1::datetime::CFacetBase::smvPutNumber2Zero (TOutIt *avOut,* int *avNumber*) [static],[protected],[noexcept],[inherited]

This function sends a 2-digit zero padded number to an output iterator.

Parameters

in	*avOut*	The output iterator.
in	*avNumber*	The 2-digit number.

Returns

The output iterator.

6.18.3.11 CFacetBase::TOutIt lfc1::datetime::CFacetBase::smvPutNumber3Zero (TOutIt *avOut,* int *avNumber*) [static],[protected],[noexcept],[inherited]

This function sends a 3-digit zero padded number to an output iterator.

Parameters

in	*avOut*	The output iterator.
in	*avNumber*	The 3-digit number.

Returns

The output iterator.

6.18.3.12 CFacetBase::TOutIt lfc1::datetime::CFacetBase::smvPutNumber4Zero (TOutIt *avOut,* int *avNumber*) [static],[protected],[noexcept],[inherited]

This function sends a 4-digit zero padded number to an output iterator.

Parameters

in	*avOut*	The output iterator.
in	*avNumber*	The 4-digit number.

Returns

The output iterator.

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

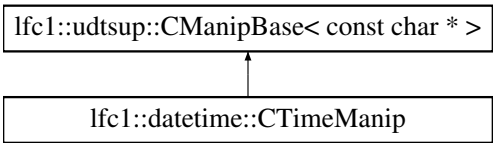
- include/lfc1/datetime/ctime.hpp
- datetime/library/src/ctimeget.cpp

6.19 lfc1::datetime::CTimeManip Class Reference

This class is a helper class for the time manipulator. This class was derived from Sections 3.3.1.3 to 3.3.1.7 of the book Standard C++ IOStreams and Locales.

`#include <lfc1/datetime/ctime.hpp>`

Inheritance diagram for lfc1::datetime::CTimeManip:



Public Member Functions

- CTimeManip (const char *apFormat)

  This function creates a CTimeManip object.
- CTimeManip (const CTimeManip &)=default

  Uses default implementation.
- CTimeManip & operator= (const CTimeManip &)=default

  Uses default implementation.

Static Public Member Functions

- static std::string smvGetFormat (std::ios_base &arIosBase)

  This function obtains the time format from a stream.
- static std::ios_base::iostate smvCopyFmtErr (std::basic_ios< char > &arIos)

  This function obtains the error information relating to time format copying.

6.19.1 Detailed Description

This class is a helper class for the time manipulator. This class was derived from Sections 3.3.1.3 to 3.3.1.7 of the book Standard C++ IOStreams and Locales.

6.19.2 Constructor & Destructor Documentation

6.19.2.1 lfc1::datetime::CTimeManip::CTimeManip (const char * *apFormat*)

This function creates a CTimeManip object.

Parameters

in	*apFormat*	The desired time format.

6.19.3 Member Function Documentation

6.19.3.1 std::ios_base::iostate lfc1::datetime::CTimeManip::smvCopyFmtErr (std::basic_ios< char > & *arlos*) [static]

This function obtains the error information relating to time format copying.

Parameters

in	*arlos*	The stream containing the error information.

Returns

The error information relating to time format copying.

6.19.3.2 std::string lfc1::datetime::CTimeManip::smvGetFormat (std::ios_base & *arlosBase*) [static]

This function obtains the time format from a stream.

Parameters

in	*arlosBase*	The stream containing the time format.

Returns

The time format.

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

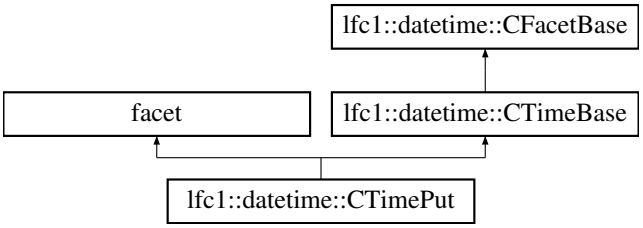
- include/lfc1/datetime/ctime.hpp
- datetime/library/src/ctimemanip.cpp

6.20 lfc1::datetime::CTimePut Class Reference

This class is a CTime class output facet.

`#include <lfc1/datetime/ctime.hpp>`

Inheritance diagram for lfc1::datetime::CTimePut:



Public Types

- typedef
  std::istreambuf_iterator< char > TInIt

This type is the input iterator.

- typedef
  std::ostreambuf_iterator< char > TOutIt

  This type is the output iterator.

Public Member Functions

- CTimePut (size_t avRelease=0)

  This function creates a CTimePut object.
- CTimePut (const CTimePut &)=delete

  Not supported.
- CTimePut (CTimePut &&)=delete

  Not supported.
- ~CTimePut () noexceptfinal

  This function destroys a CTimePut object.
- CTimePut & operator= (const CTimePut &)=delete

  Not supported.
- CTimePut & operator= (CTimePut &&)=delete

  Not supported.
- TOutIt mvPut (TOutIt avOut, const CTime &arTime, const std::string &arFormat) const

  This function outputs a time to a stream.

Static Public Attributes

- static std::locale::id id

  The facet ID.

Protected Member Functions

- virtual TOutIt mvDoPut (TOutIt avOut, const CTime &arTime, const std::string &arFormat) const

  This function implements the behavior of the mvPut() function.

Static Protected Member Functions

- static std::string smvExpandFormat (std::string avFormat)

  This function expands the time format by replacing format specifiers with their equivalent format specifiers.
- static TInIt smvGetAmPm (TInIt avNext, TInIt avEnd, std::ios_base &arIosBase, std::ios_base::iostate &arIoState, bool &arHasAmPm, bool &arIsPm)

  This function parses an input iterator for an AM/PM indicator.
- static TOutIt smvPutAmPm (TOutIt avOut, int avHour)

  This function sends AM or PM to an output iterator based on the given hour.
- static TInIt smvIgnoreChar (TInIt avNext, TInIt avEnd, std::ios_base::iostate &arIoState, char avExpectedChar) noexcept

  This function parses an input iterator for a specific character. This function moves the input iterator forward one character if the expected character is found.
- static TInIt smvGetNumber (TInIt avNext, TInIt avEnd, std::ios_base &arIosBase, std::ios_base::iostate &arIoState, int *apNumber, int avDigits, char avPadding) noexcept

  This function parses an input iterator for an integer value.
- static TOutIt smvPutNumber1 (TOutIt avOut, int avNumber) noexcept

  This function sends a 1-digit number to an output iterator.

- static TOutIt smvPutNumber2Zero (TOutIt avOut, int avNumber) noexcept

  This function sends a 2-digit zero padded number to an output iterator.
- static TOutIt smvPutNumber2Blank (TOutIt avOut, int avNumber) noexcept

  This function sends a 2-digit blank padded number to an output iterator.
- static TOutIt smvPutNumber3Zero (TOutIt avOut, int avNumber) noexcept

  This function sends a 3-digit zero padded number to an output iterator.
- static TOutIt smvPutNumber4Zero (TOutIt avOut, int avNumber) noexcept

  This function sends a 4-digit zero padded number to an output iterator.

6.20.1 Detailed Description

This class is a CTime class output facet.

Note

This class uses the Template Method design pattern.

6.20.2 Constructor & Destructor Documentation

6.20.2.1 lfc1::datetime::CTimePut::CTimePut (size_t *avRelease* = `0`) `[explicit]`

This function creates a CTimePut object.

Parameters

in	*avRelease*	Indicates who controls the lifetime of the facet. (0 means locale)

6.20.3 Member Function Documentation

6.20.3.1 CTimePut::TOutIt lfc1::datetime::CTimePut::mvDoPut (TOutIt *avOut*, const CTime & *arTime*, const std::string & *arFormat*) const `[protected]`, `[virtual]`

This function implements the behavior of the mvPut() function.

Parameters

in	*avOut*	The output iterator.
in	*arTime*	The time to be sent to the output stream.
in	*arFormat*	The format of the time to be sent to the output stream.

Returns

The output iterator.

6.20.3.2 CTimePut::TOutIt lfc1::datetime::CTimePut::mvPut (TOutIt *avOut*, const CTime & *arTime*, const std::string & *arFormat*) const

This function outputs a time to a stream.

This function recognizes the following format specifiers which are identical to the time specific format specifiers of the C language strftime() function except for N:

```
%% - A percent character.
%H - Hour in 24-hour format [00-23] zero padded.
%I - Hour in 12-hour format [01-12] zero padded.
```

```
%k - Hour in 24-hour format [00-23] blank padded.
%l - Hour in 12-hour format [01-12] blank padded.
%M - Minute [00-59].
%n - Newline character.
%N - Millisecond [000-999].
%p - AM/PM designation.
%r - Default 12-hour time format.
%R - Equivalent to %H:%M.
%S - Second [00-59].
%t - Horizontal tab character.
%T - Equivalent to %H:%M:%S.
%X - Default time format.
```

Any other sequence of characters not included in the above list will be taken literally.

Parameters

in	*avOut*	The output iterator.
in	*arTime*	The time to be sent to the output stream.
in	*arFormat*	The format of the time to be sent to the output stream.

Returns

The output iterator.

6.20.3.3 std::string lfc1::datetime::CTimeBase::smvExpandFormat (std::string *avFormat*) [static],[protected],[inherited]

This function expands the time format by replacing format specifiers with their equivalent format specifiers.

Parameters

in	*avFormat*	The time format to be expanded.

Returns

The expanded format string.

6.20.3.4 CTimeBase::TInIt lfc1::datetime::CTimeBase::smvGetAmPm (TInIt *avNext,* TInIt *avEnd,* std::ios_base & *arIosBase,* std::ios_base::iostate & *arIoState,* bool & *arHasAmPm,* bool & *arIsPm*) [static],[protected],[inherited]

This function parses an input iterator for an AM/PM indicator.

Parameters

in	*avNext*	The input iterator.
in	*avEnd*	The input end iterator.
in	*arIosBase*	The stream formatting information.
out	*arIoState*	The stream state.
out	*arHasAmPm*	The storage for the resulting has AM/PM indicator (true) or not (false).
out	*arIsPm*	The storage for the resulting PM (true) or AM (false) indicator.

Returns

The input iterator.

6.20.3.5 CFacetBase::TInIt lfc1::datetime::CFacetBase::smvGetNumber (TInIt *avNext,* TInIt *avEnd,* std::ios_base & *arIosBase,* std::ios_base::iostate & *arIoState,* int * *apNumber,* int *avDigits,* char *avPadding*) [static], [protected], [noexcept], [inherited]

This function parses an input iterator for an integer value.

Parameters

in	*avNext*	The input iterator.
in	*avEnd*	The input end iterator.
in	*arIosBase*	The stream formatting information.
out	*arIoState*	The stream state.
out	*apNumber*	The storage for the resulting integer.
in	*avDigits*	The maximum number of digits allowed for the integer.
in	*avPadding*	The padding character allowed for the integer value.

Return values

ios_base::goodbit	A valid integer was found.
ios_base::failbit	A valid integer was not found.

6.20.3.6 CFacetBase::TInIt lfc1::datetime::CFacetBase::smvIgnoreChar (TInIt *avNext,* TInIt *avEnd,* std::ios_base::iostate & *arIoState,* char *avExpectedChar*) [static], [protected], [noexcept], [inherited]

This function parses an input iterator for a specific character. This function moves the input iterator forward one character if the expected character is found.

Parameters

in	*avNext*	The input iterator.
in	*avEnd*	The input end iterator.
out	*arIoState*	The stream state.
in	*avExpectedChar*	The next character expected from the input iterator.

Return values

ios_base::goodbit	The expected character was found.
ios_base::failbit	The expected character was not found.

6.20.3.7 CTimeBase::TOutIt lfc1::datetime::CTimeBase::smvPutAmPm (TOutIt *avOut,* int *avHour*) [static], [protected], [inherited]

This function sends AM or PM to an output iterator based on the given hour.

Parameters

in	*avOut*	The output iterator.
in	*avHour*	The given hour.

Returns

The output iterator.

6.20.3.8 CFacetBase::TOutIt lfc1::datetime::CFacetBase::smvPutNumber1 (TOutIt *avOut,* int *avNumber*) [static], [protected], [noexcept], [inherited]

This function sends a 1-digit number to an output iterator.

Parameters

in	*avOut*	The output iterator.
in	*avNumber*	The 1-digit number.

Returns

The output iterator.

6.20.3.9 CFacetBase::TOutIt lfc1::datetime::CFacetBase::smvPutNumber2Blank (TOutIt *avOut,* int *avNumber*) [static], [protected], [noexcept], [inherited]

This function sends a 2-digit blank padded number to an output iterator.

Parameters

in	*avOut*	The output iterator.
in	*avNumber*	The 2-digit number.

Returns

The output iterator.

6.20.3.10 CFacetBase::TOutIt lfc1::datetime::CFacetBase::smvPutNumber2Zero (TOutIt *avOut,* int *avNumber*) [static], [protected], [noexcept], [inherited]

This function sends a 2-digit zero padded number to an output iterator.

Parameters

in	*avOut*	The output iterator.
in	*avNumber*	The 2-digit number.

Returns

The output iterator.

6.20.3.11 CFacetBase::TOutIt lfc1::datetime::CFacetBase::smvPutNumber3Zero (TOutIt *avOut,* int *avNumber*) [static], [protected], [noexcept], [inherited]

This function sends a 3-digit zero padded number to an output iterator.

Parameters

in	*avOut*	The output iterator.
in	*avNumber*	The 3-digit number.

Returns

The output iterator.

6.20.3.12 CFacetBase::TOutIt lfc1::datetime::CFacetBase::smvPutNumber4Zero (TOutIt *avOut*, int *avNumber*) [static],[protected],[noexcept],[inherited]

This function sends a 4-digit zero padded number to an output iterator.

Parameters

in	*avOut*	The output iterator.
in	*avNumber*	The 4-digit number.

Returns

The output iterator.

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

- include/lfc1/datetime/ctime.hpp
- datetime/library/src/ctimeput.cpp

6.21 lfc1::datetime::CDate::SDate Struct Reference

A structure that holds the individual parts of a date.

`#include <include/lfc1/datetime/cdate.hpp>`

Public Attributes

- int mvYear

  The year.
- int mvMonth

  The month.
- int mvDay

  The day.

6.21.1 Detailed Description

A structure that holds the individual parts of a date.

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

- include/lfc1/datetime/cdate.hpp

6.22 lfc1::datetime::CDateTime::SDateTime Struct Reference

A structure that holds the individual parts of a date and time.

`#include <include/lfc1/datetime/cdatetime.hpp>`

Public Attributes

- int mvYear

  The year.
- int mvMonth

  The month.
- int mvDay

  The day.
- int mvHour

  The hour.
- int mvMinute

  The minute.
- int mvSecond

  The second.
- int mvMillisecond

  The millisecond.
- int mvDst

  The daylight savings indicator.

6.22.1 Detailed Description

A structure that holds the individual parts of a date and time.

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

- include/lfc1/datetime/cdatetime.hpp

6.23 lfc1::datetime::CTime::STime Struct Reference

A structure that holds the individual parts of a time.

`#include <include/lfc1/datetime/ctime.hpp>`

Public Attributes

- int mvHour

  The hour.
- int mvMinute

  The minute.
- int mvSecond

  The second.
- int mvMillisecond

  The millisecond.

6.23.1 Detailed Description

A structure that holds the individual parts of a time.

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

- include/lfc1/datetime/ctime.hpp

Index