

LFC1 Libraries

0.0.1

Generated by Doxygen 1.8.3.1

Thu May 9 2013 08:12:34

Contents

1	Main Page	1
2	Module Index	3
2.1	Modules	3
3	Hierarchical Index	5
3.1	Class Hierarchy	5
4	Class Index	7
4.1	Class List	7
5	Module Documentation	9
5.1	System class library	9
5.1.1	Detailed Description	9
5.1.2	Function Documentation	10
5.1.2.1	grCategory	10
5.2	User-defined type support class library	11
5.2.1	Detailed Description	11
5.2.2	Function Documentation	11
5.2.2.1	gvDoManip	11
5.2.2.2	gvInput	12
5.2.2.3	gvInputGuarded	12
5.2.2.4	gvOutput	13
5.2.2.5	gvOutputGuarded	13
5.2.2.6	operator<<	14
5.2.2.7	operator>>	14
5.3	Miscellaneous class library	15
5.3.1	Detailed Description	15
5.4	Numeric class library	16
5.4.1	Detailed Description	16
5.4.2	Function Documentation	17
5.4.2.1	swap	17
5.5	Filesystem class library	18

5.5.1	Detailed Description	18
5.5.2	Function Documentation	18
5.5.2.1	swap	18
5.5.2.2	swap	18
5.6	Date and time class library	19
5.6.1	Detailed Description	20
5.6.2	Function Documentation	20
5.6.2.1	operator<<	20
5.6.2.2	operator<<	21
5.6.2.3	operator<<	21
5.6.2.4	operator>>	21
5.6.2.5	operator>>	21
5.6.2.6	operator>>	22
5.6.2.7	setdateformat	22
5.6.2.8	setdatetimeformat	22
5.6.2.9	settimeformat	22
5.6.2.10	swap	23
5.6.2.11	swap	23
5.6.2.12	swap	23
5.6.2.13	swap	23
5.6.2.14	swap	23
5.6.2.15	swap	23
6	Class Documentation	25
6.1	lfc1::datetime::CDate Class Reference	25
6.1.1	Detailed Description	27
6.1.2	Constructor & Destructor Documentation	27
6.1.2.1	CDate	27
6.1.2.2	CDate	27
6.1.2.3	CDate	27
6.1.2.4	CDate	28
6.1.3	Member Function Documentation	28
6.1.3.1	mvGetDate	28
6.1.3.2	mvInput	28
6.1.3.3	mvOutput	28
6.1.3.4	mvSetDate	28
6.1.3.5	operator int	29
6.1.3.6	operator++	29
6.1.3.7	operator++	29
6.1.3.8	operator+=	29

6.1.3.9	operator--	30
6.1.3.10	operator--	30
6.1.3.11	operator-=	30
6.1.3.12	operator=	30
6.1.3.13	operator=	30
6.1.3.14	operator=	31
6.1.3.15	smvGetIndex	31
6.1.3.16	smvGetMaxDay	31
6.1.3.17	smvIsLeapYear	31
6.1.3.18	smvValidateDate	32
6.1.3.19	swap	32
6.1.4	Member Data Documentation	32
6.1.4.1	DATENO_OFFSET_SUN	32
6.1.4.2	FIRST_DAY_MON	32
6.1.4.3	FIRST_DAY_SUN	32
6.1.4.4	MIN_DATE_YEAR	32
6.2	Ifc1::datetime::CDateBase Class Reference	33
6.2.1	Detailed Description	34
6.2.2	Member Function Documentation	34
6.2.2.1	smvCalcWeekBasedDetails	34
6.2.2.2	smvCalcWeekNo	34
6.2.2.3	smvExpandFormat	35
6.2.2.4	smvGetMonthName	35
6.2.2.5	smvGetNumber	35
6.2.2.6	smvGetWeekDayName	36
6.2.2.7	smvIgnoreChar	36
6.2.2.8	smvPutMonthName	36
6.2.2.9	smvPutNumber1	37
6.2.2.10	smvPutNumber2Blank	37
6.2.2.11	smvPutNumber2Zero	37
6.2.2.12	smvPutNumber3Zero	37
6.2.2.13	smvPutNumber4Zero	38
6.2.2.14	smvPutWeekDayName	38
6.3	Ifc1::datetime::CDateDuration Class Reference	38
6.3.1	Detailed Description	39
6.3.2	Constructor & Destructor Documentation	39
6.3.2.1	CDateDuration	39
6.3.2.2	CDateDuration	39
6.3.2.3	CDateDuration	39
6.3.3	Member Function Documentation	40

6.3.3.1	mvGetDuration	40
6.3.3.2	mvGetUnit	40
6.3.3.3	operator=	40
6.3.3.4	operator=	40
6.3.3.5	swap	40
6.4	lfc1::datetime::CDateGet Class Reference	41
6.4.1	Detailed Description	42
6.4.2	Constructor & Destructor Documentation	43
6.4.2.1	CDateGet	43
6.4.3	Member Function Documentation	43
6.4.3.1	mvDoGet	43
6.4.3.2	mvGet	43
6.4.3.3	smvCalcWeekBasedDetails	44
6.4.3.4	smvCalcWeekNo	44
6.4.3.5	smvExpandFormat	44
6.4.3.6	smvGetMonthName	45
6.4.3.7	smvGetNumber	45
6.4.3.8	smvGetWeekDayName	45
6.4.3.9	smvIgnoreChar	46
6.4.3.10	smvPutMonthName	46
6.4.3.11	smvPutNumber1	46
6.4.3.12	smvPutNumber2Blank	47
6.4.3.13	smvPutNumber2Zero	47
6.4.3.14	smvPutNumber3Zero	47
6.4.3.15	smvPutNumber4Zero	47
6.4.3.16	smvPutWeekDayName	48
6.5	lfc1::datetime::CDateManip Class Reference	48
6.5.1	Detailed Description	49
6.5.2	Constructor & Destructor Documentation	49
6.5.2.1	CDateManip	49
6.5.3	Member Function Documentation	49
6.5.3.1	smvCopyFmtErr	49
6.5.3.2	smvGetFormat	49
6.6	lfc1::datetime::CDatePut Class Reference	50
6.6.1	Detailed Description	51
6.6.2	Constructor & Destructor Documentation	51
6.6.2.1	CDatePut	51
6.6.3	Member Function Documentation	52
6.6.3.1	mvDoPut	52
6.6.3.2	mvPut	52

6.6.3.3	smvCalcWeekBasedDetails	53
6.6.3.4	smvCalcWeekNo	53
6.6.3.5	smvExpandFormat	53
6.6.3.6	smvGetMonthName	53
6.6.3.7	smvGetNumber	54
6.6.3.8	smvGetWeekDayName	54
6.6.3.9	smvIgnoreChar	55
6.6.3.10	smvPutMonthName	55
6.6.3.11	smvPutNumber1	55
6.6.3.12	smvPutNumber2Blank	55
6.6.3.13	smvPutNumber2Zero	56
6.6.3.14	smvPutNumber3Zero	56
6.6.3.15	smvPutNumber4Zero	56
6.6.3.16	smvPutWeekDayName	57
6.7	lfc1::datetime::CDateTime Class Reference	57
6.7.1	Detailed Description	59
6.7.2	Constructor & Destructor Documentation	59
6.7.2.1	CDateTime	59
6.7.2.2	CDateTime	59
6.7.2.3	CDateTime	59
6.7.2.4	CDateTime	60
6.7.3	Member Function Documentation	60
6.7.3.1	mvGetDateTime	60
6.7.3.2	mvInput	60
6.7.3.3	mvOutput	60
6.7.3.4	mvSetDateTime	60
6.7.3.5	operator long long	61
6.7.3.6	operator++	61
6.7.3.7	operator++	61
6.7.3.8	operator+=	61
6.7.3.9	operator--	62
6.7.3.10	operator--	62
6.7.3.11	operator-=	62
6.7.3.12	operator=	62
6.7.3.13	operator=	63
6.7.3.14	operator=	63
6.7.3.15	smvGetIndex	63
6.7.3.16	smvValidateDst	63
6.7.3.17	swap	64
6.7.4	Member Data Documentation	64

6.7.4.1	MAX_DATETIMEO	64
6.7.4.2	MIN_DATETIMEO	64
6.8	lfc1::datetime::CDateTimeBase Class Reference	64
6.8.1	Detailed Description	66
6.8.2	Member Function Documentation	66
6.8.2.1	smvCalcWeekBasedDetails	66
6.8.2.2	smvCalcWeekNo	67
6.8.2.3	smvExpandFormat	67
6.8.2.4	smvGetAmPm	67
6.8.2.5	smvGetMonthName	68
6.8.2.6	smvGetNumber	68
6.8.2.7	smvGetNumber	68
6.8.2.8	smvGetTzName	69
6.8.2.9	smvGetTzOffset	69
6.8.2.10	smvGetWeekDayName	69
6.8.2.11	smvIgnoreChar	70
6.8.2.12	smvIgnoreChar	70
6.8.2.13	smvPutAmPm	70
6.8.2.14	smvPutMonthName	71
6.8.2.15	smvPutNumber1	71
6.8.2.16	smvPutNumber1	71
6.8.2.17	smvPutNumber2Blank	71
6.8.2.18	smvPutNumber2Blank	72
6.8.2.19	smvPutNumber2Zero	72
6.8.2.20	smvPutNumber2Zero	72
6.8.2.21	smvPutNumber3Zero	72
6.8.2.22	smvPutNumber3Zero	73
6.8.2.23	smvPutNumber4Zero	73
6.8.2.24	smvPutNumber4Zero	73
6.8.2.25	smvPutTzName	73
6.8.2.26	smvPutTzOffset	74
6.8.2.27	smvPutWeekDayName	74
6.9	lfc1::datetime::CDateTimeDuration Class Reference	74
6.9.1	Detailed Description	75
6.9.2	Constructor & Destructor Documentation	75
6.9.2.1	CDateTimeDuration	75
6.9.2.2	CDateTimeDuration	75
6.9.2.3	CDateTimeDuration	76
6.9.3	Member Function Documentation	76
6.9.3.1	mvGetDuration	76

6.9.3.2	mvGetUnit	76
6.9.3.3	operator=	76
6.9.3.4	operator=	76
6.9.3.5	swap	77
6.10	Ifc1::datetime::CDateTimeGet Class Reference	77
6.10.1	Detailed Description	79
6.10.2	Constructor & Destructor Documentation	80
6.10.2.1	CDateTimeGet	80
6.10.3	Member Function Documentation	80
6.10.3.1	mvDoGet	80
6.10.3.2	mvGet	80
6.10.3.3	smvCalcWeekBasedDetails	81
6.10.3.4	smvCalcWeekNo	82
6.10.3.5	smvExpandFormat	82
6.10.3.6	smvGetAmPm	82
6.10.3.7	smvGetMonthName	82
6.10.3.8	smvGetNumber	83
6.10.3.9	smvGetNumber	83
6.10.3.10	smvGetTzName	84
6.10.3.11	smvGetTzOffset	84
6.10.3.12	smvGetWeekDayName	84
6.10.3.13	smvIgnoreChar	85
6.10.3.14	smvIgnoreChar	85
6.10.3.15	smvPutAmPm	85
6.10.3.16	smvPutMonthName	85
6.10.3.17	smvPutNumber1	86
6.10.3.18	smvPutNumber1	86
6.10.3.19	smvPutNumber2Blank	86
6.10.3.20	smvPutNumber2Blank	87
6.10.3.21	smvPutNumber2Zero	87
6.10.3.22	smvPutNumber2Zero	87
6.10.3.23	smvPutNumber3Zero	87
6.10.3.24	smvPutNumber3Zero	88
6.10.3.25	smvPutNumber4Zero	88
6.10.3.26	smvPutNumber4Zero	88
6.10.3.27	smvPutTzName	88
6.10.3.28	smvPutTzOffset	89
6.10.3.29	smvPutWeekDayName	89
6.11	Ifc1::datetime::CDateTimeManip Class Reference	89
6.11.1	Detailed Description	90

6.11.2	Constructor & Destructor Documentation	90
6.11.2.1	CDateTimeManip	90
6.11.3	Member Function Documentation	90
6.11.3.1	smvCopyFmtErr	90
6.11.3.2	smvGetFormat	91
6.12	lfc1::datetime::CDateTimePut Class Reference	91
6.12.1	Detailed Description	93
6.12.2	Constructor & Destructor Documentation	94
6.12.2.1	CDateTimePut	94
6.12.3	Member Function Documentation	94
6.12.3.1	mvDoPut	94
6.12.3.2	mvPut	94
6.12.3.3	smvCalcWeekBasedDetails	95
6.12.3.4	smvCalcWeekNo	95
6.12.3.5	smvExpandFormat	95
6.12.3.6	smvGetAmPm	96
6.12.3.7	smvGetMonthName	96
6.12.3.8	smvGetNumber	96
6.12.3.9	smvGetNumber	97
6.12.3.10	smvGetTzName	97
6.12.3.11	smvGetTzOffset	98
6.12.3.12	smvGetWeekDayName	98
6.12.3.13	smvIgnoreChar	98
6.12.3.14	smvIgnoreChar	99
6.12.3.15	smvPutAmPm	99
6.12.3.16	smvPutMonthName	99
6.12.3.17	smvPutNumber1	100
6.12.3.18	smvPutNumber1	100
6.12.3.19	smvPutNumber2Blank	100
6.12.3.20	smvPutNumber2Blank	100
6.12.3.21	smvPutNumber2Zero	101
6.12.3.22	smvPutNumber2Zero	101
6.12.3.23	smvPutNumber3Zero	101
6.12.3.24	smvPutNumber3Zero	101
6.12.3.25	smvPutNumber4Zero	102
6.12.3.26	smvPutNumber4Zero	102
6.12.3.27	smvPutTzName	102
6.12.3.28	smvPutTzOffset	102
6.12.3.29	smvPutWeekDayName	103
6.13	lfc1::misc::CDemangler Class Reference	103

6.13.1 Detailed Description	103
6.13.2 Constructor & Destructor Documentation	104
6.13.2.1 CDemangler	104
6.13.3 Member Function Documentation	104
6.13.3.1 operator const char *	104
6.14 Ifc1::filesystem::CDirectory Class Reference	104
6.14.1 Detailed Description	105
6.14.2 Constructor & Destructor Documentation	105
6.14.2.1 CDirectory	105
6.14.2.2 CDirectory	105
6.14.2.3 CDirectory	105
6.14.3 Member Function Documentation	105
6.14.3.1 begin	105
6.14.3.2 end	106
6.14.3.3 mvGetNumEntries	106
6.14.3.4 mvGetPath	106
6.14.3.5 mvRefresh	106
6.14.3.6 operator=	106
6.14.3.7 operator=	107
6.14.3.8 swap	107
6.15 Ifc1::system::CErrCategory Class Reference	107
6.15.1 Detailed Description	108
6.15.2 Member Function Documentation	108
6.15.2.1 message	108
6.15.2.2 name	108
6.15.2.3 smrGetErrCategory	109
6.16 Ifc1::datetime::CFacetBase Class Reference	109
6.16.1 Detailed Description	110
6.16.2 Member Function Documentation	110
6.16.2.1 smvGetNumber	110
6.16.2.2 smvIgnoreChar	110
6.16.2.3 smvPutNumber1	111
6.16.2.4 smvPutNumber2Blank	111
6.16.2.5 smvPutNumber2Zero	111
6.16.2.6 smvPutNumber3Zero	112
6.16.2.7 smvPutNumber4Zero	112
6.17 Ifc1::numeric::CInteger< IT > Class Template Reference	112
6.17.1 Detailed Description	113
6.17.2 Constructor & Destructor Documentation	114
6.17.2.1 CInteger	114

6.17.2.2	CInteger	114
6.17.2.3	CInteger	114
6.17.2.4	CInteger	114
6.17.2.5	~CInteger	114
6.17.3	Member Function Documentation	115
6.17.3.1	operator IT	115
6.17.3.2	operator&=	115
6.17.3.3	operator*=	115
6.17.3.4	operator++	116
6.17.3.5	operator++	116
6.17.3.6	operator+=	116
6.17.3.7	operator--	116
6.17.3.8	operator--	117
6.17.3.9	operator-=	117
6.17.3.10	operator/=	117
6.17.3.11	operator<<=	117
6.17.3.12	operator=	118
6.17.3.13	operator=	118
6.17.3.14	operator=	118
6.17.3.15	operator>>=	119
6.17.3.16	operator^=	119
6.17.3.17	operator =	119
6.17.3.18	swap	120
6.18	Ifc1::datetime::CLangInfo Class Reference	120
6.18.1	Detailed Description	121
6.18.2	Member Function Documentation	121
6.18.2.1	smvGetAMString	121
6.18.2.2	smvGetDateFormat	121
6.18.2.3	smvGetDateTimeFormat	121
6.18.2.4	smvGetLongDayName	121
6.18.2.5	smvGetLongMonthName	122
6.18.2.6	smvGetPMString	122
6.18.2.7	smvGetShortDayName	122
6.18.2.8	smvGetShortMonthName	123
6.18.2.9	smvGetTime12Format	123
6.18.2.10	smvGetTime24Format	123
6.19	Ifc1::udtsup::CManipBase< T1 > Class Template Reference	123
6.19.1	Detailed Description	124
6.19.2	Constructor & Destructor Documentation	124
6.19.2.1	CManipBase	124

6.19.2.2	~CManipBase	124
6.19.3	Friends And Related Function Documentation	124
6.19.3.1	gvDoManip	125
6.20	Ifc1::filesystem::CRecDirectory Class Reference	125
6.20.1	Detailed Description	126
6.20.2	Constructor & Destructor Documentation	126
6.20.2.1	CRecDirectory	126
6.20.2.2	CRecDirectory	126
6.20.2.3	CRecDirectory	126
6.20.3	Member Function Documentation	126
6.20.3.1	begin	126
6.20.3.2	end	127
6.20.3.3	mvGetLevelCount	127
6.20.3.4	mvGetNumEntries	127
6.20.3.5	mvGetPath	127
6.20.3.6	mvRefresh	127
6.20.3.7	operator=	128
6.20.3.8	operator=	128
6.20.3.9	swap	128
6.21	Ifc1::datetime::CTime Class Reference	128
6.21.1	Detailed Description	130
6.21.2	Constructor & Destructor Documentation	131
6.21.2.1	CTime	131
6.21.2.2	CTime	131
6.21.2.3	CTime	131
6.21.2.4	CTime	131
6.21.3	Member Function Documentation	131
6.21.3.1	mvGetTime	131
6.21.3.2	mvInput	132
6.21.3.3	mvOutput	132
6.21.3.4	mvSetTime	132
6.21.3.5	operator int	132
6.21.3.6	operator++	132
6.21.3.7	operator++	133
6.21.3.8	operator+=	133
6.21.3.9	operator--	133
6.21.3.10	operator--	133
6.21.3.11	operator-=	134
6.21.3.12	operator=	134
6.21.3.13	operator=	134

6.21.3.14	operator=	134
6.21.3.15	smvGetIndex	135
6.21.3.16	smvValidateTime	135
6.21.3.17	swap	135
6.22	lfc1::datetime::CTimeBase Class Reference	135
6.22.1	Detailed Description	137
6.22.2	Member Function Documentation	137
6.22.2.1	smvExpandFormat	137
6.22.2.2	smvGetAmPm	137
6.22.2.3	smvGetNumber	137
6.22.2.4	smvIgnoreChar	138
6.22.2.5	smvPutAmPm	138
6.22.2.6	smvPutNumber1	138
6.22.2.7	smvPutNumber2Blank	139
6.22.2.8	smvPutNumber2Zero	139
6.22.2.9	smvPutNumber3Zero	139
6.22.2.10	smvPutNumber4Zero	139
6.23	lfc1::datetime::CTimeDuration Class Reference	140
6.23.1	Detailed Description	140
6.23.2	Constructor & Destructor Documentation	141
6.23.2.1	CTimeDuration	141
6.23.2.2	CTimeDuration	141
6.23.2.3	CTimeDuration	141
6.23.3	Member Function Documentation	141
6.23.3.1	mvGetDuration	141
6.23.3.2	mvGetUnit	141
6.23.3.3	operator=	141
6.23.3.4	operator=	142
6.23.3.5	swap	142
6.24	lfc1::datetime::CTimeGet Class Reference	142
6.24.1	Detailed Description	144
6.24.2	Constructor & Destructor Documentation	144
6.24.2.1	CTimeGet	144
6.24.3	Member Function Documentation	144
6.24.3.1	mvDoGet	144
6.24.3.2	mvGet	144
6.24.3.3	smvExpandFormat	145
6.24.3.4	smvGetAmPm	145
6.24.3.5	smvGetNumber	146
6.24.3.6	smvIgnoreChar	146

6.24.3.7	smvPutAmPm	146
6.24.3.8	smvPutNumber1	147
6.24.3.9	smvPutNumber2Blank	147
6.24.3.10	smvPutNumber2Zero	147
6.24.3.11	smvPutNumber3Zero	147
6.24.3.12	smvPutNumber4Zero	148
6.25	lfc1::datetime::CTimeManip Class Reference	148
6.25.1	Detailed Description	149
6.25.2	Constructor & Destructor Documentation	149
6.25.2.1	CTimeManip	149
6.25.3	Member Function Documentation	149
6.25.3.1	smvCopyFmtErr	149
6.25.3.2	smvGetFormat	149
6.26	lfc1::datetime::CTimePut Class Reference	150
6.26.1	Detailed Description	151
6.26.2	Constructor & Destructor Documentation	151
6.26.2.1	CTimePut	151
6.26.3	Member Function Documentation	151
6.26.3.1	mvDoPut	151
6.26.3.2	mvPut	152
6.26.3.3	smvExpandFormat	152
6.26.3.4	smvGetAmPm	153
6.26.3.5	smvGetNumber	153
6.26.3.6	smvIgnoreChar	153
6.26.3.7	smvPutAmPm	154
6.26.3.8	smvPutNumber1	154
6.26.3.9	smvPutNumber2Blank	154
6.26.3.10	smvPutNumber2Zero	154
6.26.3.11	smvPutNumber3Zero	155
6.26.3.12	smvPutNumber4Zero	155
6.27	std::numeric_limits< lfc1::numeric::CInteger< IT > > Class Template Reference	155
6.27.1	Detailed Description	157
6.28	lfc1::datetime::CDate::SDate Struct Reference	157
6.28.1	Detailed Description	157
6.29	lfc1::datetime::CDateTime::SDateTime Struct Reference	158
6.29.1	Detailed Description	158
6.30	lfc1::datetime::CTime::STime Struct Reference	158
6.30.1	Detailed Description	159

Index**159**

Chapter 1

Main Page

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

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

Note

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

Chapter 2

Module Index

2.1 Modules

Here is a list of all modules:

System class library	9
User-defined type support class library	11
Miscellaneous class library	15
Numeric class library	16
Filesystem class library	18
Date and time class library	19

Chapter 3

Hierarchical Index

3.1 Class Hierarchy

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

lfc1::datetime::CDate	25
lfc1::datetime::CDateDuration	38
lfc1::datetime::CDateTime	57
lfc1::datetime::CDateTimeDuration	74
lfc1::misc::CDemangler	103
lfc1::filesystem::CDirectory	104
lfc1::datetime::CFacetBase	109
lfc1::datetime::CDateBase	33
lfc1::datetime::CDateGet	41
lfc1::datetime::CDatePut	50
lfc1::datetime::CDateTimeBase	64
lfc1::datetime::CDateTimeGet	77
lfc1::datetime::CDateTimePut	91
lfc1::datetime::CTimeBase	135
lfc1::datetime::CDateTimeBase	64
lfc1::datetime::CTimeGet	142
lfc1::datetime::CTimePut	150
lfc1::numeric::CInteger< IT >	112
lfc1::datetime::CLangInfo	120
lfc1::udtsup::CManipBase< T1 >	123
lfc1::udtsup::CManipBase< const char * >	123
lfc1::datetime::CDateManip	48
lfc1::datetime::CDateTimeManip	89
lfc1::datetime::CTimeManip	148
lfc1::filesystem::CRecDirectory	125
lfc1::datetime::CTime	128
lfc1::datetime::CTimeDuration	140
std::error_category	
lfc1::system::CErrorCategory	107
facet	
lfc1::datetime::CDateGet	41
lfc1::datetime::CDatePut	50
lfc1::datetime::CDateTimeGet	77
lfc1::datetime::CDateTimePut	91
lfc1::datetime::CTimeGet	142
lfc1::datetime::CTimePut	150
std::numeric_limits< lfc1::numeric::CInteger< IT > >	155

lfc1::datetime::CDate::SDate	157
lfc1::datetime::CDateTime::SDateTime	158
lfc1::datetime::CTime::STime	158

Chapter 4

Class Index

4.1 Class List

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

lfc1::datetime::CDate	This class represents a local date	25
lfc1::datetime::CDateBase	This class serves as the base class for date input/output facets	33
lfc1::datetime::CDateDuration	This class represents date durations	38
lfc1::datetime::CDateGet	This class is a CDate class input facet	41
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	48
lfc1::datetime::CDatePut	This class is a CDate class output facet	50
lfc1::datetime::CDateTime	This class represents the time since 01-Jan-1970	57
lfc1::datetime::CDateTimeBase	This class serves as the base class for date and time input/output facets	64
lfc1::datetime::CDateTimeDuration	This class represents date and time durations	74
lfc1::datetime::CDateTimeGet	This class is a CDateTime class input facet	77
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	89
lfc1::datetime::CDateTimePut	This class is a CDateTime class output facet	91
lfc1::misc::CDemangler	This class provides RAII to name demangling	103
lfc1::filesystem::CDirectory	This class wraps a directory iterator so that directory entries could be loaded and sorted	104
lfc1::system::CErrorCategory	This class is an error category for this library	107
lfc1::datetime::CFacetBase	This class serves as the base class for date and time input/output facets	109
lfc1::numeric::CInteger< IT >	This class template represents little endian integers on big endian operating systems or big endian integers on little endian operating systems	112

lfc1::datetime::CLangInfo	
This class obtains locale specific date and time strings	120
lfc1::udtsup::CManipBase< T1 >	
This class template provides exception/error handling for single argument manipulators of user-defined types	123
lfc1::filesystem::CRecDirectory	
This class wraps a recursive directory iterator so that directory entries could be loaded and sorted	125
lfc1::datetime::CTime	
This class represents the time since midnight	128
lfc1::datetime::CTimeBase	
This class serves as the base class for time input/output facets	135
lfc1::datetime::CTimeDuration	
This class represents time durations	140
lfc1::datetime::CTimeGet	
This class is a CTime class input facet	142
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	148
lfc1::datetime::CTimePut	
This class is a CTime class output facet	150
std::numeric_limits< lfc1::numeric::CInteger< IT > >	
This template class is a specialization for <code>std::numeric_limits</code> for <code>CInteger<IT></code>	155
lfc1::datetime::CDate::SDate	
A structure that holds the individual parts of a date	157
lfc1::datetime::CDateTime::SDateTime	
A structure that holds the individual parts of a date and time	158
lfc1::datetime::CTime::STime	
A structure that holds the individual parts of a time	158

Chapter 5

Module Documentation

5.1 System class library

Classes

- class `lfc1::system::CErrorCategory`

This class is an error category for this library.

Enumerations

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

This enumeration defines the error codes for this library.

Functions

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

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

5.1.1 Detailed Description

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

5.1.2 Function Documentation

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

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

Returns

A reference to the single error category object.

5.2 User-defined type support class library

Classes

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

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

Functions

- template<typename T2 >
void `lfc1::udtsup::gvDoManip` (std::basic_ios< char > &arIos, const CManipBase< T2 > &arManipBase)
This template function calls the user-defined manipulator function and handles exceptions from it.
- template<typename T3 >
std::istream & `lfc1::udtsup::operator>>` (std::istream &arStream, const CManipBase< T3 > &arManipBase)
This template function invokes a user-defined manipulator for an input stream.
- template<typename T3 >
std::ostream & `lfc1::udtsup::operator<<` (std::ostream &arStream, const CManipBase< T3 > &arManipBase)
This template function invokes a user-defined manipulator for an output stream.
- template<typename UDT >
std::istream & `lfc1::udtsup::gvInput` (std::istream &arStream, UDT &arUdt)
This template function is meant to be the only function called by the extractor of a user-defined type. This function provides exception/error handling for the extractor of a user-defined type.
- template<typename UDT >
std::istream & `lfc1::udtsup::gvInputGuarded` (std::istream &arStream, UDT &arUdt)
This template function is meant to be the only function called by the extractor of a user-defined type. This function provides exception/error handling for the extractor of a user-defined type.
- template<typename UDT >
std::ostream & `lfc1::udtsup::gvOutput` (std::ostream &arStream, const UDT &arUdt)
This template function is meant to be the only function called by the inserter of a user-defined type. This function provides exception/error handling for the extractor of a user-defined type.
- template<typename UDT >
std::ostream & `lfc1::udtsup::gvOutputGuarded` (std::ostream &arStream, const UDT &arUdt)
This template function is meant to be the only function called by the inserter of a user-defined type. This function provides exception/error handling for the extractor of a user-defined type.

5.2.1 Detailed Description

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

5.2.2 Function Documentation

- #### 5.2.2.1
- template<typename T2 > void `lfc1::udtsup::gvDoManip` (std::basic_ios< char > &arIos, const CManipBase< T2 > &arManipBase)

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

Template Parameters

<code>T2</code>	The user-defined manipulator function argument type.
-----------------	--

Parameters

<i>arIos</i>	The stream's state object.
<i>arManipBase</i>	The user-defined manipulator function container.

5.2.2.2 `template<typename UDT> std::istream& lfc1::udtsup::gvInput (std::istream & arStream, UDT & arUdt)`

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

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

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

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

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

Template Parameters

<i>UDT</i>	The user-defined type.
------------	------------------------

Parameters

in	<i>arStream</i>	The source stream.
out	<i>arUdt</i>	The user-defined type.

Returns

The source stream.

5.2.2.3 `template<typename UDT> std::istream& lfc1::udtsup::gvInputGuarded (std::istream & arStream, UDT & arUdt)`

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

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

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

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

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

Template Parameters

<i>UDT</i>	The user-defined type.
------------	------------------------

Parameters

in	<i>arStream</i>	The source stream.
out	<i>arUdt</i>	The user-defined type.

Returns

The source stream.

5.2.2.4 `template<typename UDT> std::ostream& lfc1::udtsup::gvOutput (std::ostream & arStream, const UDT & arUdt)`

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

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

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

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

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

Template Parameters

<i>UDT</i>	The user-defined type.
------------	------------------------

Parameters

in	<i>arStream</i>	The destination stream.
out	<i>arUdt</i>	The user-defined type.

Returns

The destination stream.

5.2.2.5 `template<typename UDT> std::ostream& lfc1::udtsup::gvOutputGuarded (std::ostream & arStream, const UDT & arUdt)`

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

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

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

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

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

Template Parameters

<i>UDT</i>	The user-defined type.
------------	------------------------

Parameters

in	<i>arStream</i>	The destination stream.
out	<i>arUdt</i>	The user-defined type.

Returns

The destination stream.

5.2.2.6 `template<typename T3 > std::ostream& lfc1::udtsup::operator<< (std::ostream & arStream, const CManipBase< T3 > & arManipBase)`

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

Template Parameters

<i>T3</i>	The user-defined manipulator function argument type.
-----------	--

Parameters

in	<i>arStream</i>	The destination stream.
in	<i>arManipBase</i>	The CManipBase object.

Returns

The destination stream.

5.2.2.7 `template<typename T3 > std::istream& lfc1::udtsup::operator>> (std::istream & arStream, const CManipBase< T3 > & arManipBase)`

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

Template Parameters

<i>T3</i>	The user-defined manipulator function argument type.
-----------	--

Parameters

in	<i>arStream</i>	The source stream.
out	<i>arManipBase</i>	The CManipBase object.

Returns

The source stream.

5.3 Miscellaneous class library

Classes

- class [lfc1::misc::CDemangler](#)

This class provides RAI to name demangling.

5.3.1 Detailed Description

This library contains a set of miscellaneous classes to support the compiler, e.g. name demangling.

5.4 Numeric class library

Classes

- class `lfc1::numeric::CInteger< IT >`
This class template represents little endian integers on big endian operating systems or big endian integers on little endian operating systems.
- class `std::numeric_limits< lfc1::numeric::CInteger< IT > >`
This template class is a specialization for `std::numeric_limits` for `CInteger<IT>`.

Typedefs

- typedef `std::int16_t lfc1::numeric::TInt16LE`
This type represents a 16-bit little endian signed integer.
- typedef `CInteger< std::int16_t > lfc1::numeric::TInt16BE`
This type represents a 16-bit big endian signed integer.
- typedef `std::uint16_t lfc1::numeric::TUInt16LE`
This type represents a 16-bit little endian unsigned integer.
- typedef `CInteger< std::uint16_t > lfc1::numeric::TUInt16BE`
This type represents a 16-bit big endian unsigned integer.
- typedef `std::int32_t lfc1::numeric::TInt32LE`
This type represents a 32-bit little endian signed integer.
- typedef `CInteger< std::int32_t > lfc1::numeric::TInt32BE`
This type represents a 32-bit big endian signed integer.
- typedef `std::uint32_t lfc1::numeric::TUInt32LE`
This type represents a 32-bit little endian unsigned integer.
- typedef `CInteger< std::uint32_t > lfc1::numeric::TUInt32BE`
This type represents a 32-bit big endian unsigned integer.
- typedef `std::int64_t lfc1::numeric::TInt64LE`
This type represents a 64-bit little endian signed integer.
- typedef `CInteger< std::int64_t > lfc1::numeric::TInt64BE`
This type represents a 64-bit big endian signed integer.
- typedef `std::uint64_t lfc1::numeric::TUInt64LE`
This type represents a 64-bit little endian unsigned integer.
- typedef `CInteger< std::uint64_t > lfc1::numeric::TUInt64BE`
This type represents a 64-bit big endian unsigned integer.

Functions

- template<typename IT >
void `lfc1::numeric::swap` (`CInteger< IT > &arLHS`, `CInteger< IT > &arRHS`) noexcept
This template function overloads `std::swap` for `CInteger<IT>` objects.

5.4.1 Detailed Description

This library contains a set of templates and classes which represent numbers stored in various ways.

5.4.2 Function Documentation

5.4.2.1 `template<typename IT > void lfc1::numeric::swap (CInteger< IT > & arLHS, CInteger< IT > & arRHS)`
`[noexcept]`

This template function overloads `std::swap` for `CInteger<IT>` objects.

Template Parameters

<i>IT</i>	The integer type.
-----------	-------------------

Parameters

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

5.5 Filesystem class library

Classes

- class [lfc1::filesystem::CDirectory](#)
This class wraps a directory iterator so that directory entries could be loaded and sorted.
- class [lfc1::filesystem::CRecDirectory](#)
This class wraps a recursive directory iterator so that directory entries could be loaded and sorted.

Functions

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

5.5.1 Detailed Description

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

5.5.2 Function Documentation

5.5.2.1 void lfc1::filesystem::swap (CDirectory & arLHS, CDirectory & arRHS) [noexcept]

This function overloads std::swap for [CDirectory](#) objects.

Parameters

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

5.5.2.2 void lfc1::filesystem::swap (CRecDirectory & arLHS, CRecDirectory & arRHS) [noexcept]

This function overloads std::swap for [CRecDirectory](#) objects.

Parameters

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

5.6 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.
- CDateManip `lfc1::datetime::setdateformat` (const char *apFormat)
This function is the date manipulator. It is named following the naming convention of the standard manipulators.
- 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.
- CDateTimeManip `lfc1::datetime::setdatetimeformat` (const char *apFormat)
This function is the date and time manipulator. It is named following the naming convention of the standard manipulators.
- 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.
- CTimeManip `lfc1::datetime::settimeformat` (const char *apFormat)
This function is the time manipulator. It is named following the naming convention of the standard manipulators.

5.6.1 Detailed Description

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

5.6.2 Function Documentation

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

This function sends a CTime object to an output stream.

Parameters

in	<i>arStream</i>	The destination output stream.
in	<i>arTime</i>	The CTime object to send.

Returns

The destination output stream.

5.6.2.2 `std::ostream & lfc1::datetime::operator<< (std::ostream & arStream, const CDate & arDate)`

This function sends a [CDate](#) object to an output stream.

Parameters

in	<i>arStream</i>	The destination output stream.
in	<i>arDate</i>	The CDate object to send.

Returns

The destination output stream.

5.6.2.3 `std::ostream & lfc1::datetime::operator<< (std::ostream & arStream, const CDateTime & arDateTime)`

This function sends a [CDateTime](#) object to an output stream.

Parameters

in	<i>arStream</i>	The destination output stream.
in	<i>arDateTime</i>	The CDateTime object to send.

Returns

The destination output stream.

5.6.2.4 `std::istream & lfc1::datetime::operator>> (std::istream & arStream, CTime & arTime)`

This function receives a [CTime](#) object from an input stream.

Parameters

in	<i>arStream</i>	The source input stream.
out	<i>arTime</i>	The received CTime object.

Returns

The source input stream.

5.6.2.5 `std::istream & lfc1::datetime::operator>> (std::istream & arStream, CDate & arDate)`

This function receives a [CDate](#) object from an input stream.

Parameters

in	<i>arStream</i>	The source input stream.
out	<i>arDate</i>	The received CDate object.

Returns

The source input stream.

5.6.2.6 `std::istream & lfc1::datetime::operator>> (std::istream & arStream, CDateTime & arDateTime)`

This function receives a [CDateTime](#) object from an input stream.

Parameters

in	<i>arStream</i>	The source input stream.
out	<i>arDateTime</i>	The received CDateTime object.

Returns

The source input stream.

5.6.2.7 `CDateManip lfc1::datetime::setdateformat (const char * apFormat)`

This function is the date manipulator. It is named following the naming convention of the standard manipulators.

Parameters

in	<i>apFormat</i>	The date format. If set to null or an empty string, it clears the date format effectively resetting it to the default date format.
----	-----------------	--

Returns

The date manipulator helper.

5.6.2.8 `CDateTimeManip lfc1::datetime::setdatetimeformat (const char * apFormat)`

This function is the date and time manipulator. It is named following the naming convention of the standard manipulators.

Parameters

in	<i>apFormat</i>	The date and time format. A null pointer or empty string will be considered to mean use the default format.
----	-----------------	---

Returns

The date and time manipulator helper.

5.6.2.9 `CTimeManip lfc1::datetime::settimeformat (const char * apFormat)`

This function is the time manipulator. It is named following the naming convention of the standard manipulators.

Parameters

in	<i>apFormat</i>	The time format. A null pointer or empty string will be considered to mean use the default format.
----	-----------------	--

Returns

The time manipulator helper.

5.6.2.10 `void lfc1::datetime::swap (CTimeDuration & arLHS, CTimeDuration & arRHS)` `[noexcept]`

This function overloads `std::swap` for [CTimeDuration](#) objects.

Parameters

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

5.6.2.11 `void lfc1::datetime::swap (CDateDuration & arLHS, CDateDuration & arRHS)` `[noexcept]`

This function overloads `std::swap` for [CDateDuration](#) objects.

Parameters

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

5.6.2.12 `void lfc1::datetime::swap (CDateTimeDuration & arLHS, CDateTimeDuration & arRHS)` `[noexcept]`

This function overloads `std::swap` for [CDateTimeDuration](#) objects.

Parameters

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

5.6.2.13 `void lfc1::datetime::swap (CTime & arLHS, CTime & arRHS)` `[noexcept]`

This function overloads `std::swap` for [CTime](#) objects.

Parameters

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

5.6.2.14 `void lfc1::datetime::swap (CDate & arLHS, CDate & arRHS)` `[noexcept]`

This function overloads `std::swap` for [CDate](#) objects.

Parameters

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

5.6.2.15 `void lfc1::datetime::swap (CDateTime & arLHS, CDateTime & arRHS)` `[noexcept]`

This function overloads `std::swap` for [CDateTime](#) objects.

Parameters

in, out	<i>arLHS</i>	The first object to swap.
in, out	<i>arRHS</i>	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 int [MIN_YEAR](#) = 0

The minimum year.
- static const int [MAX_YEAR](#) = 9999

The maximum year.
- static const int [MONTH_PER_YEAR](#) = 12

The number of months in a year.
- static const int [MIN_MONTH](#) = 1

The lowest month number in a year.
- static const int [MAX_MONTH](#) = 12

The highest month number in a year.
- static const int [DAY_PER_WEEK](#) = 7

The number of days per week.
- static const int [MIN_DAY](#) = 1

The lowest day number in a month.
- static const int [MAX_DAY](#) = 31

The highest day number in a month.

- static const int `FIRST_DAY_MON` = 0
- static const int `FIRST_DAY_SUN` = 6
- static const int `MIN_DATENO` = 2298874

Julian day number for 01-Jan-1582.

- static const int `MAX_DATENO` = 2817152

Julian day number for 31-Dec-3000.

- static const int `DATENO_OFFSET_SUN` = 1
- static const int `MIN_DATE_YEAR` = 1582
- static const int `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

<code>in</code>	<i>avDateNo</i>	The Julian day number to assign to the <code>CDate</code> object.
-----------------	-----------------	---

Exceptions

<code>std::out_of_range</code>	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

<code>in</code>	<i>avYear</i>	The given year.
<code>in</code>	<i>avMonth</i>	The given month.
<code>in</code>	<i>avDay</i>	The given day.

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

This function copy constructs a `CDate` object.

Parameters

<i>in</i>	<i>arRHS</i>	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

<i>in</i>	<i>arRHS</i>	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

<i>in</i>	<i>arStream</i>	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

<i>in</i>	<i>arStream</i>	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

<i>in</i>	<i>avYear</i>	The given year.
<i>in</i>	<i>avMonth</i>	The given month.
<i>in</i>	<i>avDay</i>	The given day.

Exceptions

<i>std::out_of_range</i>	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

<i>std::out_of_range</i>	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	<i>arDateDuration</i>	The duration to add to the date.
----	-----------------------	----------------------------------

Returns

The [CDate](#) object.

Exceptions

<i>std::out_of_range</i>	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

<i>std::out_of_range</i>	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	<i>arDateDuration</i>	The duration to subtract from the date.
----	-----------------------	---

Returns

The [CDate](#) object.

Exceptions

<i>std::out_of_range</i>	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	<i>avDateNo</i>	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	<i>arRHS</i>	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	<i>arRHS</i>	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	<i>avYear</i>	The year.
in	<i>avMonth</i>	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	<i>avYear</i>	The year.
----	---------------	-----------

Return values

<i>true</i>	The year is a leap year.
<i>false</i>	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	<i>avYear</i>	The year of a date. (0 - 9999)
in	<i>avMonth</i>	The month of a date. (1 - 12)
in	<i>avDay</i>	The day of a date. (1 - 31)

Exceptions

<i>std::out_of_range</i>	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	<i>arRHS</i>	The CDate object to swap with.
---------	--------------	--

6.1.4 Member Data Documentation

6.1.4.1 `const int 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 int 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 int 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 int 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:

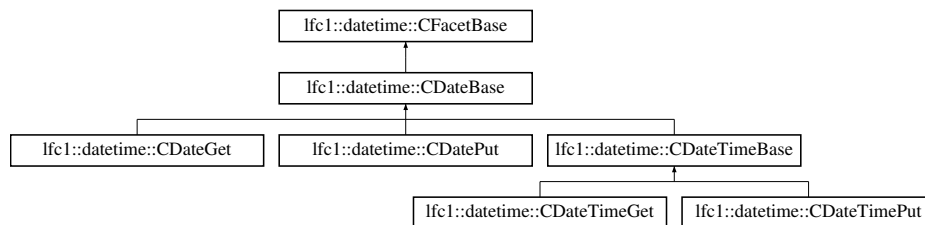
- `include/lfc1/datetime/cdate.hpp`
- `datetime/library/src/cdate.cpp`

6.2 Ifc1::datetime::CDateBase Class Reference

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

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

Inheritance diagram for Ifc1::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.
- virtual [~CDateBase](#) () noexcept
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 &arIoBase, 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 &arIoBase, 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 &arIoBase, 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	<i>arDate</i>	The given date expressed as a Julian day number.
in, out	<i>arWeekBasedYear</i>	Set to the starting week-based year on input and contains the resulting week-based year on output.
out	<i>apWeekBasedWeekNo</i>	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	<i>arDate</i>	The given date expressed as a Julian day number.
in	<i>avYear</i>	The year of the given date.

in	<i>avDay</i>	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	<i>avFormat</i>	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 & arlosBase, std::ios_base::iostate & arloState, int & arMonthNo, bool avIsShort) [static], [protected]`

This function parses an input iterator for a month name.

Parameters

in	<i>avNext</i>	The input iterator.
in	<i>avEnd</i>	The input end iterator.
in	<i>arlosBase</i>	The stream formatting information.
out	<i>arloState</i>	The stream state.
out	<i>arMonthNo</i>	The storage for the resulting month number.
in	<i>avIsShort</i>	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 & arlosBase, std::ios_base::iostate & arloState, int * apNumber, int avDigits, char avPadding) [static], [protected], [noexcept], [inherited]`

This function parses an input iterator for an integer value.

Parameters

in	<i>avNext</i>	The input iterator.
in	<i>avEnd</i>	The input end iterator.
in	<i>arlosBase</i>	The stream formatting information.
out	<i>arloState</i>	The stream state.
out	<i>apNumber</i>	The storage for the resulting integer.
in	<i>avDigits</i>	The maximum number of digits allowed for the integer.
in	<i>avPadding</i>	The padding character allowed for the integer value.

Return values

<i>ios_base::goodbit</i>	A valid integer was found.
<i>ios_base::failbit</i>	A valid integer was not found.

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

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

Parameters

in	<i>avNext</i>	The input iterator.
in	<i>avEnd</i>	The input end iterator.
in	<i>arlosBase</i>	The stream formatting information.
out	<i>arloState</i>	The stream state.
out	<i>arWeekDayNo</i>	The storage for the resulting day of the week. (0 = Sunday)
in	<i>avIsShort</i>	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 & arloState, 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	<i>avNext</i>	The input iterator.
in	<i>avEnd</i>	The input end iterator.
out	<i>arloState</i>	The stream state.
in	<i>avExpectedChar</i>	The next character expected from the input iterator.

Return values

<i>ios_base::goodbit</i>	The expected character was found.
<i>ios_base::failbit</i>	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	<i>avOut</i>	The output iterator.
in	<i>avMonthNo</i>	The number of the month.
in	<i>avIsShort</i>	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	<i>avOut</i>	The output iterator.
in	<i>avNumber</i>	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	<i>avOut</i>	The output iterator.
in	<i>avNumber</i>	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	<i>avOut</i>	The output iterator.
in	<i>avNumber</i>	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	<i>avOut</i>	The output iterator.
in	<i>avNumber</i>	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	<i>avOut</i>	The output iterator.
in	<i>avNumber</i>	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	<i>avOut</i>	The output iterator.
in	<i>avWeekDayNo</i>	The number of the week day.
in	<i>avIsShort</i>	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 Ifc1::datetime::CDateDuration::CDateDuration (int avDuration, [EUnits](#) avUnit = E_U_DAY) [noexcept]

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

Parameters

in	<i>avDuration</i>	The date duration.
in	<i>avUnit</i>	The unit of measure of the date duration.

6.3.2.2 Ifc1::datetime::CDateDuration::CDateDuration (const [CDateDuration](#) & arRHS) [noexcept]

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

Parameters

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

6.3.2.3 Ifc1::datetime::CDateDuration::CDateDuration ([CDateDuration](#) && arRHS) [noexcept]

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

Parameters

in	<i>arRHS</i>	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	<i>arRHS</i>	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	<i>arRHS</i>	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	<i>arRHS</i>	The CDateDuration object to swap with.
---------	--------------	--

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

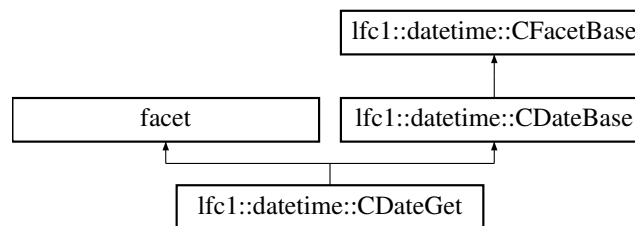
- include/lfc1/datetime/cdate.hpp
- datetime/library/src/cdateduration.cpp

6.4 Ifc1::datetime::CDateGet Class Reference

This class is a [CDate](#) class input facet.

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

Inheritance diagram for Ifc1::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.
- virtual [~CDateGet](#) () noexcept
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 &arIoBase, 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 &arlosBase, std::ios_base::iostate &arloState, [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 &arlosBase, std::ios_base::iostate &arloState, 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 &arlosBase, std::ios_base::iostate &arloState, 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 &arloState, 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 &arlosBase, std::ios_base::iostate &arloState, 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	<i>avRelease</i>	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 & *arlosBase*, std::ios_base::iostate & *arloState*, CDate & *arDate*, const std::string & *arFormat*) const [protected], [virtual]

This function implements the behavior of the [mvGet\(\)](#) function.

Parameters

in	<i>avNext</i>	The input iterator.
in	<i>avEnd</i>	The input end iterator.
in	<i>arlosBase</i>	The stream formatting information.
out	<i>arloState</i>	The stream state.
out	<i>arDate</i>	The date retrieved from the input stream.
in	<i>arFormat</i>	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 & *arlosBase*, std::ios_base::iostate & *arloState*, 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 `strptime()` 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	<i>avNext</i>	The input iterator.
in	<i>avEnd</i>	The input end iterator.
in	<i>arlosBase</i>	The stream formatting information.
out	<i>arloState</i>	The stream state.
out	<i>arDate</i>	The date retrieved from the input stream.
in	<i>arFormat</i>	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	<i>arDate</i>	The given date expressed as a Julian day number.
in, out	<i>arWeekBased-Year</i>	Set to the starting week-based year on input and contains the resulting week-based year on output.
out	<i>apWeekBased-WeekNo</i>	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	<i>arDate</i>	The given date expressed as a Julian day number.
in	<i>avYear</i>	The year of the given date.
in	<i>avDay</i>	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	<i>avFormat</i>	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 & *arlosBase*, std::ios_base::iostate & *arloState*, int & *arMonthNo*, bool *avIsShort*) [static], [protected], [inherited]

This function parses an input iterator for a month name.

Parameters

in	<i>avNext</i>	The input iterator.
in	<i>avEnd</i>	The input end iterator.
in	<i>arlosBase</i>	The stream formatting information.
out	<i>arloState</i>	The stream state.
out	<i>arMonthNo</i>	The storage for the resulting month number.
in	<i>avIsShort</i>	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 & *arlosBase*, std::ios_base::iostate & *arloState*, int * *apNumber*, int *avDigits*, char *avPadding*) [static], [protected], [noexcept], [inherited]

This function parses an input iterator for an integer value.

Parameters

in	<i>avNext</i>	The input iterator.
in	<i>avEnd</i>	The input end iterator.
in	<i>arlosBase</i>	The stream formatting information.
out	<i>arloState</i>	The stream state.
out	<i>apNumber</i>	The storage for the resulting integer.
in	<i>avDigits</i>	The maximum number of digits allowed for the integer.
in	<i>avPadding</i>	The padding character allowed for the integer value.

Return values

<i>ios_base::goodbit</i>	A valid integer was found.
<i>ios_base::failbit</i>	A valid integer was not found.

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

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

Parameters

in	<i>avNext</i>	The input iterator.
in	<i>avEnd</i>	The input end iterator.
in	<i>arlosBase</i>	The stream formatting information.
out	<i>arloState</i>	The stream state.
out	<i>arWeekDayNo</i>	The storage for the resulting day of the week. (0 = Sunday)
in	<i>avIsShort</i>	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 & arloState, 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	<i>avNext</i>	The input iterator.
in	<i>avEnd</i>	The input end iterator.
out	<i>arloState</i>	The stream state.
in	<i>avExpectedChar</i>	The next character expected from the input iterator.

Return values

<i>ios_base::goodbit</i>	The expected character was found.
<i>ios_base::failbit</i>	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	<i>avOut</i>	The output iterator.
in	<i>avMonthNo</i>	The number of the month.
in	<i>avIsShort</i>	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	<i>avOut</i>	The output iterator.
in	<i>avNumber</i>	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	<i>avOut</i>	The output iterator.
in	<i>avNumber</i>	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	<i>avOut</i>	The output iterator.
in	<i>avNumber</i>	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	<i>avOut</i>	The output iterator.
in	<i>avNumber</i>	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	<i>avOut</i>	The output iterator.
in	<i>avNumber</i>	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:

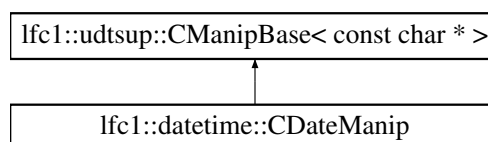
- include/lfc1/datetime/cdate.hpp
- datetime/library/src/cdateget.cpp

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 Types

- typedef void(* [TManipFunc](#))(std::ios_base &, const char *)
Single argument manipulator signature.

Public Member Functions

- [CDateManip](#) (const char *apFormat)
This function creates a [CDateManip](#) object.
- [CDateManip](#) (const [CDateManip](#) &)=default
Uses default implementation.
- [~CDateManip](#) () noexcept
This function destroys a [CDateManip](#) object.
- [CDateManip](#) & [operator=](#) (const [CDateManip](#) &)=default
Uses default implementation.

Static Public Member Functions

- static std::string [smvGetFormat](#) (std::ios_base &arlosBase)
This function obtains the date format from a stream.
- static std::ios_base::iostate [smvCopyFmtErr](#) (std::basic_ios< char > &arlos)
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	<i>apFormat</i>	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	<i>arlos</i>	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	<i>arlosBase</i>	The stream containing the date format.
----	------------------	--

Returns

The date format.

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

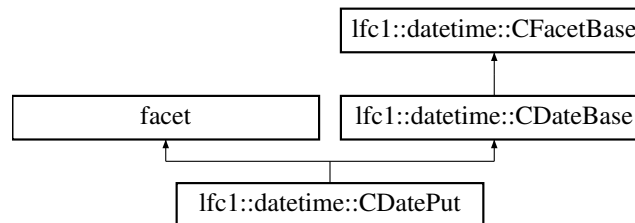
- include/lfc1/datetime/cdate.hpp
- datetime/library/src/cdatemanip.cpp

6.6 Ifc1::datetime::CDatePut Class Reference

This class is a [CDate](#) class output facet.

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

Inheritance diagram for Ifc1::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.
- virtual [~CDatePut](#) () noexcept
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 &arlosBase, std::ios_base::iostate &arloState, 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 &arlosBase, std::ios_base::iostate &arloState, 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 &arloState, 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 &arlosBase, std::ios_base::iostate &arloState, 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	<i>avOut</i>	The output iterator.
in	<i>arDate</i>	The date to be sent to the output stream.
in	<i>arFormat</i>	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	<i>avOut</i>	The output iterator.
in	<i>arDate</i>	The date to be sent to the output stream.
in	<i>arFormat</i>	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	<i>arDate</i>	The given date expressed as a Julian day number.
in, out	<i>arWeekBased-Year</i>	Set to the starting week-based year on input and contains the resulting week-based year on output.
out	<i>apWeekBased-WeekNo</i>	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	<i>arDate</i>	The given date expressed as a Julian day number.
in	<i>avYear</i>	The year of the given date.
in	<i>avDay</i>	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	<i>avFormat</i>	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 & arIosState, int & arMonthNo, bool avIsShort) [static], [protected], [inherited]`

This function parses an input iterator for a month name.

Parameters

in	<i>avNext</i>	The input iterator.
in	<i>avEnd</i>	The input end iterator.
in	<i>arlosBase</i>	The stream formatting information.
out	<i>arloState</i>	The stream state.
out	<i>arMonthNo</i>	The storage for the resulting month number.
in	<i>avIsShort</i>	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 & *arlosBase*, std::ios_base::iostate & *arloState*, int * *apNumber*, int *avDigits*, char *avPadding*) [static], [protected], [noexcept], [inherited]

This function parses an input iterator for an integer value.

Parameters

in	<i>avNext</i>	The input iterator.
in	<i>avEnd</i>	The input end iterator.
in	<i>arlosBase</i>	The stream formatting information.
out	<i>arloState</i>	The stream state.
out	<i>apNumber</i>	The storage for the resulting integer.
in	<i>avDigits</i>	The maximum number of digits allowed for the integer.
in	<i>avPadding</i>	The padding character allowed for the integer value.

Return values

<i>ios_base::goodbit</i>	A valid integer was found.
<i>ios_base::failbit</i>	A valid integer was not found.

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

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

Parameters

in	<i>avNext</i>	The input iterator.
in	<i>avEnd</i>	The input end iterator.
in	<i>arlosBase</i>	The stream formatting information.
out	<i>arloState</i>	The stream state.
out	<i>arWeekDayNo</i>	The storage for the resulting day of the week. (0 = Sunday)
in	<i>avIsShort</i>	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 & arloState, 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	<i>avNext</i>	The input iterator.
in	<i>avEnd</i>	The input end iterator.
out	<i>arloState</i>	The stream state.
in	<i>avExpectedChar</i>	The next character expected from the input iterator.

Return values

<i>ios_base::goodbit</i>	The expected character was found.
<i>ios_base::failbit</i>	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	<i>avOut</i>	The output iterator.
in	<i>avMonthNo</i>	The number of the month.
in	<i>avIsShort</i>	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	<i>avOut</i>	The output iterator.
in	<i>avNumber</i>	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	<i>avOut</i>	The output iterator.
in	<i>avNumber</i>	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	<i>avOut</i>	The output iterator.
in	<i>avNumber</i>	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	<i>avOut</i>	The output iterator.
in	<i>avNumber</i>	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	<i>avOut</i>	The output iterator.
in	<i>avNumber</i>	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 av-Millisecond, 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 increments 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 long long MIN_DATETIMENO = 0LL
- static const long long MAX_DATETIMENO = 325352159999999LL

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	<i>avDateTimeNo</i>	The number of milliseconds since 01-Jan-1970 00:00:00.000.
----	---------------------	--

Exceptions

<i>std::out_of_range</i>	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	<i>avYear</i>	The given year.
in	<i>avMonth</i>	The given month.
in	<i>avDay</i>	The given day.
in	<i>avHour</i>	The given hour.
in	<i>avMinute</i>	The given minute.
in	<i>avSecond</i>	The given second.
in	<i>avMillisecond</i>	The given millisecond.
in	<i>avDst</i>	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	<i>arRHS</i>	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

<i>in</i>	<i>arRHS</i>	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

<i>in</i>	<i>arStream</i>	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

<i>in</i>	<i>arStream</i>	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

<i>in</i>	<i>avYear</i>	The given year.
<i>in</i>	<i>avMonth</i>	The given month.

in	<i>avDay</i>	The given day.
in	<i>avHour</i>	The given hour.
in	<i>avMinute</i>	The given minute.
in	<i>avSecond</i>	The given second.
in	<i>avMillisecond</i>	The given millisecond.
in	<i>avDst</i>	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

<i>std::system_error</i>	Indicates that conversion to a <code>time_t</code> value failed.
<i>std::out_of_range</i>	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

<i>std::out_of_range</i>	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 increments the [CDateTime](#) object by the given duration.

Parameters

in	<i>arDateTimeDuration</i>	The duration to add to the date and time.
----	---------------------------	---

Returns

The [CDateTime](#) object.

Exceptions

<i>std::out_of_range</i>	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

<i>std::out_of_range</i>	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

<i>in</i>	<i>arDateTimeDuration</i>	The duration to subtract from the date and time.
-----------	---------------------------	--

Returns

The [CDateTime](#) object.

Exceptions

<i>std::out_of_range</i>	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	<i>avDateTimeNo</i>	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	<i>arRHS</i>	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	<i>arRHS</i>	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	<i>avDst</i>	The daylight savings indicator (-1, 0 or 1).
----	--------------	--

Exceptions

<i>std::out_of_range</i>	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

<code>in, out</code>	<code>arRHS</code>	The CDateTime object to swap with.
----------------------	--------------------	--

6.7.4 Member Data Documentation

6.7.4.1 `const long long lfc1::datetime::CDateTime::MAX_DATETIMEO = 32535215999999LL [static]`

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

6.7.4.2 `const long long lfc1::datetime::CDateTime::MIN_DATETIMEO = 0LL [static]`

DateTimeNo for

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

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

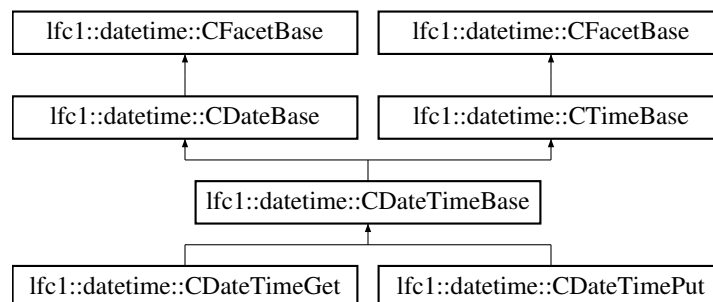
- `include/lfc1/datetime/cdatetime.hpp`
- `datetime/library/src/cdatetime.cpp`

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.
- virtual ~CDateTimeBase () noexcept
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 &arIoBase, 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 &arIoBase, 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 &arIoBase, 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 &arlosBase, std::ios_base::iostate &arloState, 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 &arlosBase, std::ios_base::iostate &arloState, 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 &arloState, 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 &arlosBase, std::ios_base::iostate &arloState, 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	<i>arDate</i>	The given date expressed as a Julian day number.
in, out	<i>arWeekBased-Year</i>	Set to the starting week-based year on input and contains the resulting week-based year on output.
out	<i>apWeekBased-WeekNo</i>	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	<i>arDate</i>	The given date expressed as a Julian day number.
in	<i>avYear</i>	The year of the given date.
in	<i>avDay</i>	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	<i>avFormat</i>	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 & arlosBase, std::ios_base::iostate & arloState, bool & arHasAmPm, bool & arIsPm) [static], [protected], [inherited]`

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

Parameters

in	<i>avNext</i>	The input iterator.
in	<i>avEnd</i>	The input end iterator.
in	<i>arlosBase</i>	The stream formatting information.
out	<i>arloState</i>	The stream state.
out	<i>arHasAmPm</i>	The storage for the resulting has AM/PM indicator (true) or not (false).
out	<i>arIsPm</i>	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 & *arlosBase*, std::ios_base::iostate & *arloState*, int & *arMonthNo*, bool *avIsShort*) [static], [protected], [inherited]

This function parses an input iterator for a month name.

Parameters

in	<i>avNext</i>	The input iterator.
in	<i>avEnd</i>	The input end iterator.
in	<i>arlosBase</i>	The stream formatting information.
out	<i>arloState</i>	The stream state.
out	<i>arMonthNo</i>	The storage for the resulting month number.
in	<i>avIsShort</i>	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 & *arlosBase*, std::ios_base::iostate & *arloState*, int * *apNumber*, int *avDigits*, char *avPadding*) [static], [protected], [noexcept], [inherited]

This function parses an input iterator for an integer value.

Parameters

in	<i>avNext</i>	The input iterator.
in	<i>avEnd</i>	The input end iterator.
in	<i>arlosBase</i>	The stream formatting information.
out	<i>arloState</i>	The stream state.
out	<i>apNumber</i>	The storage for the resulting integer.
in	<i>avDigits</i>	The maximum number of digits allowed for the integer.
in	<i>avPadding</i>	The padding character allowed for the integer value.

Return values

<i>ios_base::goodbit</i>	A valid integer was found.
<i>ios_base::failbit</i>	A valid integer was not found.

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

This function parses an input iterator for an integer value.

Parameters

in	<i>avNext</i>	The input iterator.
in	<i>avEnd</i>	The input end iterator.
in	<i>arlosBase</i>	The stream formatting information.
out	<i>arloState</i>	The stream state.
out	<i>apNumber</i>	The storage for the resulting integer.
in	<i>avDigits</i>	The maximum number of digits allowed for the integer.
in	<i>avPadding</i>	The padding character allowed for the integer value.

Return values

<i>ios_base::goodbit</i>	A valid integer was found.
<i>ios_base::failbit</i>	A valid integer was not found.

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

This function parses an input iterator for a timezone name.

Parameters

in	<i>avNext</i>	The input iterator.
in	<i>avEnd</i>	The input end iterator.
out	<i>arloState</i>	The stream state.
out	<i>arDst</i>	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 & arlosBase, std::ios_base::iostate & arloState, int & arDst) [static], [protected]

This function parses an input iterator for a timezone name.

Parameters

in	<i>avNext</i>	The input iterator.
in	<i>avEnd</i>	The input end iterator.
in	<i>arlosBase</i>	The stream formatting information.
out	<i>arloState</i>	The stream state.
out	<i>arDst</i>	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 & arlosBase, std::ios_base::iostate & arloState, int & arWeekDayNo, bool avIsShort) [static], [protected], [inherited]

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

Parameters

in	<i>avNext</i>	The input iterator.
in	<i>avEnd</i>	The input end iterator.
in	<i>arlosBase</i>	The stream formatting information.
out	<i>arloState</i>	The stream state.
out	<i>arWeekDayNo</i>	The storage for the resulting day of the week. (0 = Sunday)
in	<i>avIsShort</i>	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 & *arloState*, 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	<i>avNext</i>	The input iterator.
in	<i>avEnd</i>	The input end iterator.
out	<i>arloState</i>	The stream state.
in	<i>avExpectedChar</i>	The next character expected from the input iterator.

Return values

<i>ios_base::goodbit</i>	The expected character was found.
<i>ios_base::failbit</i>	The expected character was not found.

6.8.2.12 CFacetBase::TInIt lfc1::datetime::CFacetBase::smvIgnoreChar (TInIt *avNext*, TInIt *avEnd*, std::ios_base::iostate & *arloState*, 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	<i>avNext</i>	The input iterator.
in	<i>avEnd</i>	The input end iterator.
out	<i>arloState</i>	The stream state.
in	<i>avExpectedChar</i>	The next character expected from the input iterator.

Return values

<i>ios_base::goodbit</i>	The expected character was found.
<i>ios_base::failbit</i>	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	<i>avOut</i>	The output iterator.
in	<i>avHour</i>	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	<i>avOut</i>	The output iterator.
in	<i>avMonthNo</i>	The number of the month.
in	<i>avIsShort</i>	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	<i>avOut</i>	The output iterator.
in	<i>avNumber</i>	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	<i>avOut</i>	The output iterator.
in	<i>avNumber</i>	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	<i>avOut</i>	The output iterator.
in	<i>avNumber</i>	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	<i>avOut</i>	The output iterator.
in	<i>avNumber</i>	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	<i>avOut</i>	The output iterator.
in	<i>avNumber</i>	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	<i>avOut</i>	The output iterator.
in	<i>avNumber</i>	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	<i>avOut</i>	The output iterator.
in	<i>avNumber</i>	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	<i>avOut</i>	The output iterator.
in	<i>avNumber</i>	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	<i>avOut</i>	The output iterator.
in	<i>avNumber</i>	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	<i>avOut</i>	The output iterator.
in	<i>avNumber</i>	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	<i>avOut</i>	The output iterator.
in	<i>avDst</i>	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	<i>avOut</i>	The output iterator.
in	<i>avDst</i>	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	<i>avOut</i>	The output iterator.
in	<i>avWeekDayNo</i>	The number of the week day.
in	<i>avIsShort</i>	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	<i>avDuration</i>	The date duration.
in	<i>avUnit</i>	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	<i>arRHS</i>	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

<code>in</code>	<code><i>arRHS</i></code>	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

<code>in</code>	<code><i>arRHS</i></code>	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

<code>in</code>	<code><i>arRHS</i></code>	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:

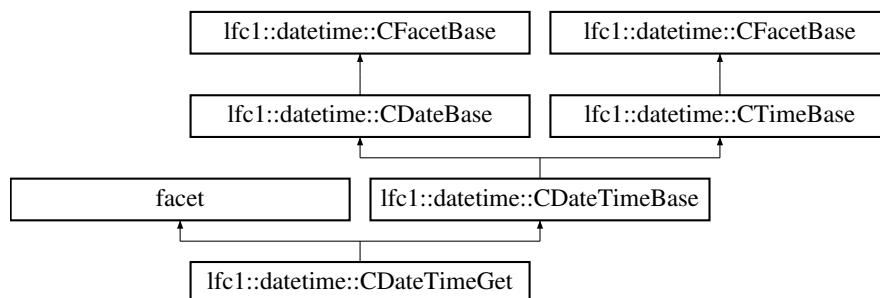
- include/lfc1/datetime/cdatetime.hpp
- datetime/library/src/cdatetimeduration.cpp

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.

- virtual `~CDateTimeGet () noexcept`

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 &arlosBase, std::ios_base::iostate &arloState, 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 &arlosBase, std::ios_base::iostate &arloState, 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 &arloState, int &arDst)`

This function parses an input iterator for a timezone name.

- static `TInIt smvGetTzOffset (TInIt arNext, TInIt arEnd, std::ios_base &arlosBase, std::ios_base::iostate &arloState, 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 &arlosBase, std::ios_base::iostate &arloState, 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 &arlosBase, std::ios_base::iostate &arloState, 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 &arloState, 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 &arloState, 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 &arlosBase, std::ios_base::iostate &arloState, 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 &arlosBase, std::ios_base::iostate &arloState, 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 &arlosBase, std::ios_base::iostate &arloState, 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	<i>avRelease</i>	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 & arlosBase, std::ios_base::iostate & arloState, CDateTime & arDateTime, const std::string & arFormat) const [protected], [virtual]`

This function implements the behavior of the [mvGet\(\)](#) function.

Parameters

in	<i>avNext</i>	The input iterator.
in	<i>avEnd</i>	The input end iterator.
in	<i>arlosBase</i>	The stream formatting information.
out	<i>arloState</i>	The stream state.
out	<i>arDateTime</i>	The date and time retrieved from the input stream.
in	<i>arFormat</i>	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 & arlosBase, std::ios_base::iostate & arloState, 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	<i>avNext</i>	The input iterator.
in	<i>avEnd</i>	The input end iterator.
in	<i>arlosBase</i>	The stream formatting information.
out	<i>arloState</i>	The stream state.
out	<i>arDateTime</i>	The date and time retrieved from the input stream.
in	<i>arFormat</i>	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	<i>arDate</i>	The given date expressed as a Julian day number.
in, out	<i>arWeekBased-Year</i>	Set to the starting week-based year on input and contains the resulting week-based year on output.
out	<i>apWeekBased-WeekNo</i>	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	<i>arDate</i>	The given date expressed as a Julian day number.
in	<i>avYear</i>	The year of the given date.
in	<i>avDay</i>	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	<i>avFormat</i>	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 & arlosBase, std::ios_base::iostate & arloState, bool & arHasAmPm, bool & arIsPm) [static], [protected], [inherited]`

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

Parameters

in	<i>avNext</i>	The input iterator.
in	<i>avEnd</i>	The input end iterator.
in	<i>arlosBase</i>	The stream formatting information.
out	<i>arloState</i>	The stream state.
out	<i>arHasAmPm</i>	The storage for the resulting has AM/PM indicator (true) or not (false).
out	<i>arIsPm</i>	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 & arlosBase, std::ios_base::iostate & arloState, int & arMonthNo, bool avIsShort) [static], [protected], [inherited]`

This function parses an input iterator for a month name.

Parameters

in	<i>avNext</i>	The input iterator.
in	<i>avEnd</i>	The input end iterator.
in	<i>arlosBase</i>	The stream formatting information.
out	<i>arloState</i>	The stream state.
out	<i>arMonthNo</i>	The storage for the resulting month number.
in	<i>avIsShort</i>	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 & *arlosBase*, std::ios_base::iostate & *arloState*, int * *apNumber*, int *avDigits*, char *avPadding*) [static], [protected], [noexcept], [inherited]

This function parses an input iterator for an integer value.

Parameters

in	<i>avNext</i>	The input iterator.
in	<i>avEnd</i>	The input end iterator.
in	<i>arlosBase</i>	The stream formatting information.
out	<i>arloState</i>	The stream state.
out	<i>apNumber</i>	The storage for the resulting integer.
in	<i>avDigits</i>	The maximum number of digits allowed for the integer.
in	<i>avPadding</i>	The padding character allowed for the integer value.

Return values

<i>ios_base::goodbit</i>	A valid integer was found.
<i>ios_base::failbit</i>	A valid integer was not found.

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

This function parses an input iterator for an integer value.

Parameters

in	<i>avNext</i>	The input iterator.
in	<i>avEnd</i>	The input end iterator.
in	<i>arlosBase</i>	The stream formatting information.
out	<i>arloState</i>	The stream state.
out	<i>apNumber</i>	The storage for the resulting integer.
in	<i>avDigits</i>	The maximum number of digits allowed for the integer.
in	<i>avPadding</i>	The padding character allowed for the integer value.

Return values

<i>ios_base::goodbit</i>	A valid integer was found.
<i>ios_base::failbit</i>	A valid integer was not found.

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

This function parses an input iterator for a timezone name.

Parameters

in	<i>avNext</i>	The input iterator.
in	<i>avEnd</i>	The input end iterator.
out	<i>arloState</i>	The stream state.
out	<i>arDst</i>	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 & arlosBase, std::ios_base::iostate & arloState, int & arDst) [static], [protected], [inherited]

This function parses an input iterator for a timezone name.

Parameters

in	<i>avNext</i>	The input iterator.
in	<i>avEnd</i>	The input end iterator.
in	<i>arlosBase</i>	The stream formatting information.
out	<i>arloState</i>	The stream state.
out	<i>arDst</i>	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 & arlosBase, std::ios_base::iostate & arloState, int & arWeekDayNo, bool avIsShort) [static], [protected], [inherited]

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

Parameters

in	<i>avNext</i>	The input iterator.
in	<i>avEnd</i>	The input end iterator.
in	<i>arlosBase</i>	The stream formatting information.
out	<i>arloState</i>	The stream state.
out	<i>arWeekDayNo</i>	The storage for the resulting day of the week. (0 = Sunday)
in	<i>avIsShort</i>	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 & arloState, 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	<i>avNext</i>	The input iterator.
in	<i>avEnd</i>	The input end iterator.
out	<i>arloState</i>	The stream state.
in	<i>avExpectedChar</i>	The next character expected from the input iterator.

Return values

<i>ios_base::goodbit</i>	The expected character was found.
<i>ios_base::failbit</i>	The expected character was not found.

6.10.3.14 CFacetBase::TInIt lfc1::datetime::CFacetBase::smvIgnoreChar (TInIt avNext, TInIt avEnd, std::ios_base::iostate & arloState, 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	<i>avNext</i>	The input iterator.
in	<i>avEnd</i>	The input end iterator.
out	<i>arloState</i>	The stream state.
in	<i>avExpectedChar</i>	The next character expected from the input iterator.

Return values

<i>ios_base::goodbit</i>	The expected character was found.
<i>ios_base::failbit</i>	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	<i>avOut</i>	The output iterator.
in	<i>avHour</i>	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	<i>avOut</i>	The output iterator.
in	<i>avMonthNo</i>	The number of the month.
in	<i>avIsShort</i>	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	<i>avOut</i>	The output iterator.
in	<i>avNumber</i>	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	<i>avOut</i>	The output iterator.
in	<i>avNumber</i>	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	<i>avOut</i>	The output iterator.
in	<i>avNumber</i>	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	<i>avOut</i>	The output iterator.
in	<i>avNumber</i>	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	<i>avOut</i>	The output iterator.
in	<i>avNumber</i>	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	<i>avOut</i>	The output iterator.
in	<i>avNumber</i>	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	<i>avOut</i>	The output iterator.
in	<i>avNumber</i>	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	<i>avOut</i>	The output iterator.
in	<i>avNumber</i>	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	<i>avOut</i>	The output iterator.
in	<i>avNumber</i>	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	<i>avOut</i>	The output iterator.
in	<i>avNumber</i>	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	<i>avOut</i>	The output iterator.
in	<i>avDst</i>	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	<i>avOut</i>	The output iterator.
in	<i>avDst</i>	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	<i>avOut</i>	The output iterator.
in	<i>avWeekDayNo</i>	The number of the week day.
in	<i>avIsShort</i>	Short (true) or long (false) week day name indicator.

Returns

The output iterator.

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

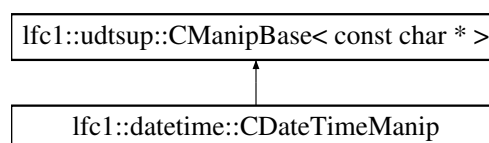
- include/lfc1/datetime/cdatetime.hpp
- datetime/library/src/cdatetimeget.cpp

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 Types

- typedef void(* [TManipFunc](#))(std::ios_base &, const char *)
Single argument manipulator signature.

Public Member Functions

- [CDateTimeManip](#) (const char *apFormat)
This function creates a [CDateTimeManip](#) object.
- [CDateTimeManip](#) (const [CDateTimeManip](#) &)=default
Uses default implementation.
- [~CDateTimeManip](#) () noexcept
This function destroys a [CDateTimeManip](#) object.
- [CDateTimeManip](#) & operator= (const [CDateTimeManip](#) &)=default
Uses default implementation.

Static Public Member Functions

- static std::string [smvGetFormat](#) (std::ios_base &arlosBase)
This function obtains the date and time format from a stream.
- static std::ios_base::iostate [smvCopyFmtErr](#) (std::basic_ios< char > &arlos)
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 (const char * apFormat)

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

Parameters

in	<i>apFormat</i>	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	<i>arlos</i>	The stream containing the error information.
----	--------------	--

Returns

The error information relating to date and time format copying.

6.11.3.2 std::string Ifc1::datetime::CDateTimeManip::smvGetFormat (std::ios_base & *arlosBase*) [static]

This function obtains the date and time format from a stream.

Parameters

in	<i>arlosBase</i>	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:

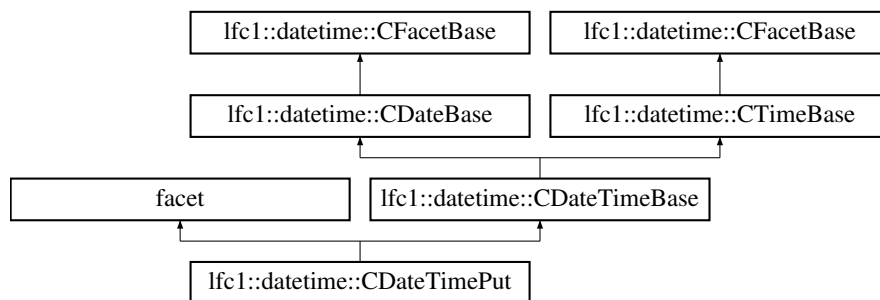
- include/lfc1/datetime/cdatetime.hpp
- datetime/library/src/cdatetimemanip.cpp

6.12 Ifc1::datetime::CDateTimePut Class Reference

This class is a [CDateTime](#) class output facet.

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

Inheritance diagram for Ifc1::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.*
 - virtual ~[CDateTimePut](#) () noexcept
 - 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 &arloState, int &arDst)
- This function parses an input iterator for a timezone name.*
- static [TInIt](#) [smvGetTzOffset](#) ([TInIt](#) arNext, [TInIt](#) arEnd, std::ios_base &arlosBase, std::ios_base::iostate &arloState, 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 &arlosBase, std::ios_base::iostate &arloState, 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 &arlosBase, std::ios_base::iostate &arloState, 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 [TInlt smvIgnoreChar](#) ([TInlt](#) avNext, [TInlt](#) 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 [TInlt smvIgnoreChar](#) ([TInlt](#) avNext, [TInlt](#) 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 [TInlt smvGetNumber](#) ([TInlt](#) avNext, [TInlt](#) avEnd, std::ios_base &arIoBase, std::ios_base::iostate &arIoState, int *apNumber, int avDigits, char avPadding) noexcept
This function parses an input iterator for an integer value.
- static [TInlt smvGetNumber](#) ([TInlt](#) avNext, [TInlt](#) avEnd, std::ios_base &arIoBase, std::ios_base::iostate &arIoState, int *apNumber, int avDigits, char avPadding) noexcept
This function parses an input iterator for an integer value.
- static [TOutlt smvPutNumber1](#) ([TOutlt](#) avOut, int avNumber) noexcept
This function sends a 1-digit number to an output iterator.
- static [TOutlt smvPutNumber1](#) ([TOutlt](#) avOut, int avNumber) noexcept
This function sends a 1-digit number to an output iterator.
- static [TOutlt smvPutNumber2Zero](#) ([TOutlt](#) avOut, int avNumber) noexcept
This function sends a 2-digit zero padded number to an output iterator.
- static [TOutlt smvPutNumber2Zero](#) ([TOutlt](#) avOut, int avNumber) noexcept
This function sends a 2-digit zero padded number to an output iterator.
- static [TOutlt smvPutNumber2Blank](#) ([TOutlt](#) avOut, int avNumber) noexcept
This function sends a 2-digit blank padded number to an output iterator.
- static [TOutlt smvPutNumber2Blank](#) ([TOutlt](#) avOut, int avNumber) noexcept
This function sends a 2-digit blank padded number to an output iterator.
- static [TOutlt smvPutNumber3Zero](#) ([TOutlt](#) avOut, int avNumber) noexcept
This function sends a 3-digit zero padded number to an output iterator.
- static [TOutlt smvPutNumber3Zero](#) ([TOutlt](#) avOut, int avNumber) noexcept
This function sends a 3-digit zero padded number to an output iterator.
- static [TOutlt smvPutNumber4Zero](#) ([TOutlt](#) avOut, int avNumber) noexcept
This function sends a 4-digit zero padded number to an output iterator.
- static [TOutlt smvPutNumber4Zero](#) ([TOutlt](#) avOut, int avNumber) noexcept
This function sends a 4-digit zero padded number to an output iterator.
- static [TInlt smvGetAmPm](#) ([TInlt](#) avNext, [TInlt](#) avEnd, std::ios_base &arIoBase, std::ios_base::iostate &arIoState, bool &arHasAmPm, bool &arIsPm)
This function parses an input iterator for an AM/PM indicator.
- static [TOutlt smvPutAmPm](#) ([TOutlt](#) 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	<i>avRelease</i>	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	<i>avOut</i>	The output iterator.
in	<i>arDateTime</i>	The date and time to be sent to the output stream.
in	<i>arFormat</i>	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	<i>avOut</i>	The output iterator.
in	<i>arDateTime</i>	The date and time to be sent to the output stream.
in	<i>arFormat</i>	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	<i>arDate</i>	The given date expressed as a Julian day number.
in, out	<i>arWeekBased-Year</i>	Set to the starting week-based year on input and contains the resulting week-based year on output.
out	<i>apWeekBased-WeekNo</i>	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	<i>arDate</i>	The given date expressed as a Julian day number.
in	<i>avYear</i>	The year of the given date.
in	<i>avDay</i>	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	<i>avFormat</i>	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 & *arlosBase*, std::ios_base::iostate & *arloState*, bool & *arHasAmPm*, bool & *arIsPm*) [static], [protected], [inherited]

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

Parameters

in	<i>avNext</i>	The input iterator.
in	<i>avEnd</i>	The input end iterator.
in	<i>arlosBase</i>	The stream formatting information.
out	<i>arloState</i>	The stream state.
out	<i>arHasAmPm</i>	The storage for the resulting has AM/PM indicator (true) or not (false).
out	<i>arIsPm</i>	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 & *arlosBase*, std::ios_base::iostate & *arloState*, int & *arMonthNo*, bool *avIsShort*) [static], [protected], [inherited]

This function parses an input iterator for a month name.

Parameters

in	<i>avNext</i>	The input iterator.
in	<i>avEnd</i>	The input end iterator.
in	<i>arlosBase</i>	The stream formatting information.
out	<i>arloState</i>	The stream state.
out	<i>arMonthNo</i>	The storage for the resulting month number.
in	<i>avIsShort</i>	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 & *arlosBase*, std::ios_base::iostate & *arloState*, int * *apNumber*, int *avDigits*, char *avPadding*) [static], [protected], [noexcept], [inherited]

This function parses an input iterator for an integer value.

Parameters

in	<i>avNext</i>	The input iterator.
in	<i>avEnd</i>	The input end iterator.
in	<i>arlosBase</i>	The stream formatting information.
out	<i>arloState</i>	The stream state.
out	<i>apNumber</i>	The storage for the resulting integer.
in	<i>avDigits</i>	The maximum number of digits allowed for the integer.
in	<i>avPadding</i>	The padding character allowed for the integer value.

Return values

<i>ios_base::goodbit</i>	A valid integer was found.
<i>ios_base::failbit</i>	A valid integer was not found.

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

This function parses an input iterator for an integer value.

Parameters

in	<i>avNext</i>	The input iterator.
in	<i>avEnd</i>	The input end iterator.
in	<i>arlosBase</i>	The stream formatting information.
out	<i>arloState</i>	The stream state.
out	<i>apNumber</i>	The storage for the resulting integer.
in	<i>avDigits</i>	The maximum number of digits allowed for the integer.
in	<i>avPadding</i>	The padding character allowed for the integer value.

Return values

<i>ios_base::goodbit</i>	A valid integer was found.
<i>ios_base::failbit</i>	A valid integer was not found.

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

This function parses an input iterator for a timezone name.

Parameters

in	<i>avNext</i>	The input iterator.
in	<i>avEnd</i>	The input end iterator.
out	<i>arloState</i>	The stream state.
out	<i>arDst</i>	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 & *arlosBase*, std::ios_base::iostate & *arloState*, int & *arDst*) [static], [protected], [inherited]

This function parses an input iterator for a timezone name.

Parameters

in	<i>avNext</i>	The input iterator.
in	<i>avEnd</i>	The input end iterator.
in	<i>arlosBase</i>	The stream formatting information.
out	<i>arloState</i>	The stream state.
out	<i>arDst</i>	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 & *arlosBase*, std::ios_base::iostate & *arloState*, int & *arWeekDayNo*, bool *avIsShort*) [static], [protected], [inherited]

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

Parameters

in	<i>avNext</i>	The input iterator.
in	<i>avEnd</i>	The input end iterator.
in	<i>arlosBase</i>	The stream formatting information.
out	<i>arloState</i>	The stream state.
out	<i>arWeekDayNo</i>	The storage for the resulting day of the week. (0 = Sunday)
in	<i>avIsShort</i>	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 & *arloState*, 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	<i>avNext</i>	The input iterator.
in	<i>avEnd</i>	The input end iterator.
out	<i>arloState</i>	The stream state.
in	<i>avExpectedChar</i>	The next character expected from the input iterator.

Return values

<i>ios_base::goodbit</i>	The expected character was found.
<i>ios_base::failbit</i>	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], [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	<i>avNext</i>	The input iterator.
in	<i>avEnd</i>	The input end iterator.
out	<i>arIoState</i>	The stream state.
in	<i>avExpectedChar</i>	The next character expected from the input iterator.

Return values

<i>ios_base::goodbit</i>	The expected character was found.
<i>ios_base::failbit</i>	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	<i>avOut</i>	The output iterator.
in	<i>avHour</i>	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	<i>avOut</i>	The output iterator.
in	<i>avMonthNo</i>	The number of the month.
in	<i>avIsShort</i>	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	<i>avOut</i>	The output iterator.
in	<i>avNumber</i>	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	<i>avOut</i>	The output iterator.
in	<i>avNumber</i>	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	<i>avOut</i>	The output iterator.
in	<i>avNumber</i>	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	<i>avOut</i>	The output iterator.
in	<i>avNumber</i>	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	<i>avOut</i>	The output iterator.
in	<i>avNumber</i>	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	<i>avOut</i>	The output iterator.
in	<i>avNumber</i>	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	<i>avOut</i>	The output iterator.
in	<i>avNumber</i>	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	<i>avOut</i>	The output iterator.
in	<i>avNumber</i>	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	<i>avOut</i>	The output iterator.
in	<i>avNumber</i>	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	<i>avOut</i>	The output iterator.
in	<i>avNumber</i>	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	<i>avOut</i>	The output iterator.
in	<i>avDst</i>	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	<i>avOut</i>	The output iterator.
in	<i>avDst</i>	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::misc::CDemangler Class Reference

This class provides RAI to name demangling.

```
#include <lfc1/misc/cdemangler.hpp>
```

Public Member Functions

- [CDemangler](#) (const char *apTypeName) noexcept
This function creates a [CDemangler](#) object.
- [CDemangler](#) (const [CDemangler](#) &)=delete
Not supported.
- [CDemangler](#) ([CDemangler](#) &&)=delete
Not supported.
- [~CDemangler](#) () noexcept
This function destroys a [CDemangler](#) object.
- [CDemangler](#) & operator= (const [CDemangler](#) &)=delete
Not supported.
- [CDemangler](#) & operator= ([CDemangler](#) &&)=delete
Not supported.
- operator const char * () const noexcept
This function obtains the demangled name.

6.13.1 Detailed Description

This class provides RAI to name demangling.

Some platforms require name demangling while other platforms do not. This class is aware which platforms require name demangling and does nothing for those platforms that do not require name demangling.

6.13.2 Constructor & Destructor Documentation

6.13.2.1 lfc1::misc::CDemangler::CDemangler (const char * *apTypeName*) [explicit], [noexcept]

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

Parameters

in	<i>apTypeName</i>	The name to demangle.
----	-------------------	-----------------------

6.13.3 Member Function Documentation

6.13.3.1 lfc1::misc::CDemangler::operator const char * () const [noexcept]

This function obtains the demangled name.

Returns

The demangled name.

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

- include/lfc1/misc/cdemangler.hpp
- misc/library/src/cdemangler.cpp

6.14 lfc1::filesystem::CDirectory Class Reference

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

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

Public Types

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

Public Member Functions

- [CDirectory](#) (const boost::filesystem::path &arPath)
This function creates a [CDirectory](#) object.
- [CDirectory](#) (const [CDirectory](#) &arRHS)
This function copy constructs a [CDirectory](#) object.
- [CDirectory](#) ([CDirectory](#) &&arRHS)
This function move constructs a [CDirectory](#) object.
- [~CDirectory](#) () noexcept
This function destroys a [CDirectory](#) object.
- [CDirectory](#) & operator= (const [CDirectory](#) &arRHS)

- This function assigns a [CDirectory](#) object to another [CDirectory](#) object.*
- [CDirectory](#) & operator= ([CDirectory](#) &&arRHS) noexcept
- This function moves a [CDirectory](#) object to another [CDirectory](#) object.*
- boost::filesystem::path [mvGetPath](#) () const
- This function obtains the name of the directory.*
- unsigned [mvGetNumEntries](#) () const noexcept
- This function obtains the number of directory entries.*
- [TIterator](#) [begin](#) () const noexcept
- This function obtains the beginning iterator to the list of directory entries.*
- [TIterator](#) [end](#) () const noexcept
- This function obtains the ending iterator to the list of directory entries.*
- void [swap](#) ([CDirectory](#) &arRHS) noexcept
- This function swaps a [CDirectory](#) object with another [CDirectory](#) object.*
- void [mvRefresh](#) ()
- This function loads and sorts the contents of a directory.*

6.14.1 Detailed Description

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

6.14.2 Constructor & Destructor Documentation

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

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

Parameters

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

6.14.2.2 lfc1::filesystem::CDirectory::CDirectory (const [CDirectory](#) & arRHS)

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

Parameters

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

6.14.2.3 lfc1::filesystem::CDirectory::CDirectory ([CDirectory](#) && arRHS)

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

Parameters

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

6.14.3 Member Function Documentation

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

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

Returns

The beginning iterator to the list of directory entries.

Note

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

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

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

Returns

The ending iterator to the list of directory entries.

Note

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

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

This function obtains the number of directory entries.

Returns

The number of directory entries.

6.14.3.4 **boost::filesystem::path lfc1::filesystem::CDirectory::mvGetPath () const**

This function obtains the name of the directory.

Returns

The name of the directory.

6.14.3.5 **void lfc1::filesystem::CDirectory::mvRefresh ()**

This function loads and sorts the contents of a directory.

Exceptions

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

6.14.3.6 **CDirectory & lfc1::filesystem::CDirectory::operator= (const CDirectory & arRHS)**

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

Parameters

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

Returns

The [CDirectory](#) object assigned to.

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

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

Parameters

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

Returns

The [CDirectory](#) object assigned to.

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

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

Parameters

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

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

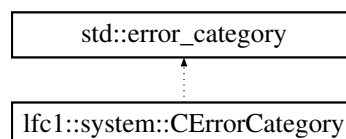
- include/lfc1/filesystem/cdirectory.hpp
- filesystem/library/src/cdirectory.cpp

6.15 lfc1::system::CErrorCategory Class Reference

This class is an error category for this library.

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

Inheritance diagram for lfc1::system::CErrorCategory:



Public Member Functions

- [CErrorCategory](#) (const [CErrorCategory](#) &)=delete
Not supported.
- [CErrorCategory](#) ([CErrorCategory](#) &&)=delete
Not supported.

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

Static Public Member Functions

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

Protected Member Functions

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

6.15.1 Detailed Description

This class is an error category for this library.

Note

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

6.15.2 Member Function Documentation

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

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

Parameters

in	ev	The error code.
----	----	-----------------

Returns

The message equivalent to the given error code.

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

This function obtains the name of the error category.

Returns

The name of the error category.

6.15.2.3 std::error_category & Ifc1::system::CErrCategory::smrGetErrorCategory () [static], [noexcept]

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

Returns

A reference to a single error category object.

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

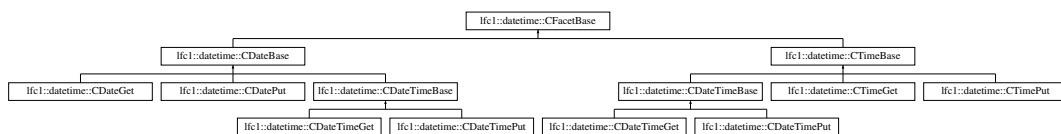
- include/lfc1/system/cerrorcategory.hpp
- system/library/src/cerrorcategory.cpp

6.16 Ifc1::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 Ifc1::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 &arloState, 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 &arlosBase, std::ios_base::iostate &arloState, 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 date and time input/output facets.

6.16.2 Member Function Documentation

- 6.16.2.1 **CFacetBase::TInIt lfc1::datetime::CFacetBase::smvGetNumber** ([TInIt avNext](#), [TInIt avEnd](#), std::ios_base & [arlosBase](#), std::ios_base::iostate & [arloState](#), 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	arlosBase	The stream formatting information.
out	arloState	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.

- 6.16.2.2 **CFacetBase::TInIt lfc1::datetime::CFacetBase::smvIgnoreChar** ([TInIt avNext](#), [TInIt avEnd](#), std::ios_base::iostate & [arloState](#), 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	<i>avNext</i>	The input iterator.
in	<i>avEnd</i>	The input end iterator.
out	<i>arIoState</i>	The stream state.
in	<i>avExpectedChar</i>	The next character expected from the input iterator.

Return values

<i>ios_base::goodbit</i>	The expected character was found.
<i>ios_base::failbit</i>	The expected character was not found.

6.16.2.3 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	<i>avOut</i>	The output iterator.
in	<i>avNumber</i>	The 1-digit number.

Returns

The output iterator.

6.16.2.4 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	<i>avOut</i>	The output iterator.
in	<i>avNumber</i>	The 2-digit number.

Returns

The output iterator.

6.16.2.5 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	<i>avOut</i>	The output iterator.
in	<i>avNumber</i>	The 2-digit number.

Returns

The output iterator.

6.16.2.6 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.

6.16.2.7 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

6.17 lfc1::numeric::CInteger< IT > Class Template Reference

This class template represents little endian integers on big endian operating systems or big endian integers on little endian operating systems.

```
#include <lfc1/numeric/cinteger.hpp>
```

Public Member Functions

- [CInteger](#) () noexcept
This template function creates a default CInteger<IT> object.
- [CInteger](#) (IT avInteger) noexcept
This template function creates a CInteger<IT> object and assigns the given value to it.
- [CInteger](#) (const [CInteger](#) &arRHS) noexcept
This template function copy constructs a CInteger<IT> object.
- [CInteger](#) ([CInteger](#) &&arRHS) noexcept
This template function move constructs a CInteger<IT> object.
- [~CInteger](#) () noexcept
This template function destroys a CInteger<IT> object.
- [CInteger](#) & [operator=](#) (IT avInteger) noexcept

- This template function assigns the given value to a CInteger<IT> object.*
 - `CInteger & operator= (const CInteger &arRHS) noexcept`
- This template function assigns a CInteger<IT> object to another CInteger<IT> object.*
 - `CInteger & operator= (CInteger &&arRHS) noexcept`
- This template function moves a CInteger<IT> object to another CInteger<IT> object.*
 - `operator IT () const noexcept`
- This template function converts a CInteger<IT> object to its corresponding integer type.*
 - `void swap (CInteger &arRHS) noexcept`
- This template function swaps a CInteger<IT> object with another CInteger<IT> object.*
 - `CInteger & operator++ () noexcept`
- This template function pre-increments a CInteger<IT> object.*
 - `CInteger operator++ (int) noexcept`
- This template function post-increments a CInteger<IT> object.*
 - `CInteger & operator-- () noexcept`
- This template function pre-decrements a CInteger<IT> object.*
 - `CInteger operator-- (int) noexcept`
- This template function post-decrements a CInteger<IT> object.*
 - `CInteger & operator+= (const CInteger &arRHS) noexcept`
- This template function adds another CInteger<IT> object to a CInteger<IT> object.*
 - `CInteger & operator-= (const CInteger &arRHS) noexcept`
- This template function subtracts another CInteger<IT> object from a CInteger<IT> object.*
 - `CInteger & operator*= (const CInteger &arRHS) noexcept`
- This template function multiplies a CInteger<IT> object by another CInteger<IT> object.*
 - `CInteger & operator/= (const CInteger &arRHS) noexcept`
- This template function divides a CInteger<IT> object by another CInteger<IT> object.*
 - `CInteger & operator&= (const CInteger &arRHS) noexcept`
- This template function logical ANDs a CInteger<IT> object with another CInteger<IT> object.*
 - `CInteger & operator|= (const CInteger &arRHS) noexcept`
- This template function logical ORs a CInteger<IT> object with another CInteger<IT> object.*
 - `CInteger & operator^= (const CInteger &arRHS) noexcept`
- This template function logical XORs a CInteger<IT> object with another CInteger<IT> object.*
 - `CInteger & operator<<= (const CInteger &arRHS) noexcept`
- This template function left shifts a CInteger<IT> object by the number of bits given by another CInteger<IT> object.*
 - `CInteger & operator>>= (const CInteger &arRHS) noexcept`
- This template function right shifts a CInteger<IT> object by the number of bits given by another CInteger<IT> object.*

6.17.1 Detailed Description

```
template<typename IT>class lfc1::numeric::CInteger< IT >
```

This class template represents little endian integers on big endian operating systems or big endian integers on little endian operating systems.

This class supports the following operators:

```
binary arithmetic operators: +=, -=, *=, /=
unary arithmetic operators: ++, --
binary bitwise operators:   &=, |=, ^=
shift operators:           <<=, >>=
```

Template Parameters

<i>IT</i>	The integer type.
-----------	-------------------

6.17.2 Constructor & Destructor Documentation

6.17.2.1 `template<typename IT > lfc1::numeric::CInteger< IT >::CInteger () [noexcept]`

This template function creates a default CInteger<IT> object.

Template Parameters

<i>IT</i>	The integer type.
-----------	-------------------

6.17.2.2 `template<typename IT > lfc1::numeric::CInteger< IT >::CInteger (IT avInteger) [noexcept]`

This template function creates a CInteger<IT> object and assigns the given value to it.

Template Parameters

<i>IT</i>	The integer type.
-----------	-------------------

Parameters

<i>in</i>	<i>avInteger</i>	The value to assign to the CInteger<IT> object.
-----------	------------------	---

6.17.2.3 `template<typename IT > lfc1::numeric::CInteger< IT >::CInteger (const CInteger< IT > & arRHS) [noexcept]`

This template function copy constructs a CInteger<IT> object.

Template Parameters

<i>IT</i>	The integer type.
-----------	-------------------

Parameters

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

6.17.2.4 `template<typename IT > lfc1::numeric::CInteger< IT >::CInteger (CInteger< IT > && arRHS) [noexcept]`

This template function move constructs a CInteger<IT> object.

Template Parameters

<i>IT</i>	The integer type.
-----------	-------------------

Parameters

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

6.17.2.5 `template<typename IT > lfc1::numeric::CInteger< IT >::~~CInteger () [noexcept]`

This template function destroys a CInteger<IT> object.

Template Parameters

<i>IT</i>	The integer type.
-----------	-------------------

6.17.3 Member Function Documentation

6.17.3.1 `template<typename IT > lfc1::numeric::CInteger< IT >::operator IT () const` `[noexcept]`

This template function converts a CInteger<IT> object to its corresponding integer type.

Template Parameters

<i>IT</i>	The integer type.
-----------	-------------------

Returns

The value of the integer type.

6.17.3.2 `template<typename IT > CInteger< IT > & lfc1::numeric::CInteger< IT >::operator&= (const CInteger< IT > & arRHS)` `[noexcept]`

This template function logical ANDs a CInteger<IT> object with another CInteger<IT> object.

Template Parameters

<i>IT</i>	The integer type.
-----------	-------------------

Parameters

<i>in</i>	<i>arRHS</i>	The other CInteger<IT> object.
-----------	--------------	--------------------------------

Returns

The CInteger<IT> object.

6.17.3.3 `template<typename IT > CInteger< IT > & lfc1::numeric::CInteger< IT >::operator*= (const CInteger< IT > & arRHS)` `[noexcept]`

This template function multiplies a CInteger<IT> object by another CInteger<IT> object.

Template Parameters

<i>IT</i>	The integer type.
-----------	-------------------

Parameters

<i>in</i>	<i>arRHS</i>	The other CInteger<IT> object.
-----------	--------------	--------------------------------

Returns

The CInteger<IT> object.

6.17.3.4 `template<typename IT > CInteger< IT > & lfc1::numeric::CInteger< IT >::operator++ ()` [noexcept]

This template function pre-increments a CInteger<IT> object.

Template Parameters

<i>IT</i>	The integer type.
-----------	-------------------

Returns

The CInteger<IT> object.

6.17.3.5 `template<typename IT > CInteger< IT > lfc1::numeric::CInteger< IT >::operator++ (int)` [noexcept]

This template function post-increments a CInteger<IT> object.

Template Parameters

<i>IT</i>	The integer type.
-----------	-------------------

Returns

The value of the CInteger<IT> object before the increment.

6.17.3.6 `template<typename IT > CInteger< IT > & lfc1::numeric::CInteger< IT >::operator+= (const CInteger< IT > & arRHS)` [noexcept]

This template function adds another CInteger<IT> object to a CInteger<IT> object.

Template Parameters

<i>IT</i>	The integer type.
-----------	-------------------

Parameters

<i>in</i>	<i>arRHS</i>	The other CInteger<IT> object.
-----------	--------------	--------------------------------

Returns

The CInteger<IT> object.

6.17.3.7 `template<typename IT > CInteger< IT > & lfc1::numeric::CInteger< IT >::operator-- ()` [noexcept]

This template function pre-decrements a CInteger<IT> object.

Template Parameters

<i>IT</i>	The integer type.
-----------	-------------------

Returns

The CInteger<IT> object.

6.17.3.8 `template<typename IT > CInteger< IT > lfc1::numeric::CInteger< IT >::operator-- (int)` [noexcept]

This template function post-decrements a CInteger<IT> object.

Template Parameters

<i>IT</i>	The integer type.
-----------	-------------------

Returns

The value of the CInteger<IT> object before the decrement.

6.17.3.9 `template<typename IT > CInteger< IT > & lfc1::numeric::CInteger< IT >::operator= (const CInteger< IT > & arRHS)` [noexcept]

This template function subtracts another CInteger<IT> object from a CInteger<IT> object.

Template Parameters

<i>IT</i>	The integer type.
-----------	-------------------

Parameters

in	<i>arRHS</i>	The other CInteger<IT> object.
----	--------------	--------------------------------

Returns

The CInteger<IT> object.

6.17.3.10 `template<typename IT > CInteger< IT > & lfc1::numeric::CInteger< IT >::operator/= (const CInteger< IT > & arRHS)` [noexcept]

This template function divides a CInteger<IT> object by another CInteger<IT> object.

Template Parameters

<i>IT</i>	The integer type.
-----------	-------------------

Parameters

in	<i>arRHS</i>	The other CInteger<IT> object.
----	--------------	--------------------------------

Returns

The CInteger<IT> object.

6.17.3.11 `template<typename IT > CInteger< IT > & lfc1::numeric::CInteger< IT >::operator<<= (const CInteger< IT > & arRHS)` [noexcept]

This template function left shifts a CInteger<IT> object by the number of bits given by another CInteger<IT> object.

Template Parameters

<i>IT</i>	The integer type.
-----------	-------------------

Parameters

<i>in</i>	<i>arRHS</i>	The other CInteger<IT> object.
-----------	--------------	--------------------------------

Returns

The CInteger<IT> object.

6.17.3.12 `template<typename IT > CInteger< IT > & lfc1::numeric::CInteger< IT >::operator= (IT avInteger)`
`[noexcept]`

This template function assigns the given value to a CInteger<IT> object.

Template Parameters

<i>IT</i>	The integer type.
-----------	-------------------

Parameters

<i>in</i>	<i>avInteger</i>	The value to assign to the CInteger<IT> object.
-----------	------------------	---

Returns

The CInteger<IT> object assigned to.

6.17.3.13 `template<typename IT > CInteger< IT > & lfc1::numeric::CInteger< IT >::operator= (const CInteger< IT > & arRHS)`
`[noexcept]`

This template function assigns a CInteger<IT> object to another CInteger<IT> object.

Template Parameters

<i>IT</i>	The integer type.
-----------	-------------------

Parameters

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

Returns

The CInteger<IT> object assigned to.

6.17.3.14 `template<typename IT > CInteger< IT > & lfc1::numeric::CInteger< IT >::operator= (CInteger< IT > && arRHS)`
`[noexcept]`

This template function moves a CInteger<IT> object to another CInteger<IT> object.

Template Parameters

<i>IT</i>	The integer type.
-----------	-------------------

Parameters

in	<i>arRHS</i>	The CInteger<IT> object to move to another CInteger<IT> object.
----	--------------	---

Returns

The CInteger<IT> object assigned to.

6.17.3.15 `template<typename IT > CInteger< IT > & lfc1::numeric::CInteger< IT >::operator>= (const CInteger< IT > & arRHS) [noexcept]`

This template function right shifts a CInteger<IT> object by the number of bits given by another CInteger<IT> object.

Template Parameters

<i>IT</i>	The integer type.
-----------	-------------------

Parameters

in	<i>arRHS</i>	The other CInteger<IT> object.
----	--------------	--------------------------------

Returns

The CInteger<IT> object.

6.17.3.16 `template<typename IT > CInteger< IT > & lfc1::numeric::CInteger< IT >::operator^= (const CInteger< IT > & arRHS) [noexcept]`

This template function logical XORs a CInteger<IT> object with another CInteger<IT> object.

Template Parameters

<i>IT</i>	The integer type.
-----------	-------------------

Parameters

in	<i>arRHS</i>	The other CInteger<IT> object.
----	--------------	--------------------------------

Returns

The CInteger<IT> object.

6.17.3.17 `template<typename IT > CInteger< IT > & lfc1::numeric::CInteger< IT >::operator|= (const CInteger< IT > & arRHS) [noexcept]`

This template function logical ORs a CInteger<IT> object with another CInteger<IT> object.

Template Parameters

<i>IT</i>	The integer type.
-----------	-------------------

Parameters

in	arRHS	The other CInteger<IT> object.
----	-------	--------------------------------

Returns

The CInteger<IT> object.

6.17.3.18 `template<typename IT > void lfc1::numeric::CInteger< IT >::swap (CInteger< IT > & arRHS)`
`[noexcept]`

This template function swaps a CInteger<IT> object with another CInteger<IT> object.

Template Parameters

IT	The integer type.
----	-------------------

Parameters

in, out	arRHS	The CInteger<IT> object to swap with.
---------	-------	---------------------------------------

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

- include/lfc1/numeric/cinteger.hpp

6.18 lfc1::datetime::CLangInfo Class Reference

This class obtains locale specific date and time strings.

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

Public Member Functions

- [CLangInfo](#) (const [CLangInfo](#) &)=delete
Not supported.
- [CLangInfo](#) ([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.

6.18.1 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.

6.18.2 Member Function Documentation

6.18.2.1 std::string lfc1::datetime::CLangInfo::smvGetAMString () [static]

This function obtains the string for AM.

Returns

The string for AM.

6.18.2.2 std::string lfc1::datetime::CLangInfo::smvGetDateFormat () [static]

This function obtains the date format.

Returns

The date format.

6.18.2.3 std::string lfc1::datetime::CLangInfo::smvGetDateTimeFormat () [static]

This function obtains the date and time format.

Returns

The date and time format.

6.18.2.4 std::string lfc1::datetime::CLangInfo::smvGetLongDayName (int avWeekDay) [static]

This function obtains the long name of the specified week day.

Parameters

in	<i>avWeekDay</i>	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

<i>std::out_of_range</i>	Indicates the given week day is out-of-range.
--------------------------	---

6.18.2.5 `std::string lfc1::datetime::CLangInfo::smvGetLongMonthName (int avMonth) [static]`

This function obtains the long name of the specified month.

Parameters

in	<i>avMonth</i>	The number of the month in the range 1 to 12 with 1 being January.
----	----------------	--

Returns

The long name of the specified month.

Exceptions

<i>std::out_of_range</i>	Indicates the given month is out-of-range.
--------------------------	--

6.18.2.6 `std::string lfc1::datetime::CLangInfo::smvGetPMString () [static]`

This function obtains the string for PM.

Returns

The string for PM.

6.18.2.7 `std::string lfc1::datetime::CLangInfo::smvGetShortDayName (int avWeekDay) [static]`

This function obtains the short name of the specified week day.

Parameters

in	<i>avWeekDay</i>	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

<i>std::out_of_range</i>	Indicates the given week day is out-of-range.
--------------------------	---

6.18.2.8 `std::string lfc1::datetime::CLangInfo::smvGetShortMonthName (int avMonth) [static]`

This function obtains the short name of the specified month.

Parameters

<code>in</code>	<code>avMonth</code>	The number of the month in the range 1 to 12 with 1 being January.
-----------------	----------------------	--

Returns

The short name of the specified month.

Exceptions

<code>std::out_of_range</code>	Indicates the given month is out-of-range.
--------------------------------	--

6.18.2.9 `std::string lfc1::datetime::CLangInfo::smvGetTime12Format () [static]`

This function obtains the 12-hour time format.

Returns

The 12-hour time format.

6.18.2.10 `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`

6.19 lfc1::udtsup::CManipBase< T1 > Class Template Reference

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

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

Public Types

- `typedef void(* TManipFunc)(std::ios_base &, T1)`
Single argument manipulator signature.

Public Member Functions

- `CManipBase (TManipFunc apManipFunc, T1 &arArg)`
This template function creates a default CManipBase object.

- [CManipBase](#) (const [CManipBase](#) &)=default
Uses default implementation.
- virtual [~CManipBase](#) () noexcept
This template function destroys a [CManipBase](#) object.
- [CManipBase](#) & operator= (const [CManipBase](#) &)=default
Uses default implementation.

Friends

- template<typename T2 >
void [gvDoManip](#) (std::basic_ios< char > &arIos, const [CManipBase](#)< T2 > &arManipBase)
This template function calls the user-defined manipulator function and handles exceptions from it.

6.19.1 Detailed Description

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

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

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

Template Parameters

<i>T1</i>	Manipulator function argument type.
-----------	-------------------------------------

6.19.2 Constructor & Destructor Documentation

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

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

Template Parameters

<i>T1</i>	Manipulator function argument type.
-----------	-------------------------------------

Parameters

in	<i>apManipFunc</i>	The user-defined manipulator function.
in	<i>arArg</i>	The user-defined manipulator function argument.

6.19.2.2 template<typename T1 > lfc1::udtsup::CManipBase< T1 >::~~CManipBase () [virtual],
[noexcept]

This template function destroys a [CManipBase](#) object.

Template Parameters

<i>T1</i>	Manipulator function argument type.
-----------	-------------------------------------

6.19.3 Friends And Related Function Documentation

6.19.3.1 `template<typename T1> template<typename T2 > void gvDoManip (std::basic_ios< char > & arlos, const CManipBase< T2 > & arManipBase) [friend]`

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

Template Parameters

<i>T2</i>	The user-defined manipulator function argument type.
-----------	--

Parameters

<i>arlos</i>	The stream's state object.
<i>arManipBase</i>	The user-defined manipulator function container.

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

- include/lfc1/udtsup/cmanipbase.hpp

6.20 lfc1::filesystem::CRecDirectory Class Reference

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

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

Public Types

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

Public Member Functions

- [CRecDirectory](#) (const boost::filesystem::path &arPath)
This function creates a default [CRecDirectory](#) object.
- [CRecDirectory](#) (const [CRecDirectory](#) &arRHS)
This function copy constructs a [CRecDirectory](#) object.
- [CRecDirectory](#) ([CRecDirectory](#) &&arRHS)
This function move constructs a [CRecDirectory](#) object.
- [~CRecDirectory](#) () noexcept
This function destroys a [CRecDirectory](#) object.
- [CRecDirectory](#) & operator= (const [CRecDirectory](#) &arRHS)
This function assigns a [CRecDirectory](#) object to another [CRecDirectory](#) object.
- [CRecDirectory](#) & operator= ([CRecDirectory](#) &&arRHS) noexcept
This function moves a [CRecDirectory](#) object to another [CRecDirectory](#) object.
- int [mvGetLevelCount](#) () const noexcept
This function obtains the number of directory levels.
- boost::filesystem::path [mvGetPath](#) () const

This function obtains the name of the directory.

- unsigned [mvGetNumEntries](#) () const noexcept

This function obtains the number of directory entries.

- [TIterator begin](#) () const noexcept

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

- [TIterator end](#) () const noexcept

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

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

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

- void [mvRefresh](#) ()

This function loads and sorts the contents of a directory.

6.20.1 Detailed Description

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

6.20.2 Constructor & Destructor Documentation

6.20.2.1 `lfc1::filesystem::CRecDirectory::CRecDirectory (const boost::filesystem::path & arPath)`

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

Parameters

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

6.20.2.2 `lfc1::filesystem::CRecDirectory::CRecDirectory (const CRecDirectory & arRHS)`

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

Parameters

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

6.20.2.3 `lfc1::filesystem::CRecDirectory::CRecDirectory (CRecDirectory && arRHS)`

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

Parameters

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

6.20.3 Member Function Documentation

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

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

Returns

The beginning iterator to the list of directory entries.

Note

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

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

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

Returns

The ending iterator to the list of directory entries.

Note

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

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

This function obtains the number of directory levels.

Returns

The number of directory levels.

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

This function obtains the number of directory entries.

Returns

The number of directory entries.

6.20.3.5 boost::filesystem::path lfc1::filesystem::CRecDirectory::mvGetPath () const

This function obtains the name of the directory.

Returns

The name of the directory.

6.20.3.6 void lfc1::filesystem::CRecDirectory::mvRefresh ()

This function loads and sorts the contents of a directory.

Exceptions

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

6.20.3.7 CRecDirectory & lfc1::filesystem::CRecDirectory::operator= (const CRecDirectory & arRHS)

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

Parameters

in	arRHS	The CRecDirectory object to assign to another CRecDirectory object.
----	-------	---

Returns

The [CRecDirectory](#) object assigned to.

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

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

Parameters

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

Returns

The [CRecDirectory](#) object assigned to.

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

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

Parameters

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

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

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

6.21 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 int `HR_PER_DAY` = 24
The number of hours in a day.
- static const int `MIN_HR` = 0
The lowest hour number in a day.
- static const int `MAX_HR` = 23
The highest hour number in a day.
- static const int `MIN_PER_HR` = 60
The number of minutes in an hour.
- static const int `MIN_MIN` = 0
The lowest minute number in an hour.
- static const int `MAX_MIN` = 59
The highest minute number in an hour.
- static const int `SEC_PER_MIN` = 60
The number of seconds in a minute.
- static const int `MIN_SEC` = 0
The lowest second number in a minute.
- static const int `MAX_SEC` = 59
The highest second number in a minute.
- static const int `MSEC_PER_SEC` = 1000
The number of milliseconds in a second.
- static const int `MIN_MSEC` = 0
The lowest millisecond number in a second.
- static const int `MAX_MSEC` = 999
The highest millisecond number in a second.
- static const int `MSEC_PER_MIN` = `SEC_PER_MIN` * `MSEC_PER_SEC`
The number of millisecs. per minute.
- static const int `MSEC_PER_HR` = `MIN_PER_HR` * `MSEC_PER_MIN`
The number of milliseconds per hour.
- static const int `MSEC_PER_DAY` = `HR_PER_DAY` * `MSEC_PER_HR`
The number of milliseconds per day.
- static const int `SEC_PER_HR` = `MIN_PER_HR` * `SEC_PER_MIN`
The number of seconds per hour.
- static const int `SEC_PER_DAY` = `HR_PER_DAY` * `SEC_PER_HR`
The number of seconds per day.
- static const int `MIN_PER_DAY` = `HR_PER_DAY` * `MIN_PER_HR`
The number of minutes per day.
- static const int `MIN_TIMENO` = 0
The lowest millisecond of a day.
- static const int `MAX_TIMENO` = `MSEC_PER_DAY` - 1
The highest millisecond of a day.

6.21.1 Detailed Description

This class represents the time since midnight.

This class supports the following operators:

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

6.21.2 Constructor & Destructor Documentation

6.21.2.1 Ifc1::datetime::CTime::CTime (int *avTimeNo*)

This function creates a [CTime](#) object using the given Julian day number.

Parameters

in	<i>avTimeNo</i>	The milliseconds since midnight number to assign to the CTime object.
----	-----------------	---

Exceptions

<i>std::out_of_range</i>	Indicates that the given milliseconds since midnight number is not within the valid range.
--------------------------	--

6.21.2.2 Ifc1::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	<i>avHour</i>	The given hour.
in	<i>avMinute</i>	The given minute.
in	<i>avSecond</i>	The given second.
in	<i>avMillisecond</i>	The given millisecond.

6.21.2.3 Ifc1::datetime::CTime::CTime (const [CTime](#) & *arRHS*) [noexcept]

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

Parameters

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

6.21.2.4 Ifc1::datetime::CTime::CTime ([CTime](#) && *arRHS*) [noexcept]

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

Parameters

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

6.21.3 Member Function Documentation

6.21.3.1 [CTime::STime](#) Ifc1::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.

6.21.3.2 `std::ios_base::iostate lfc1::datetime::CTime::mvInput (std::istream & arStream)`

This function receives a [CTime](#) object from the given input stream.

Parameters

<code>in</code>	<code><i>arStream</i></code>	The source stream.
-----------------	------------------------------	--------------------

Returns

The resulting stream state.

6.21.3.3 `std::ios_base::iostate lfc1::datetime::CTime::mvOutput (std::ostream & arStream) const`

This function sends a [CTime](#) object to the given output stream.

Parameters

<code>in</code>	<code><i>arStream</i></code>	The destination stream.
-----------------	------------------------------	-------------------------

Returns

The resulting stream state.

6.21.3.4 `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

<code>in</code>	<code><i>avHour</i></code>	The given hour.
<code>in</code>	<code><i>avMinute</i></code>	The given minute.
<code>in</code>	<code><i>avSecond</i></code>	The given second.
<code>in</code>	<code><i>avMillisecond</i></code>	The given millisecond.

6.21.3.5 `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.

6.21.3.6 `CTime & lfc1::datetime::CTime::operator++ ()`

This function pre-increments the [CTime](#) object by one millisecond.

Returns

The [CTime](#) object.

Exceptions

<i>std::out_of_range</i>	Indicates incrementing the time will make the time go past the maximum supported time.
--------------------------	--

6.21.3.7 CTime lfc1::datetime::CTime::operator++ (int)

This function post-increments the [CTime](#) object by one millisecond.

Returns

The [CTime](#) object before the increment.

6.21.3.8 CTime & lfc1::datetime::CTime::operator+= (const CTimeDuration & arTimeDuration)

This function increments the [CTime](#) object by the given duration.

Parameters

<i>in</i>	<i>arTimeDuration</i>	The duration to add to the time.
-----------	-----------------------	----------------------------------

Returns

The [CTime](#) object.

Exceptions

<i>std::out_of_range</i>	Indicates incrementing the time will make the time go past the maximum supported time.
--------------------------	--

6.21.3.9 CTime & lfc1::datetime::CTime::operator-- ()

This function pre-decrements the [CTime](#) object by one millisecond.

Returns

The [CTime](#) object.

Exceptions

<i>std::out_of_range</i>	Indicates incrementing the time will make the time go below the minimum supported time.
--------------------------	---

6.21.3.10 CTime lfc1::datetime::CTime::operator-- (int)

This function post-decrements the [CTime](#) object by one millisecond.

Returns

The [CTime](#) object before the decrement.

6.21.3.11 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.
-------------------	---

6.21.3.12 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 midnight number to assign to the CTime object.
----	----------	---

Returns

The CTime object assigned to.

6.21.3.13 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.

6.21.3.14 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.

6.21.3.15 `int Ifc1::datetime::CTime::smvGetIndex () [static], [noexcept]`

This function obtains an iword/pword index.

Returns

The iword/pword index.

6.21.3.16 `void Ifc1::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	<i>avHour</i>	The hour of a time. (0 - 23)
in	<i>avMinute</i>	The minute of a time. (0 - 59)
in	<i>avSecond</i>	The second of a time. (0 - 59)
in	<i>avMillisecond</i>	The millisecond of a time. (0 - 999)

Exceptions

<i>std::out_of_range</i>	Indicates either the hour, minute, second or millisecond are not within valid ranges.
--------------------------	---

6.21.3.17 `void Ifc1::datetime::CTime::swap (CTime & arRHS) [noexcept]`

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

Parameters

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

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

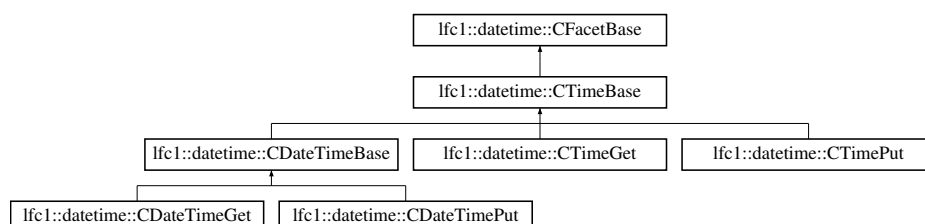
- include/lfc1/datetime/ctime.hpp
- datetime/library/src/ctime.cpp

6.22 Ifc1::datetime::CTimeBase Class Reference

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

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

Inheritance diagram for Ifc1::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.
- virtual [~CTimeBase](#) () noexcept
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 &arLosBase, std::ios_base::iostate &arloState, 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 &arloState, 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 &arLosBase, std::ios_base::iostate &arloState, 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.

6.22.1 Detailed Description

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

6.22.2 Member Function Documentation

6.22.2.1 `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	<i>avFormat</i>	The time format to be expanded.
----	-----------------	---------------------------------

Returns

The expanded format string.

6.22.2.2 `CTimeBase::TInIt lfc1::datetime::CTimeBase::smvGetAmPm (TInIt avNext, TInIt avEnd, std::ios_base & arlosBase, std::ios_base::iostate & arloState, bool & arHasAmPm, bool & arIsPm)` `[static]`, `[protected]`

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

Parameters

in	<i>avNext</i>	The input iterator.
in	<i>avEnd</i>	The input end iterator.
in	<i>arlosBase</i>	The stream formatting information.
out	<i>arloState</i>	The stream state.
out	<i>arHasAmPm</i>	The storage for the resulting has AM/PM indicator (true) or not (false).
out	<i>arIsPm</i>	The storage for the resulting PM (true) or AM (false) indicator.

Returns

The input iterator.

6.22.2.3 `CFacetBase::TInIt lfc1::datetime::CFacetBase::smvGetNumber (TInIt avNext, TInIt avEnd, std::ios_base & arlosBase, std::ios_base::iostate & arloState, int * apNumber, int avDigits, char avPadding)` `[static]`, `[protected]`, `[noexcept]`, `[inherited]`

This function parses an input iterator for an integer value.

Parameters

in	<i>avNext</i>	The input iterator.
in	<i>avEnd</i>	The input end iterator.
in	<i>arlosBase</i>	The stream formatting information.
out	<i>arloState</i>	The stream state.
out	<i>apNumber</i>	The storage for the resulting integer.
in	<i>avDigits</i>	The maximum number of digits allowed for the integer.
in	<i>avPadding</i>	The padding character allowed for the integer value.

Return values

<i>ios_base::goodbit</i>	A valid integer was found.
<i>ios_base::failbit</i>	A valid integer was not found.

6.22.2.4 CFacetBase::TInIt lfc1::datetime::CFacetBase::smvIgnoreChar (TInIt *avNext*, TInIt *avEnd*, std::ios_base::iostate & *arloState*, 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	<i>avNext</i>	The input iterator.
in	<i>avEnd</i>	The input end iterator.
out	<i>arloState</i>	The stream state.
in	<i>avExpectedChar</i>	The next character expected from the input iterator.

Return values

<i>ios_base::goodbit</i>	The expected character was found.
<i>ios_base::failbit</i>	The expected character was not found.

6.22.2.5 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	<i>avOut</i>	The output iterator.
in	<i>avHour</i>	The given hour.

Returns

The output iterator.

6.22.2.6 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	<i>avOut</i>	The output iterator.
in	<i>avNumber</i>	The 1-digit number.

Returns

The output iterator.

6.22.2.7 CFacetBase::TOutIt Ifc1::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	<i>avOut</i>	The output iterator.
in	<i>avNumber</i>	The 2-digit number.

Returns

The output iterator.

6.22.2.8 CFacetBase::TOutIt Ifc1::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	<i>avOut</i>	The output iterator.
in	<i>avNumber</i>	The 2-digit number.

Returns

The output iterator.

6.22.2.9 CFacetBase::TOutIt Ifc1::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	<i>avOut</i>	The output iterator.
in	<i>avNumber</i>	The 3-digit number.

Returns

The output iterator.

6.22.2.10 CFacetBase::TOutIt Ifc1::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	<i>avOut</i>	The output iterator.
in	<i>avNumber</i>	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

6.23 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.

6.23.1 Detailed Description

This class represents time durations.

The purpose of this class is to support the increment and decrement operators of the [CTime](#) class.

6.23.2 Constructor & Destructor Documentation

6.23.2.1 `lfc1::datetime::CTimeDuration::CTimeDuration (int avDuration, EUnits avUnit = E_U_MILLISECOND)`
`[noexcept]`

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

Parameters

in	<i>avDuration</i>	The time duration.
in	<i>avUnit</i>	The unit of measure of the time duration.

6.23.2.2 `lfc1::datetime::CTimeDuration::CTimeDuration (const CTimeDuration & arRHS)` `[noexcept]`

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

Parameters

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

6.23.2.3 `lfc1::datetime::CTimeDuration::CTimeDuration (CTimeDuration && arRHS)` `[noexcept]`

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

Parameters

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

6.23.3 Member Function Documentation

6.23.3.1 `int lfc1::datetime::CTimeDuration::mvGetDuration () const` `[noexcept]`

This function obtains the time duration.

Returns

The time duration.

6.23.3.2 `CTimeDuration::EUnits lfc1::datetime::CTimeDuration::mvGetUnit () const` `[noexcept]`

This function obtains the unit of the time duration.

Returns

The unit of the time duration.

6.23.3.3 `CTimeDuration & lfc1::datetime::CTimeDuration::operator= (const CTimeDuration & arRHS)` `[noexcept]`

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

Parameters

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

Returns

The [CTimeDuration](#) object assigned to.

6.23.3.4 [CTimeDuration](#) & [lfc1::datetime::CTimeDuration::operator=](#) ([CTimeDuration](#) && *arRHS*) [noexcept]

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

Parameters

in	<i>arRHS</i>	The CTimeDuration object to move to another CTimeDuration object.
----	--------------	---

Returns

The [CTimeDuration](#) object assigned to.

6.23.3.5 void [lfc1::datetime::CTimeDuration::swap](#) ([CTimeDuration](#) & *arRHS*) [noexcept]

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

Parameters

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

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

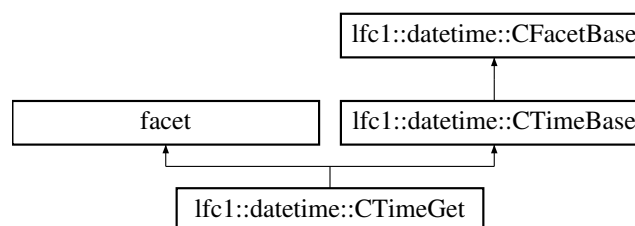
- include/lfc1/datetime/ctime.hpp
- datetime/library/src/ctimeduration.cpp

6.24 [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.
- virtual [~CTimeGet](#) () noexcept
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 &arlosBase, std::ios_base::iostate &arloState, [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 &arlosBase, std::ios_base::iostate &arloState, [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 &arlosBase, std::ios_base::iostate &arloState, 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 &arloState, 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 &arlosBase, std::ios_base::iostate &arloState, 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.

6.24.1 Detailed Description

This class is a [CTime](#) class input facet.

Note

This class uses the Template Method design pattern.

6.24.2 Constructor & Destructor Documentation

6.24.2.1 [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)
----	-----------	--

6.24.3 Member Function Documentation

6.24.3.1 [CTimeGet::TInIt lfc1::datetime::CTimeGet::mvDoGet](#) ([TInIt](#) avNext, [TInIt](#) avEnd, [std::ios_base & arlosBase](#), [std::ios_base::iostate & arloState](#), [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	arlosBase	The stream formatting information.
out	arloState	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.

6.24.3.2 [CTimeGet::TInIt lfc1::datetime::CTimeGet::mvGet](#) ([TInIt](#) avNext, [TInIt](#) avEnd, [std::ios_base & arlosBase](#), [std::ios_base::iostate & arloState](#), [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	<i>avNext</i>	The input stream iterator.
in	<i>avEnd</i>	The input stream end iterator.
in	<i>arlosBase</i>	The stream formatting information.
out	<i>arloState</i>	The stream state. Set to <code>ios_base::failbit</code> if parsing failed.
out	<i>arTime</i>	The time retrieved from the input stream.
in	<i>arFormat</i>	The format of the time retrieved from the input stream.

Returns

The input iterator.

6.24.3.3 `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	<i>avFormat</i>	The time format to be expanded.
----	-----------------	---------------------------------

Returns

The expanded format string.

6.24.3.4 `CTimeBase::TInIt lfc1::datetime::CTimeBase::smvGetAmPm (TInIt avNext, TInIt avEnd, std::ios_base & arlosBase, std::ios_base::iostate & arloState, bool & arHasAmPm, bool & arIsPm)` `[static]`, `[protected]`, `[inherited]`

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

Parameters

in	<i>avNext</i>	The input iterator.
in	<i>avEnd</i>	The input end iterator.
in	<i>arlosBase</i>	The stream formatting information.
out	<i>arloState</i>	The stream state.
out	<i>arHasAmPm</i>	The storage for the resulting has AM/PM indicator (true) or not (false).
out	<i>arIsPm</i>	The storage for the resulting PM (true) or AM (false) indicator.

Returns

The input iterator.

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

This function parses an input iterator for an integer value.

Parameters

in	<i>avNext</i>	The input iterator.
in	<i>avEnd</i>	The input end iterator.
in	<i>arlosBase</i>	The stream formatting information.
out	<i>arloState</i>	The stream state.
out	<i>apNumber</i>	The storage for the resulting integer.
in	<i>avDigits</i>	The maximum number of digits allowed for the integer.
in	<i>avPadding</i>	The padding character allowed for the integer value.

Return values

<i>ios_base::goodbit</i>	A valid integer was found.
<i>ios_base::failbit</i>	A valid integer was not found.

6.24.3.6 CFacetBase::TInIt lfc1::datetime::CFacetBase::smvIgnoreChar (TInIt avNext, TInIt avEnd, std::ios_base::iostate & arloState, 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	<i>avNext</i>	The input iterator.
in	<i>avEnd</i>	The input end iterator.
out	<i>arloState</i>	The stream state.
in	<i>avExpectedChar</i>	The next character expected from the input iterator.

Return values

<i>ios_base::goodbit</i>	The expected character was found.
<i>ios_base::failbit</i>	The expected character was not found.

6.24.3.7 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	<i>avOut</i>	The output iterator.
in	<i>avHour</i>	The given hour.

Returns

The output iterator.

6.24.3.8 CFacetBase::TOutIt Ifc1::datetime::CFacetBase::smvPutNumber1 (TOutIt *avOut*, int *avNumber*) [static], [protected], [noexcept], [inherited]

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

Parameters

in	<i>avOut</i>	The output iterator.
in	<i>avNumber</i>	The 1-digit number.

Returns

The output iterator.

6.24.3.9 CFacetBase::TOutIt Ifc1::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	<i>avOut</i>	The output iterator.
in	<i>avNumber</i>	The 2-digit number.

Returns

The output iterator.

6.24.3.10 CFacetBase::TOutIt Ifc1::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	<i>avOut</i>	The output iterator.
in	<i>avNumber</i>	The 2-digit number.

Returns

The output iterator.

6.24.3.11 CFacetBase::TOutIt Ifc1::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	<i>avOut</i>	The output iterator.
in	<i>avNumber</i>	The 3-digit number.

Returns

The output iterator.

6.24.3.12 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:

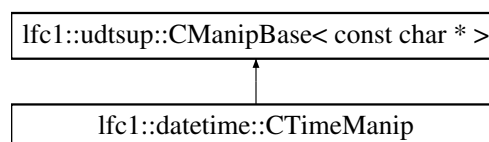
- include/lfc1/datetime/ctime.hpp
- datetime/library/src/ctimeget.cpp

6.25 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 Types**

- typedef void(* [TManipFunc](#))(std::ios_base &, const char *)
Single argument manipulator signature.

Public Member Functions

- [CTimeManip](#) (const char *apFormat)
This function creates a [CTimeManip](#) object.
- [CTimeManip](#) (const [CTimeManip](#) &)=default
Uses default implementation.
- [~CTimeManip](#) () noexcept
This function destroys a [CTimeManip](#) object.
- [CTimeManip](#) & [operator=](#) (const [CTimeManip](#) &)=default
Uses default implementation.

Static Public Member Functions

- static std::string [smvGetFormat](#) (std::ios_base &arlosBase)
This function obtains the time format from a stream.
- static std::ios_base::iostate [smvCopyFmtErr](#) (std::basic_ios< char > &arlos)
This function obtains the error information relating to time format copying.

6.25.1 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.

6.25.2 Constructor & Destructor Documentation

6.25.2.1 lfc1::datetime::CTimeManip::CTimeManip (const char * *apFormat*)

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

Parameters

in	<i>apFormat</i>	The desired time format.
----	-----------------	--------------------------

6.25.3 Member Function Documentation

6.25.3.1 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	<i>arlos</i>	The stream containing the error information.
----	--------------	--

Returns

The error information relating to time format copying.

6.25.3.2 std::string lfc1::datetime::CTimeManip::smvGetFormat (std::ios_base & *arlosBase*) [static]

This function obtains the time format from a stream.

Parameters

in	<i>arlosBase</i>	The stream containing the time format.
----	------------------	--

Returns

The time format.

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

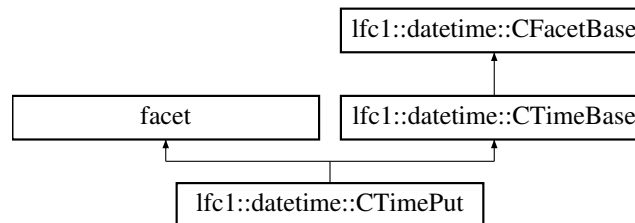
- include/lfc1/datetime/ctime.hpp
- datetime/library/src/ctimemanip.cpp

6.26 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.
- virtual [~CTimePut](#) () noexcept
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 [TInlt smvGetAmPm](#) ([TInlt](#) avNext, [TInlt](#) avEnd, std::ios_base &arlosBase, std::ios_base::iostate &arloState, bool &arHasAmPm, bool &arIsPm)
This function parses an input iterator for an AM/PM indicator.
- static [TOutlt smvPutAmPm](#) ([TOutlt](#) avOut, int avHour)
This function sends AM or PM to an output iterator based on the given hour.
- static [TInlt smvIgnoreChar](#) ([TInlt](#) avNext, [TInlt](#) avEnd, std::ios_base::iostate &arloState, 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 [TInlt smvGetNumber](#) ([TInlt](#) avNext, [TInlt](#) avEnd, std::ios_base &arlosBase, std::ios_base::iostate &arloState, int *apNumber, int avDigits, char avPadding) noexcept
This function parses an input iterator for an integer value.
- static [TOutlt smvPutNumber1](#) ([TOutlt](#) avOut, int avNumber) noexcept
This function sends a 1-digit number to an output iterator.
- static [TOutlt smvPutNumber2Zero](#) ([TOutlt](#) avOut, int avNumber) noexcept
This function sends a 2-digit zero padded number to an output iterator.
- static [TOutlt smvPutNumber2Blank](#) ([TOutlt](#) avOut, int avNumber) noexcept
This function sends a 2-digit blank padded number to an output iterator.
- static [TOutlt smvPutNumber3Zero](#) ([TOutlt](#) avOut, int avNumber) noexcept
This function sends a 3-digit zero padded number to an output iterator.
- static [TOutlt smvPutNumber4Zero](#) ([TOutlt](#) avOut, int avNumber) noexcept
This function sends a 4-digit zero padded number to an output iterator.

6.26.1 Detailed Description

This class is a [CTime](#) class output facet.

Note

This class uses the Template Method design pattern.

6.26.2 Constructor & Destructor Documentation

6.26.2.1 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)
----	---------------------------	--

6.26.3 Member Function Documentation

6.26.3.1 CTimePut::TOutlt lfc1::datetime::CTimePut::mvDoPut (TOutlt avOut, const CTime & arTime, const std::string & arFormat) const [protected], [virtual]

This function implements the behavior of the [mvPut\(\)](#) function.

Parameters

in	<i>avOut</i>	The output iterator.
in	<i>arTime</i>	The time to be sent to the output stream.
in	<i>arFormat</i>	The format of the time to be sent to the output stream.

Returns

The output iterator.

6.26.3.2 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	<i>avOut</i>	The output iterator.
in	<i>arTime</i>	The time to be sent to the output stream.
in	<i>arFormat</i>	The format of the time to be sent to the output stream.

Returns

The output iterator.

6.26.3.3 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	<i>avFormat</i>	The time format to be expanded.
----	-----------------	---------------------------------

Returns

The expanded format string.

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

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

Parameters

in	<i>avNext</i>	The input iterator.
in	<i>avEnd</i>	The input end iterator.
in	<i>arlosBase</i>	The stream formatting information.
out	<i>arloState</i>	The stream state.
out	<i>arHasAmPm</i>	The storage for the resulting has AM/PM indicator (true) or not (false).
out	<i>arIsPm</i>	The storage for the resulting PM (true) or AM (false) indicator.

Returns

The input iterator.

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

This function parses an input iterator for an integer value.

Parameters

in	<i>avNext</i>	The input iterator.
in	<i>avEnd</i>	The input end iterator.
in	<i>arlosBase</i>	The stream formatting information.
out	<i>arloState</i>	The stream state.
out	<i>apNumber</i>	The storage for the resulting integer.
in	<i>avDigits</i>	The maximum number of digits allowed for the integer.
in	<i>avPadding</i>	The padding character allowed for the integer value.

Return values

<i>ios_base::goodbit</i>	A valid integer was found.
<i>ios_base::failbit</i>	A valid integer was not found.

6.26.3.6 CFacetBase::TInIt lfc1::datetime::CFacetBase::smvIgnoreChar (TInIt *avNext*, TInIt *avEnd*, std::ios_base::iostate & *arloState*, 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	<i>avNext</i>	The input iterator.
in	<i>avEnd</i>	The input end iterator.
out	<i>arloState</i>	The stream state.
in	<i>avExpectedChar</i>	The next character expected from the input iterator.

Return values

<i>ios_base::goodbit</i>	The expected character was found.
<i>ios_base::failbit</i>	The expected character was not found.

6.26.3.7 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	<i>avOut</i>	The output iterator.
in	<i>avHour</i>	The given hour.

Returns

The output iterator.

6.26.3.8 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	<i>avOut</i>	The output iterator.
in	<i>avNumber</i>	The 1-digit number.

Returns

The output iterator.

6.26.3.9 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	<i>avOut</i>	The output iterator.
in	<i>avNumber</i>	The 2-digit number.

Returns

The output iterator.

6.26.3.10 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

<code>in</code>	<code>avOut</code>	The output iterator.
<code>in</code>	<code>avNumber</code>	The 2-digit number.

Returns

The output iterator.

6.26.3.11 `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

<code>in</code>	<code>avOut</code>	The output iterator.
<code>in</code>	<code>avNumber</code>	The 3-digit number.

Returns

The output iterator.

6.26.3.12 `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

<code>in</code>	<code>avOut</code>	The output iterator.
<code>in</code>	<code>avNumber</code>	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`

6.27 `std::numeric_limits< lfc1::numeric::CInteger< IT > >` Class Template Reference

This template class is a specialization for `std::numeric_limits` for `CInteger<IT>`.

```
#include <lfc1/numeric/cinteger.hpp>
```

Static Public Member Functions

- static `lfc1::numeric::CInteger< IT > min ()` `noexcept`
`< Lowest possible value.`

- static `lfc1::numeric::CInteger`
`< IT > max () noexcept`
< Highest possible value.
- static `lfc1::numeric::CInteger`
`< IT > epsilon () noexcept`
Not applicable.
- static `lfc1::numeric::CInteger`
`< IT > round_error () noexcept`
Not applicable.
- static `lfc1::numeric::CInteger`
`< IT > infinity () noexcept`
Not applicable.
- static `lfc1::numeric::CInteger`
`< IT > quiet_NaN () noexcept`
Not applicable.
- static `lfc1::numeric::CInteger`
`< IT > signaling_NaN () noexcept`
Not applicable.
- static `lfc1::numeric::CInteger`
`< IT > denorm_min () noexcept`
Not applicable.

Static Public Attributes

- static const bool `is_specialized` = true
Have information for this type.
- static const int `digits` = `numeric_limits<IT>::digits`
Number of significant bits.
- static const int `digits10` = `numeric_limits<IT>::digits10`
Number of significant digits.
- static const bool `is_signed` = `numeric_limits<IT>::is_signed`
Has sign or not.
- static const bool `is_integer` = true
Is a whole number.
- static const bool `is_exact` = true
Is an exact number.
- static const int `radix` = 2
Uses binary representation.
- static const int `min_exponent` = 0
Not applicable.
- static const int `min_exponent10` = 0
Not applicable.
- static const int `max_exponent` = 0
Not applicable.
- static const int `max_exponent10` = 0
Not applicable.
- static const bool `has_infinity` = false
Not applicable.
- static const bool `has_quiet_NaN` = false
Not applicable.
- static const bool `has_signaling_NaN` = false

- Not applicable.*
- static const float `_denorm_style` [has_denorm](#) = `denorm_absent`
- Not applicable.*
- static const bool [has_denorm_loss](#) = `false`
- Not applicable.*
- static const bool [is_iec559](#) = `false`
- Not applicable.*
- static const bool [is_bounded](#) = `true`
- Has a range of valid values.*
- static const bool [is_modulo](#) = `true`
- Values wrap around on underflow/overflow.*
- static const bool [traps](#) = `false`
- Not applicable.*
- static const bool [tinyness_before](#) = `false`
- Not applicable.*
- static const float `_round_style` [round_style](#) = `round_toward_zero`
- Not applicable.*

6.27.1 Detailed Description

```
template<typename IT>class std::numeric_limits<lfc1::numeric::CInteger< IT > >
```

This template class is a specialization for `std::numeric_limits` for `CInteger<IT>`.

Template Parameters

<i>IT</i>	The integer type.
-----------	-------------------

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

- `include/lfc1/numeric/cinteger.hpp`

6.28 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.

6.28.1 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`

6.29 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.

6.29.1 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`

6.30 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.

6.30.1 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

- ~CInteger
 - lfc1::numeric::CInteger, [114](#)
- ~CManipBase
 - lfc1::udtsup::CManipBase, [124](#)
- begin
 - lfc1::filesystem::CDirectory, [105](#)
 - lfc1::filesystem::CRecDirectory, [126](#)
- CDate
 - lfc1::datetime::CDate, [27](#), [28](#)
- CDateDuration
 - lfc1::datetime::CDateDuration, [39](#)
- CDateGet
 - lfc1::datetime::CDateGet, [43](#)
- CDateManip
 - lfc1::datetime::CDateManip, [49](#)
- CDatePut
 - lfc1::datetime::CDatePut, [51](#)
- CDateTime
 - lfc1::datetime::CDateTime, [59](#)
- CDateTimeDuration
 - lfc1::datetime::CDateTimeDuration, [75](#)
- CDateTimeGet
 - lfc1::datetime::CDateTimeGet, [80](#)
- CDateTimeManip
 - lfc1::datetime::CDateTimeManip, [90](#)
- CDateTimePut
 - lfc1::datetime::CDateTimePut, [94](#)
- CDemangler
 - lfc1::misc::CDemangler, [104](#)
- CDirectory
 - lfc1::filesystem::CDirectory, [105](#)
- CInteger
 - lfc1::numeric::CInteger, [114](#)
- CManipBase
 - lfc1::udtsup::CManipBase, [124](#)
- CRecDirectory
 - lfc1::filesystem::CRecDirectory, [126](#)
- CTime
 - lfc1::datetime::CTime, [131](#)
- CTimeDuration
 - lfc1::datetime::CTimeDuration, [141](#)
- CTimeGet
 - lfc1::datetime::CTimeGet, [144](#)
- CTimeManip
 - lfc1::datetime::CTimeManip, [149](#)
- CTimePut
 - lfc1::datetime::CTimePut, [151](#)
- DATENO_OFFSET_SUN
 - lfc1::datetime::CDate, [32](#)
- Date and time class library, [19](#)
 - operator<<, [20](#), [21](#)
 - operator>>, [21](#), [22](#)
 - setdateformat, [22](#)
 - setdatetimeformat, [22](#)
 - settimeformat, [22](#)
 - swap, [23](#)
- end
 - lfc1::filesystem::CDirectory, [106](#)
 - lfc1::filesystem::CRecDirectory, [127](#)
- FIRST_DAY_MON
 - lfc1::datetime::CDate, [32](#)
- FIRST_DAY_SUN
 - lfc1::datetime::CDate, [32](#)
- Filesystem class library, [18](#)
 - swap, [18](#)
- grCategory
 - System class library, [10](#)
- gvDoManip
 - lfc1::udtsup::CManipBase, [124](#)
 - User-defined type support class library, [11](#)
- gvInput
 - User-defined type support class library, [12](#)
- gvInputGuarded
 - User-defined type support class library, [12](#)
- gvOutput
 - User-defined type support class library, [13](#)
- gvOutputGuarded
 - User-defined type support class library, [13](#)
- lfc1::datetime::CDate, [25](#)
 - CDate, [27](#), [28](#)
 - FIRST_DAY_MON, [32](#)
 - FIRST_DAY_SUN, [32](#)
 - MIN_DATE_YEAR, [32](#)
 - mvGetDate, [28](#)
 - mvInput, [28](#)
 - mvOutput, [28](#)
 - mvSetDate, [28](#)
 - operator int, [29](#)
 - operator++, [29](#)
 - operator+==, [29](#)
 - operator--, [29](#), [30](#)
 - operator-=, [30](#)
 - operator=, [30](#), [31](#)

- smvGetIndex, 31
- smvGetMaxDay, 31
- smvIsLeapYear, 31
- smvValidateDate, 31
- swap, 32
- lfc1::datetime::CDate::SDate, 157
- lfc1::datetime::CDateBase, 33
 - smvCalcWeekBasedDetails, 34
 - smvCalcWeekNo, 34
 - smvExpandFormat, 35
 - smvGetMonthName, 35
 - smvGetNumber, 35
 - smvGetWeekDayName, 36
 - smvIgnoreChar, 36
 - smvPutMonthName, 36
 - smvPutNumber1, 37
 - smvPutNumber2Blank, 37
 - smvPutNumber2Zero, 37
 - smvPutNumber3Zero, 37
 - smvPutNumber4Zero, 38
 - smvPutWeekDayName, 38
- lfc1::datetime::CDateDuration, 38
 - CDateDuration, 39
 - mvGetDuration, 40
 - mvGetUnit, 40
 - operator=, 40
 - swap, 40
- lfc1::datetime::CDateGet, 41
 - CDateGet, 43
 - mvDoGet, 43
 - mvGet, 43
 - smvCalcWeekBasedDetails, 44
 - smvCalcWeekNo, 44
 - smvExpandFormat, 44
 - smvGetMonthName, 45
 - smvGetNumber, 45
 - smvGetWeekDayName, 45
 - smvIgnoreChar, 46
 - smvPutMonthName, 46
 - smvPutNumber1, 46
 - smvPutNumber2Blank, 47
 - smvPutNumber2Zero, 47
 - smvPutNumber3Zero, 47
 - smvPutNumber4Zero, 47
 - smvPutWeekDayName, 48
- lfc1::datetime::CDateManip, 48
 - CDateManip, 49
 - smvCopyFmtErr, 49
 - smvGetFormat, 49
- lfc1::datetime::CDatePut, 50
 - CDatePut, 51
 - mvDoPut, 52
 - mvPut, 52
 - smvCalcWeekBasedDetails, 53
 - smvCalcWeekNo, 53
 - smvExpandFormat, 53
 - smvGetMonthName, 53
 - smvGetNumber, 54
 - smvGetWeekDayName, 54
 - smvIgnoreChar, 54
 - smvPutMonthName, 55
 - smvPutNumber1, 55
 - smvPutNumber2Blank, 55
 - smvPutNumber2Zero, 56
 - smvPutNumber3Zero, 56
 - smvPutNumber4Zero, 56
 - smvPutWeekDayName, 57
- lfc1::datetime::CDateTime, 57
 - CDateTime, 59
 - MAX_DATETIMENO, 64
 - MIN_DATETIMENO, 64
 - mvGetDateTime, 60
 - mvInput, 60
 - mvOutput, 60
 - mvSetDateTime, 60
 - operator long long, 61
 - operator++, 61
 - operator+=", 61
 - operator--, 62
 - operator-=, 62
 - operator=, 62, 63
 - smvGetIndex, 63
 - smvValidateDst, 63
 - swap, 63
- lfc1::datetime::CDateTime::SDateTime, 158
- lfc1::datetime::CDateTimeBase, 64
 - smvCalcWeekBasedDetails, 66
 - smvCalcWeekNo, 67
 - smvExpandFormat, 67
 - smvGetAmPm, 67
 - smvGetMonthName, 67
 - smvGetNumber, 68
 - smvGetTzName, 69
 - smvGetTzOffset, 69
 - smvGetWeekDayName, 69
 - smvIgnoreChar, 70
 - smvPutAmPm, 70
 - smvPutMonthName, 70
 - smvPutNumber1, 71
 - smvPutNumber2Blank, 71, 72
 - smvPutNumber2Zero, 72
 - smvPutNumber3Zero, 72, 73
 - smvPutNumber4Zero, 73
 - smvPutTzName, 73
 - smvPutTzOffset, 74
 - smvPutWeekDayName, 74
- lfc1::datetime::CDateTimeDuration, 74
 - CDateTimeDuration, 75
 - mvGetDuration, 76
 - mvGetUnit, 76
 - operator=, 76
 - swap, 76
- lfc1::datetime::CDateTimeGet, 77
 - CDateTimeGet, 80
 - mvDoGet, 80
 - mvGet, 80

- smvCalcWeekBasedDetails, 81
- smvCalcWeekNo, 81
- smvExpandFormat, 82
- smvGetAmPm, 82
- smvGetMonthName, 82
- smvGetNumber, 83
- smvGetTzName, 83
- smvGetTzOffset, 84
- smvGetWeekDayName, 84
- smvIgnoreChar, 84, 85
- smvPutAmPm, 85
- smvPutMonthName, 85
- smvPutNumber1, 86
- smvPutNumber2Blank, 86
- smvPutNumber2Zero, 87
- smvPutNumber3Zero, 87, 88
- smvPutNumber4Zero, 88
- smvPutTzName, 88
- smvPutTzOffset, 89
- smvPutWeekDayName, 89
- lfc1::datetime::CDateTimeManip, 89
 - CDateTimeManip, 90
 - smvCopyFmtErr, 90
 - smvGetFormat, 90
- lfc1::datetime::CDateTimePut, 91
 - CDateTimePut, 94
 - mvDoPut, 94
 - mvPut, 94
 - smvCalcWeekBasedDetails, 95
 - smvCalcWeekNo, 95
 - smvExpandFormat, 95
 - smvGetAmPm, 96
 - smvGetMonthName, 96
 - smvGetNumber, 96, 97
 - smvGetTzName, 97
 - smvGetTzOffset, 98
 - smvGetWeekDayName, 98
 - smvIgnoreChar, 98, 99
 - smvPutAmPm, 99
 - smvPutMonthName, 99
 - smvPutNumber1, 99, 100
 - smvPutNumber2Blank, 100
 - smvPutNumber2Zero, 101
 - smvPutNumber3Zero, 101
 - smvPutNumber4Zero, 102
 - smvPutTzName, 102
 - smvPutTzOffset, 102
 - smvPutWeekDayName, 103
- lfc1::datetime::CFacetBase, 109
 - smvGetNumber, 110
 - smvIgnoreChar, 110
 - smvPutNumber1, 111
 - smvPutNumber2Blank, 111
 - smvPutNumber2Zero, 111
 - smvPutNumber3Zero, 111
 - smvPutNumber4Zero, 112
- lfc1::datetime::CLangInfo, 120
 - smvGetAMString, 121
 - smvGetDateFormat, 121
 - smvGetDateTimeFormat, 121
 - smvGetLongDayName, 121
 - smvGetLongMonthName, 122
 - smvGetPMString, 122
 - smvGetShortDayName, 122
 - smvGetShortMonthName, 122
 - smvGetTime12Format, 123
 - smvGetTime24Format, 123
- lfc1::datetime::CTime, 128
 - CTime, 131
 - mvGetTime, 131
 - mvInput, 131
 - mvOutput, 132
 - mvSetTime, 132
 - operator int, 132
 - operator++, 132, 133
 - operator+=", 133
 - operator--, 133
 - operator-=, 133
 - operator=, 134
 - smvGetIndex, 134
 - smvValidateTime, 135
 - swap, 135
- lfc1::datetime::CTime::STime, 158
- lfc1::datetime::CTimeBase, 135
 - smvExpandFormat, 137
 - smvGetAmPm, 137
 - smvGetNumber, 137
 - smvIgnoreChar, 138
 - smvPutAmPm, 138
 - smvPutNumber1, 138
 - smvPutNumber2Blank, 139
 - smvPutNumber2Zero, 139
 - smvPutNumber3Zero, 139
 - smvPutNumber4Zero, 139
- lfc1::datetime::CTimeDuration, 140
 - CTimeDuration, 141
 - mvGetDuration, 141
 - mvGetUnit, 141
 - operator=, 141, 142
 - swap, 142
- lfc1::datetime::CTimeGet, 142
 - CTimeGet, 144
 - mvDoGet, 144
 - mvGet, 144
 - smvExpandFormat, 145
 - smvGetAmPm, 145
 - smvGetNumber, 146
 - smvIgnoreChar, 146
 - smvPutAmPm, 146
 - smvPutNumber1, 147
 - smvPutNumber2Blank, 147
 - smvPutNumber2Zero, 147
 - smvPutNumber3Zero, 147
 - smvPutNumber4Zero, 148
- lfc1::datetime::CTimeManip, 148
 - CTimeManip, 149

- smvCopyFmtErr, 149
- smvGetFormat, 149
- lfc1::datetime::CTimePut, 150
 - CTimePut, 151
 - mvDoPut, 151
 - mvPut, 152
 - smvExpandFormat, 152
 - smvGetAmPm, 152
 - smvGetNumber, 153
 - smvIgnoreChar, 153
 - smvPutAmPm, 154
 - smvPutNumber1, 154
 - smvPutNumber2Blank, 154
 - smvPutNumber2Zero, 154
 - smvPutNumber3Zero, 155
 - smvPutNumber4Zero, 155
- lfc1::filesystem::CDirectory, 104
 - begin, 105
 - CDirectory, 105
 - end, 106
 - mvGetNumEntries, 106
 - mvGetPath, 106
 - mvRefresh, 106
 - operator=, 106, 107
 - swap, 107
- lfc1::filesystem::CRecDirectory, 125
 - begin, 126
 - CRecDirectory, 126
 - end, 127
 - mvGetLevelCount, 127
 - mvGetNumEntries, 127
 - mvGetPath, 127
 - mvRefresh, 127
 - operator=, 127, 128
 - swap, 128
- lfc1::misc::CDemangler, 103
 - CDemangler, 104
 - operator const char *, 104
- lfc1::numeric::CInteger
 - ~CInteger, 114
 - CInteger, 114
 - operator IT, 115
 - operator <=, 117
 - operator >=, 119
 - operator *=, 115
 - operator ^=, 119
 - operator ++, 115, 116
 - operator +=, 116
 - operator --, 116
 - operator -=, 117
 - operator /=, 117
 - operator =, 118
 - operator &=, 115
 - swap, 120
- lfc1::numeric::CInteger < IT >, 112
- lfc1::system::CErrorCategory, 107
 - message, 108
 - name, 108
 - smrGetErrorCategory, 108
- lfc1::udtsup::CManipBase
 - ~CManipBase, 124
 - CManipBase, 124
 - gvDoManip, 124
- lfc1::udtsup::CManipBase < T1 >, 123
- MAX_DATETIMENO
 - lfc1::datetime::CDateTime, 64
- MIN_DATE_YEAR
 - lfc1::datetime::CDate, 32
- MIN_DATETIMENO
 - lfc1::datetime::CDateTime, 64
- message
 - lfc1::system::CErrorCategory, 108
- Miscellaneous class library, 15
- mvDoGet
 - lfc1::datetime::CDateGet, 43
 - lfc1::datetime::CDateTimeGet, 80
 - lfc1::datetime::CTimeGet, 144
- mvDoPut
 - lfc1::datetime::CDatePut, 52
 - lfc1::datetime::CDateTimePut, 94
 - lfc1::datetime::CTimePut, 151
- mvGet
 - lfc1::datetime::CDateGet, 43
 - lfc1::datetime::CDateTimeGet, 80
 - lfc1::datetime::CTimeGet, 144
- mvGetDate
 - lfc1::datetime::CDate, 28
- mvGetDateTime
 - lfc1::datetime::CDateTime, 60
- mvGetDuration
 - lfc1::datetime::CDateDuration, 40
 - lfc1::datetime::CDateTimeDuration, 76
 - lfc1::datetime::CTimeDuration, 141
- mvGetLevelCount
 - lfc1::filesystem::CRecDirectory, 127
- mvGetNumEntries
 - lfc1::filesystem::CDirectory, 106
 - lfc1::filesystem::CRecDirectory, 127
- mvGetPath
 - lfc1::filesystem::CDirectory, 106
 - lfc1::filesystem::CRecDirectory, 127
- mvGetTime
 - lfc1::datetime::CTime, 131
- mvGetUnit
 - lfc1::datetime::CDateDuration, 40
 - lfc1::datetime::CDateTimeDuration, 76
 - lfc1::datetime::CTimeDuration, 141
- mvInput
 - lfc1::datetime::CDate, 28
 - lfc1::datetime::CDateTime, 60
 - lfc1::datetime::CTime, 131
- mvOutput
 - lfc1::datetime::CDate, 28
 - lfc1::datetime::CDateTime, 60
 - lfc1::datetime::CTime, 132
- mvPut

- lfc1::datetime::CDatePut, 52
- lfc1::datetime::CDateTimePut, 94
- lfc1::datetime::CTimePut, 152
- mvRefresh
 - lfc1::filesystem::CDirectory, 106
 - lfc1::filesystem::CRecDirectory, 127
- mvSetDate
 - lfc1::datetime::CDate, 28
- mvSetDateTime
 - lfc1::datetime::CDateTime, 60
- mvSetTime
 - lfc1::datetime::CTime, 132
- name
 - lfc1::system::CErrorCategory, 108
- Numeric class library, 16
 - swap, 17
- operator const char *
 - lfc1::misc::CDemangler, 104
- operator IT
 - lfc1::numeric::CInteger, 115
- operator int
 - lfc1::datetime::CDate, 29
 - lfc1::datetime::CTime, 132
- operator long long
 - lfc1::datetime::CDateTime, 61
- operator<<
 - Date and time class library, 20, 21
 - User-defined type support class library, 14
- operator<<=
 - lfc1::numeric::CInteger, 117
- operator>>
 - Date and time class library, 21, 22
 - User-defined type support class library, 14
- operator>>=
 - lfc1::numeric::CInteger, 119
- operator*=
 - lfc1::numeric::CInteger, 115
- operator^=
 - lfc1::numeric::CInteger, 119
- operator++
 - lfc1::datetime::CDate, 29
 - lfc1::datetime::CDateTime, 61
 - lfc1::datetime::CTime, 132, 133
 - lfc1::numeric::CInteger, 115, 116
- operator+=
 - lfc1::datetime::CDate, 29
 - lfc1::datetime::CDateTime, 61
 - lfc1::datetime::CTime, 133
 - lfc1::numeric::CInteger, 116
- operator--
 - lfc1::datetime::CDate, 29, 30
 - lfc1::datetime::CDateTime, 62
 - lfc1::datetime::CTime, 133
 - lfc1::numeric::CInteger, 116
- operator-=
 - lfc1::datetime::CDate, 30
 - lfc1::datetime::CDateTime, 62
- lfc1::datetime::CTime, 133
- lfc1::numeric::CInteger, 117
- operator/=
 - lfc1::numeric::CInteger, 117
- operator=
 - lfc1::datetime::CDate, 30, 31
 - lfc1::datetime::CDateDuration, 40
 - lfc1::datetime::CDateTime, 62, 63
 - lfc1::datetime::CDateTimeDuration, 76
 - lfc1::datetime::CTime, 134
 - lfc1::datetime::CTimeDuration, 141, 142
 - lfc1::filesystem::CDirectory, 106, 107
 - lfc1::filesystem::CRecDirectory, 127, 128
 - lfc1::numeric::CInteger, 118
- operator&=
 - lfc1::numeric::CInteger, 115
- setdateformat
 - Date and time class library, 22
- setdatetimeformat
 - Date and time class library, 22
- settimeformat
 - Date and time class library, 22
- smrGetErrorCategory
 - lfc1::system::CErrorCategory, 108
- smvCalcWeekBasedDetails
 - lfc1::datetime::CDateBase, 34
 - lfc1::datetime::CDateGet, 44
 - lfc1::datetime::CDatePut, 53
 - lfc1::datetime::CDateTimeBase, 66
 - lfc1::datetime::CDateTimeGet, 81
 - lfc1::datetime::CDateTimePut, 95
- smvCalcWeekNo
 - lfc1::datetime::CDateBase, 34
 - lfc1::datetime::CDateGet, 44
 - lfc1::datetime::CDatePut, 53
 - lfc1::datetime::CDateTimeBase, 67
 - lfc1::datetime::CDateTimeGet, 81
 - lfc1::datetime::CDateTimePut, 95
- smvCopyFmtErr
 - lfc1::datetime::CDateManip, 49
 - lfc1::datetime::CDateTimeManip, 90
 - lfc1::datetime::CTimeManip, 149
- smvExpandFormat
 - lfc1::datetime::CDateBase, 35
 - lfc1::datetime::CDateGet, 44
 - lfc1::datetime::CDatePut, 53
 - lfc1::datetime::CDateTimeBase, 67
 - lfc1::datetime::CDateTimeGet, 82
 - lfc1::datetime::CDateTimePut, 95
 - lfc1::datetime::CTimeBase, 137
 - lfc1::datetime::CTimeGet, 145
 - lfc1::datetime::CTimePut, 152
- smvGetAMString
 - lfc1::datetime::CLangInfo, 121
- smvGetAmPm
 - lfc1::datetime::CDateTimeBase, 67
 - lfc1::datetime::CDateTimeGet, 82
 - lfc1::datetime::CDateTimePut, 96

- lfc1::datetime::CTimeBase, 137
- lfc1::datetime::CTimeGet, 145
- lfc1::datetime::CTimePut, 152
- smvGetDateFormat
 - lfc1::datetime::CLangInfo, 121
- smvGetDateTimeFormat
 - lfc1::datetime::CLangInfo, 121
- smvGetFormat
 - lfc1::datetime::CDateManip, 49
 - lfc1::datetime::CDateTimeManip, 90
 - lfc1::datetime::CTimeManip, 149
- smvGetIndex
 - lfc1::datetime::CDate, 31
 - lfc1::datetime::CDateTime, 63
 - lfc1::datetime::CTime, 134
- smvGetLongDayName
 - lfc1::datetime::CLangInfo, 121
- smvGetLongMonthName
 - lfc1::datetime::CLangInfo, 122
- smvGetMaxDay
 - lfc1::datetime::CDate, 31
- smvGetMonthName
 - lfc1::datetime::CDateBase, 35
 - lfc1::datetime::CDateGet, 45
 - lfc1::datetime::CDatePut, 53
 - lfc1::datetime::CDateTimeBase, 67
 - lfc1::datetime::CDateTimeGet, 82
 - lfc1::datetime::CDateTimePut, 96
- smvGetNumber
 - lfc1::datetime::CDateBase, 35
 - lfc1::datetime::CDateGet, 45
 - lfc1::datetime::CDatePut, 54
 - lfc1::datetime::CDateTimeBase, 68
 - lfc1::datetime::CDateTimeGet, 83
 - lfc1::datetime::CDateTimePut, 96, 97
 - lfc1::datetime::CFacetBase, 110
 - lfc1::datetime::CTimeBase, 137
 - lfc1::datetime::CTimeGet, 146
 - lfc1::datetime::CTimePut, 153
- smvGetPMString
 - lfc1::datetime::CLangInfo, 122
- smvGetShortDayName
 - lfc1::datetime::CLangInfo, 122
- smvGetShortMonthName
 - lfc1::datetime::CLangInfo, 122
- smvGetTime12Format
 - lfc1::datetime::CLangInfo, 123
- smvGetTime24Format
 - lfc1::datetime::CLangInfo, 123
- smvGetTzName
 - lfc1::datetime::CDateTimeBase, 69
 - lfc1::datetime::CDateTimeGet, 83
 - lfc1::datetime::CDateTimePut, 97
- smvGetTzOffset
 - lfc1::datetime::CDateTimeBase, 69
 - lfc1::datetime::CDateTimeGet, 84
 - lfc1::datetime::CDateTimePut, 98
- smvGetWeekDayName
 - lfc1::datetime::CDateBase, 36
 - lfc1::datetime::CDateGet, 45
 - lfc1::datetime::CDatePut, 54
 - lfc1::datetime::CDateTimeBase, 69
 - lfc1::datetime::CDateTimeGet, 84
 - lfc1::datetime::CDateTimePut, 98
- smvIgnoreChar
 - lfc1::datetime::CDateBase, 36
 - lfc1::datetime::CDateGet, 46
 - lfc1::datetime::CDatePut, 54
 - lfc1::datetime::CDateTimeBase, 70
 - lfc1::datetime::CDateTimeGet, 84, 85
 - lfc1::datetime::CDateTimePut, 98, 99
 - lfc1::datetime::CFacetBase, 110
 - lfc1::datetime::CTimeBase, 138
 - lfc1::datetime::CTimeGet, 146
 - lfc1::datetime::CTimePut, 153
- smvIsLeapYear
 - lfc1::datetime::CDate, 31
- smvPutAmPm
 - lfc1::datetime::CDateTimeBase, 70
 - lfc1::datetime::CDateTimeGet, 85
 - lfc1::datetime::CDateTimePut, 99
 - lfc1::datetime::CTimeBase, 138
 - lfc1::datetime::CTimeGet, 146
 - lfc1::datetime::CTimePut, 154
- smvPutMonthName
 - lfc1::datetime::CDateBase, 36
 - lfc1::datetime::CDateGet, 46
 - lfc1::datetime::CDatePut, 55
 - lfc1::datetime::CDateTimeBase, 70
 - lfc1::datetime::CDateTimeGet, 85
 - lfc1::datetime::CDateTimePut, 99
- smvPutNumber1
 - lfc1::datetime::CDateBase, 37
 - lfc1::datetime::CDateGet, 46
 - lfc1::datetime::CDatePut, 55
 - lfc1::datetime::CDateTimeBase, 71
 - lfc1::datetime::CDateTimeGet, 86
 - lfc1::datetime::CDateTimePut, 99, 100
 - lfc1::datetime::CFacetBase, 111
 - lfc1::datetime::CTimeBase, 138
 - lfc1::datetime::CTimeGet, 147
 - lfc1::datetime::CTimePut, 154
- smvPutNumber2Blank
 - lfc1::datetime::CDateBase, 37
 - lfc1::datetime::CDateGet, 47
 - lfc1::datetime::CDatePut, 55
 - lfc1::datetime::CDateTimeBase, 71, 72
 - lfc1::datetime::CDateTimeGet, 86
 - lfc1::datetime::CDateTimePut, 100
 - lfc1::datetime::CFacetBase, 111
 - lfc1::datetime::CTimeBase, 139
 - lfc1::datetime::CTimeGet, 147
 - lfc1::datetime::CTimePut, 154
- smvPutNumber2Zero
 - lfc1::datetime::CDateBase, 37
 - lfc1::datetime::CDateGet, 47

- lfc1::datetime::CDatePut, 56
- lfc1::datetime::CDateTimeBase, 72
- lfc1::datetime::CDateTimeGet, 87
- lfc1::datetime::CDateTimePut, 101
- lfc1::datetime::CFacetBase, 111
- lfc1::datetime::CTimeBase, 139
- lfc1::datetime::CTimeGet, 147
- lfc1::datetime::CTimePut, 154
- smvPutNumber3Zero
 - lfc1::datetime::CDateBase, 37
 - lfc1::datetime::CDateGet, 47
 - lfc1::datetime::CDatePut, 56
 - lfc1::datetime::CDateTimeBase, 72, 73
 - lfc1::datetime::CDateTimeGet, 87, 88
 - lfc1::datetime::CDateTimePut, 101
 - lfc1::datetime::CFacetBase, 111
 - lfc1::datetime::CTimeBase, 139
 - lfc1::datetime::CTimeGet, 147
 - lfc1::datetime::CTimePut, 155
- smvPutNumber4Zero
 - lfc1::datetime::CDateBase, 38
 - lfc1::datetime::CDateGet, 47
 - lfc1::datetime::CDatePut, 56
 - lfc1::datetime::CDateTimeBase, 73
 - lfc1::datetime::CDateTimeGet, 88
 - lfc1::datetime::CDateTimePut, 102
 - lfc1::datetime::CFacetBase, 112
 - lfc1::datetime::CTimeBase, 139
 - lfc1::datetime::CTimeGet, 148
 - lfc1::datetime::CTimePut, 155
- smvPutTzName
 - lfc1::datetime::CDateTimeBase, 73
 - lfc1::datetime::CDateTimeGet, 88
 - lfc1::datetime::CDateTimePut, 102
- smvPutTzOffset
 - lfc1::datetime::CDateTimeBase, 74
 - lfc1::datetime::CDateTimeGet, 89
 - lfc1::datetime::CDateTimePut, 102
- smvPutWeekDayName
 - lfc1::datetime::CDateBase, 38
 - lfc1::datetime::CDateGet, 48
 - lfc1::datetime::CDatePut, 57
 - lfc1::datetime::CDateTimeBase, 74
 - lfc1::datetime::CDateTimeGet, 89
 - lfc1::datetime::CDateTimePut, 103
- smvValidateDate
 - lfc1::datetime::CDate, 31
- smvValidateDst
 - lfc1::datetime::CDateTime, 63
- smvValidateTime
 - lfc1::datetime::CTime, 135
- std::numeric_limits< lfc1::numeric::CInteger< IT > >, 155
- swap
 - Date and time class library, 23
 - Filesystem class library, 18
 - lfc1::datetime::CDate, 32
 - lfc1::datetime::CDateDuration, 40
 - lfc1::datetime::CDateTime, 63
 - lfc1::datetime::CDateTimeDuration, 76
 - lfc1::datetime::CTime, 135
 - lfc1::datetime::CTimeDuration, 142
 - lfc1::filesystem::CDirectory, 107
 - lfc1::filesystem::CRecDirectory, 128
 - lfc1::numeric::CInteger, 120
 - Numeric class library, 17
 - System class library, 9
 - grCategory, 10
 - User-defined type support class library, 11
 - gvDoManip, 11
 - gvInput, 12
 - gvInputGuarded, 12
 - gvOutput, 13
 - gvOutputGuarded, 13
 - operator<<, 14
 - operator>>, 14