

Quantlab Instrument Library

<Fixed Income Swaps>

Last update: 2018-11-25, ver. 0.99.

© 2012-2019 Algorithmica Research AB. All rights reserved.

Table of Contents

1	Introduction	5
2	Def Constructors	5
2.1	Extended Def Constructors	5
2.1.1	Extended ql_swap_float_def	5
2.1.2	Extended ql_swap_fix_def	8
2.1.3	Extended ql_swap_ois_def	9
2.1.4	Extended ql_swap_fix_zero_def	10
2.2	Specific Def Constructors	11
2.2.1	Plain vanilla ql_swap_float_def	11
2.2.2	Plain vanilla ql_swap_fix_def	12
2.2.3	Plain vanilla ql_swap_ois_def	12
2.2.4	Plain vanilla ql_swap_fix_zero_def	13
3	Leg Constructors	15
3.1	Plain vanilla float leg constructor	15
3.2	Plain vanilla float leg mtm constructor	16
3.3	Plain vanilla fixed leg constructor	18
3.4	Plain vanilla ois leg constructor	18
3.5	Plain vanilla fix_zero leg constructor	20
4	Instrument Constructors	20
4.1	Basic Instrument Constructors	20
4.2	Extended Instrument Constructors	21
4.2.1	Extended fix/float	21
4.2.2	Extended fix/fix	24
4.2.3	Extended float/float	24

4.2.4	Extended fix/ois	24
4.2.5	Extended fix_zero/float	27
4.3	Specific Instrument Constructors	30
4.3.1	Plain vanilla fix/float	30
4.3.2	Plain vanilla fix/float with external coupon flows.....	32
4.3.3	Plain vanilla fix/float with sinkfunds and fees	33
4.3.4	Plain vanilla fix/fix	36
4.3.5	Plain vanilla float/float.....	36
4.3.6	Plain vanilla float/float mtm	36
4.3.7	Plain vanilla fix/ois	38
4.3.8	Plain vanilla fix_zero/float	39
5	Data types	41
5.1	Interest calculation rule	41
5.2	The implied forward rate	41
5.3	The fixed rate	43
5.4	Reset frequency greater than coupon frequency.....	44
5.5	Stub periods	45
5.6	Fixing rates	46
5.7	Tenor surface	48
5.8	Exchange of notional.....	48
5.9	Sinking funds	48
5.10	Zero rate compounding	50
6	Member functions and other functions.....	52
6.1	ql_fixed_income_swap members.....	52
6.2	ql_swap_leg members	54

6.3	ql_float_leg members	54
6.4	ql_fix_leg members.....	56
6.5	ql_ois_leg members.....	58
6.6	ql_fix_zero_leg members.....	59
6.7	other functions.....	60

1 Introduction

This document is a short explanation of functions relating to fixed income swaps in the Quantlab instrument library.

2 Def Constructors

A Def defines static data for a swap leg.

2.1 Extended Def Constructors

The extended def constructors includes all available arguments, hence they are normally only used if a more specific def constructor does not exist.

2.1.1 Extended ql_swap_float_def

The extended ql_swap_float_def contains all arguments relevant for a floating leg definition.

Argument	Data type	Description
name	string	Name.
currency	string	Currency.
settle_code	date_code	A date_code that determines the settledate as an offset to trade date.
coupon_freq	coupon_freq	Coupon frequency.
daycount	day_count_method	Daycount method.
roll_method	flt_roll_method	Date roll method. Default = FLR STRAIGHT.
bus_day	bd_convention	Businessday convention.
calendar	calendar	The calendar used to determine business days.
eom	logical	End-of-month.
reset_freq (nullable)	reset_freq	The reset frequency for the floating rate. Must always be more frequent or the same frequency as the coupon frequency. Default = coupon_freq.
reset_days	number	Number of business days before a coupon date the rate will be reset.
index_tenor (nullable)	tenor_code	The tenor for the floating rate index. If this is not defined it will default to the tenor implied by the coupon frequency or the reset frequency (if this is more frequent)

index_rate_type (nullable)	flt_index_rate_type	Type of rate for the index. Default = IRT_SIMPLE .
index_cpn_freq (nullable)	coupon_freq	Coupon frequency of the index rate. Only required and applicable when index_rate_type is of type swap rate i.e. index_rate_type = IRT_SWAP .
index_daycount (nullable)	day_count_method	Daycount method of the index rate. Default = daycount.
index_bus_day (nullable)	bd_convention	Businessday convention of the index rate. Default = bus_day.
index_calendar (nullable)	calendar	Calendar for the index. Default = calendar.
index_eom (nullable)	logical	End-of-month setting for the index. Default = eom.
quote_style	swap_quote_style	<work-in-progress>
pv_round_code (nullable)	round_code	Rounding of present value. Default = no rounding.
fwd_rate_round_code (nullable)	round_code	Rounding of calculated implied forward rates. Fixing rates will not be rounded. Default = no rounding.
fwd_rate_style (nullable)	flt_fwd_style	Determines start and end date for the implied forward rate calculation. See section <The implied forward rate>. Default = FF_IDX .
fwd_stub_rate_style (nullable)	flt_stub_fwd_style	Determines start and end date for the implied forward rate calculation in a stub (start or end) period. See section <Stub periods>. Default = FSF_IDX .
start_stub_sprd_decomp (nullable)	logical	If true, the floating rate spread will be decompounded for a start stub period, otherwise the spread is used as input. See section <Stub periods>. Default = false.
end_stub_sprd_decomp (nullable)	logical	If true, the floating rate spread will be decompounded for an end stub period, otherwise the spread is used as input. See section <Stub periods>. Default = false.
interest_rule (nullable)	interest_rule	Rule for calculating interest. See section <Interest calculation rule>. Default = IR_MAT_NOADJ .
fixing_stub_style (nullable)	fixing_stub_style	Determines how fixing rates are handled for stub periods. See section <Fixing rates>. Default = FSC_IDX .
pmt_delay (nullable)	number	Number of business days after a coupon date the payment will be made. This does not change the number of interest days in a period. Default = 0.
mat_pmt_delay	number	Number of business days after

(nullable)		maturity the final payment will be made. This does not change the number of interest days in the last period. Default = 0.
comp_avg (nullable)	flt_comp_avg_type	Type of method used when reset_freq > coupon_freq. See section <Reset frequency greater than coupon frequency>. Required if reset frequency is more frequent than coupon frequency.
spread_comp_method (nullable)	flt_sprd_comp_method	Type of compounding method. See section <Reset frequency greater than coupon frequency>. Required if reset frequency is more frequent than coupon frequency and comp_avg = CAT_COMP. It is also required when compound_to_mat is true.
avg_method (nullable)	flt_avg_method	Type of averaging method. See section <Reset frequency greater than coupon frequency>. Required if reset frequency is more frequent than coupon frequency and comp_avg = CAT_AVG.
reset_style	flt_reset_style	Type of forward rate. See section <The implied forward rate>. Default = FRS NORM.
notional_exchange	notional_exchg_style	see section <Exchange of notional>
average_notional (nullable)	logical	Relevant only when there is an amortization between coupon payment dates. If this is true the notional used for determining the interest payment will be the weighted average notional. See section <sinking funds>. Default = false.
compound_to_mat (nullable)	logical	If true all interest cash flows are compounded to maturity. The convention used for compounding is determined by the spread_comp_method argument. Default = false.
sub_factor (nullable)	number	The rate will be calculated as sub_factor (C) minus reset rate (r) i.e. C-r. Default = none.
multi_factor (nullable)	number	The rate will be calculated as multi_factor (M) times the reset rate (r) i.e. M*r. Default = none.
power_factor (nullable)	number	The rate will be calculated as the reset rate (r) raised to the power = power_factor (P) i.e. r^P. Default = none.

2.1.2 Extended ql_swap_fix_def

The extended ql_swap_fix_def contains all arguments relevant for a fixed leg definition.

Argument	Data type	Description
name	string	Name.
currency	string	Currency.
settle_code	date_code	A date_code that determines the settledate as an offset to trade date.
coupon_freq	coupon_freq	Coupon frequency.
daycount	day_count_method	Daycount method.
roll_method	fix_roll_method	Date roll method.
bus_day	bd_convention	Businessday convention.
calendar	calendar	The calendar used to determine business days.
eom	logical	End-of-month.
constant_coupon (nullable)	logical	If true the coupon will be calculated only from the frequency i.e. the cash flow is independent of the number of days in the coupon period. Default = false.
compound_freq (nullable)	compound_freq	Determines the compounding frequency of the swap coupon rate. Default = coupon_freq.
quote_style	swap_quote_style	<work-in-progress>
pv_round_code (nullable)	round_code	Rounding of present value. Default = no rounding.
start_stub_decomp (nullable)	logical	Determines if the swap rate should be decomposed before calculating the first interest cash flow. Relevant only if there is a start stub period. Default = false.
end_stub_decomp (nullable)	logical	Determines if the swap rate should be decomposed before calculating the last interest cash flow. Relevant only if there is an end stub period. Default = false.
interest_rule (nullable)	interest_rule	Rule for calculating interest. See section <Interest calculation rule>. Default = IR MAT NOADJ.
pmt_delay (nullable)	number	Number of business days after a coupon date the payment will be made. This does not change number of interest days in a period. Default = 0.
mat_pmt_delay (nullable)	number	Number of business days after maturity the final payment will be made. This does not change number of interest days in the last period. Default = 0.
notional_exchange	notional_exchg_style	see section <Exchange of notional>
average_notional (nullable)	logical	Relevant only when there is an amortization between coupon payment dates. If this is true the notional used for determining the interest payment will be the weighted average

		notional. See section <sinking funds>. Default = false.
compound_to_mat (nullable)	logical	If this is true all interest cash flows will be compounded to maturity. Default = false.

2.1.3 Extended ql_swap_ois_def

The extended ql_swap_ois_def contains all arguments relevant for an ois leg definition.

Argument	Data type	Description
name	string	Name.
currency	string	Currency.
settle_code	date_code	A date_code that determines the settledate as an offset to trade date.
coupon_freq	coupon_freq	Coupon frequency.
daycount	day_count_method	Daycount method.
roll_method	flt_roll_method	Date roll method.
bus_day	bd_convention	Businessday convention.
calendar	calendar	The calendar used to determine business days.
eom	logical	End-of-month.
index_daycount (nullable)	day_count_method	Daycount method of the index rate. Default = daycount.
index_calendar (nullable)	calendar	Calendar for the index. Default = calendar.
quote_style	swap_quote_style	<work-in-progress>
pv_round_code (nullable)	round_code	Rounding of present value. Default = no rounding.
comp_factor_round_code (nullable)	round_code	Rounding of compounding factors. Default = no rounding.
imp_rate_pcmt_round_code (nullable)	round_code	Rounding of the calculated inferred period rate (both fixing rates and implied forward rates is typically used to calculate the inferred rate). The rounding decimals is defined for the rate in percentage form. Default = no rounding.
pmt_delay (nullable)	number	Number of business days after a coupon date the payment will be made. This does not change number of interest days in a period.
mat_pmt_delay (nullable)	number	Number of business days after maturity the final payment will be made. This does not change number of interest days in the last period.
comp_avg (nullable)	flt_comp_avg_type	Type of method used when reset_freq > coupon_freq. See section <Reset frequency greater

		than coupon frequency>.
spread_comp_method (nullable)	flt_sprd_comp_method	Type of compounding method. See section <Reset frequency greater than coupon frequency>.
avg_method (nullable)	flt_avg_method	Type of averaging method. See section <Reset frequency greater than coupon frequency>.
reset_style	flt_reset_style	Type of forward rate. See section <The implied forward rate>
notional_exchange	notional_exchg_style	see section <Exchange of notional>
average_notional (nullable)	logical	Relevant only when there is an amortization between coupon payment dates. If this is true the notional used for determining the interest payment will be the weighted average notional. See section <sinking funds>.
is_tn	logical	If true the ois swap is a tom-next rate ois otherwise it is an overnight rate ois.
publication_lag (nullable)	integer	The number of days after the start date of the overnight period the fixing rate will be published.
Sub_factor (nullable)	number	The rate will be calculated as sub_factor (C) minus reset rate (r) i.e. C-r.
multi_factor (nullable)	number	The rate will be calculated as multi_factor (M) times the reset rate (r) i.e. M*r.
power_factor (nullable)	number	The rate will be calculated as the reset rate (r) raised to the power = power_factor (P) i.e. r^P .

2.1.4 Extended ql_swap_fix_zero_def

The extended ql_swap_fix_zero_def contains all arguments relevant for a fixzero leg definition.

Argument	Data type	Description
name	string	Name.
currency	string	Currency.
settle_code	date_code	A date_code that determines the settledate as an offset to trade date.
daycount	day_count_method	Daycount method.
bus_day	bd_convention	Businessday convention.
calendar	calendar	The calendar used to determine business days.
comp_style	zero_rate_comp_style	Determines how the zero coupon rate will be interpreted. It can either be

		a rate with a certain compounding frequency, a simple rate or a discount rate. See section <Zero rate compounding>
notional_net (nullable)	logical	If this is true the notional as input will be reduced by the interest payment. This means that the total payment (including interest) at maturity will be equal to the notional. This argument is not applicable for a reverse zero swap (see below). (Default = false)
reverse (nullable)	logical	If this is true the interest will be paid on the start date. This type of swap is commonly called a reverse zero coupon swap. The notional_net argument is always false for a reverse zero swap. (Default = false)
quote_style	swap_quote_style	<work-in-progress>
pv_round_code (nullable)	round_code	Rounding of present value. Default = no rounding.
pmt_delay (nullable)	number	Number of business days after a coupon date the payment will be made. This does not change number of interest days in a period. (Default = 0)
mat_pmt_delay (nullable)	number	Number of business days after maturity the final payment will be made. This does not change number of interest days in the last period. (Default = 0)
notional_exchange	notional_exchg_style	see section <Exchange of notional>
average_notional (nullable)	logical	Relevant only when there is an amortization between coupon payment dates. If this is true the notional used for determining the interest payment will be the weighted average notional. See section <sinking funds>. (Default = false)

2.2 Specific Def Constructors

A specific def contains a subset of the parameters in the extended def i.e. some parameters are set to appropriate default values depending on the type of def.

2.2.1 Plain vanilla ql_swap_float_def

The “plain vanilla” ql_swap_float_def contains the most common arguments relevant for a floating leg definition.

Argument	Data type	Description
name	string	Name.
currency	string	Currency.
settle_code	date_code	A date_code that determines the settledate as an offset to trade date.
coupon_freq	coupon_freq	Coupon frequency.
daycount	day count method	Daycount method.
bus_day	bd_convention	Businessday convention.
calendar	calendar	The calendar used to determine business days.
eom	logical	End-of-month.
reset_days	number	Number of business days before a coupon date the rate will be reset.
reset_style	flt_reset_style	Type of forward rate. See section <The implied forward rate>
notional_exchange	notional_exchg_style	see section <Exchange of notional>

2.2.2 Plain vanilla ql_swap_fix_def

The “plain vanilla” ql_swap_fix_def contains the most common arguments relevant for a fixed leg definition.

Argument	Data type	Description
name	string	Name.
currency	string	Currency.
Settle_code	date_code	A date_code that determines the settledate as an offset to trade date.
Coupon_freq	coupon_freq	Coupon frequency.
Daycount	day count method	Daycount method.
Bus_day	bd convention	Businessday convention.
Calendar	calendar	The calendar used to determine business days.
Eom	logical	End-of-month.
Notional_exchange	notional_exchg_style	see section <Exchange of notional>

2.2.3 Plain vanilla ql_swap_ois_def

The “plain vanilla” ql_swap_ois_def contains the most common arguments relevant for an ois leg definition.

Argument	Data type	Description
name	string	Name.
currency	string	Currency.
settle_code	date_code	A date_code that determines the settledate as an offset to trade date.

coupon_freq	coupon_freq	Coupon frequency.
daycount	day_count_method	Daycount method.
bus_day	bd_convention	Businessday convention.
calendar	calendar	The calendar used to determine business days.
eom	logical	End-of-month.
comp_factor_round_code (nullable)	round_code	Rounding of compounding factors. Default = no rounding.
imp_rate_pcmt_round_code (nullable)	round_code	Rounding of the calculated inferred period rate (both fixing rates and implied forward rates is typically used to calculate the inferred rate). The rounding decimals is defined for the rate in percentage form. Default = no rounding.
pmt_delay (nullable)	number	Number of business days after maturity the final payment will be made. This does not change number of interest days in the last period. This setting will also be used for coupon payment dates.
comp_avg	flt_comp_avg_type	Compounding or averaging. See section <Reset frequency greater than coupon frequency>.
spread_comp_method (nullable)	flt_sprd_comp_method	Type of compounding method. See section <Reset frequency greater than coupon frequency>.
avg_method (nullable)	flt_avg_method	Type of averaging method. See section <Reset frequency greater than coupon frequency>.
notional_exchange	notional_exchg_style	see section <Exchange of notional>
is_tn	logical	If true the ois swap is a tom-next rate ois otherwise it is an overnight rate ois.
publication_lag	integer	The number of days after the start date of the overnight period the fixing rate will be published.

2.2.4 Plain vanilla ql_swap_fix_zero_def

The “plain vanilla” ql_swap_fix_zero_def contains the most common arguments relevant for a fixzero leg definition.

Argument	Data type	Description
name	string	Name.
currency	string	Currency.
settle_code	date_code	A date_code that determines the settledate as an offset to trade date.
daycount	day_count_method	Daycount method.

bus_day	bd_convention	Businessday convention.
calendar	calendar	The calendar used to determine business days.
comp_style	zero_rate_comp_style	Determines how the coupon rate will be interpreted. It can either be a rate with a certain compounding frequency, a simple rate or a discount rate. See section <Zero rate compounding>
notional_net (nullable)	logical	If this is true the notional as input will be reduced by the interest payment. This means that the total payment (including interest) at maturity will be equal to the notional. This argument is not applicable for a reverse zero swap (see below). (Default = false)
reverse (nullable)	logical	If this is true the interest will be paid on the start date. This type of swap is commonly called a reverse zero coupon swap. The notional_net argument is always false for a reverse zero swap. (Default = false)
notional_exchange	notional_exchg_style	see section <Exchange of notional>

3 Leg Constructors

A complete swap leg can be created outside of a swap. A swap leg is not an instrument until it is a part of a swap. When independently created legs are used to create a swap it is important to make sure the legs are compatible with each other. Therefore, this is normally not the way to create a swap but can still be useful in some circumstances. The standard way to create a swap is by using the constructors in the next section (section 4).

3.1 Plain vanilla float leg constructor

The “plain vanilla” floatleg constructor contains the most common arguments relevant for a float leg.

Argument	Data type	Description
ql_swapflt_def	ql_swapfloat_def	
common:		
id	string	Id.
trade_date	date	Trade date.
settle_date (nullable)	date	Settlement date. If this date is not defined the settlecode as defined in the def will be used.
start_date	date	Start date.
maturity (nullable)	date	Maturity (unadjusted).
maturity_code (nullable)	tenor_code	Maturity code. If maturity is not defined this code will be used to calculate the maturity.
pay_float	logical	If true the float leg is paid.
float-leg:		
flt_notional	number	Notional amount for the float leg. If the currency of the two legs are the same this amount must be the same for the two legs.
flt_first_cpn_date (nullable)	date	First coupon date after start date.
flt_last_reg_date (nullable)	date	Last regular coupon date i.e. the last coupon that falls on the regular coupon date schedule. This date and the first coupon date must always occur on the regular coupon date schedule. If maturity is after the last regular coupon date an end stub period is defined.
flt_spread (nullable)	number	Float rate spread. Default = 0.
flt_df (nullable)	disc_func	Discount function.
flt_tenor_surface (nullable)	tenor_surface	Tenor surface for calculating implied forward rates.
flt_cpn_roll_day (nullable)	number	The roll day. If null, it will default to the day of unadjusted flt_last_reg_date, or if this date is null, the unadjusted maturity date.
flt_fx_rate (nullable)	number	Fx-rate. This should be the fx-rate as of the settlement date (or pv

		date). Default = 1.
flt_fixing_dates (nullable)	vector(date)	Fixing date vector.
flt_fixings (nullable)	vector(number)	Fixing rate vector.
flt_fixing_proxy (nullable)	number	If this argument contains a valid rate then it will be used as a fixing rate for all rates that have a reset date before or on the trade date. In other words, it is a way to bypass the need of having a history of rates (flt_fixings) and dates (flt_fixing_dates). Obviously, since this is a single number, the historical cash flows may not be correct. See section <Fixing rates>.
flt_fixing_as_fwd (nullable)	logical	If true the fixing data will be used also for reset dates beyond trade_date. If fixing data is missing the rate will be calculated in the usual way. Default = false.

3.2 Plain vanilla float leg mtm constructor

The “plain vanilla” floatleg mtm constructor contains the most common arguments relevant for a mtm float leg. An mtm (mark-to-market) leg is a leg in cross currency basis swap where the notional is reset at the start of every period based on the changes in the fx rate. It is possible to create a separate float leg with the mtm feature but note that it is necessary to input information from the other currency leg in order to calculate the changes in the notional amount.

Argument	Data type	Description
ql_swap_float_def	ql_swap_float_def	
common:		
id	string	Id.
trade_date	date	Trade date.
settle_date (nullable)	date	Settlement date. If this date is not defined the settlecode as defined in the def will be used.
start_date	date	Start date.
maturity (nullable)	date	Maturity (unadjusted).
maturity_code (nullable)	tenor_code	Maturity code. If maturity is not defined this code will be used to calculate the maturity.
pay_float	logical	If true the float leg is paid.
float-leg:		
flt_notional	number	Notional amount for the float leg. If the currency of the two legs are the same this amount must be the same for the two legs.
flt_first_cpn_date (nullable)	date	First coupon date after start date.
flt_last_reg_date	date	Last regular coupon date i.e. the

(nullable)		last coupon that falls on the regular coupon date schedule. This date and the first coupon date must always occur on the regular coupon date schedule. If maturity is after the last regular coupon date an end stub period is defined.
flt_spread (nullable)	number	Float rate spread. Default = 0.
flt_df (nullable)	disc_func	Discount function.
flt_tenor_surface (nullable)	tenor_surface	Tenor surface for calculating implied forward rates.
flt_cpn_roll_day (nullable)	number	The roll day. If null, it will default to the day of unadjusted flt_last_reg_date, or if this date is null, the unadjusted maturity date.
flt_fx_rate (nullable)	number	Fx-rate. This should be the fx-rate as of the settlement date (or pv date). Default = 1.
flt_fixing_dates (nullable)	vector(date)	Fixing date vector.
flt_fixings (nullable)	vector(number)	Fixing rate vector.
flt_fixing_proxy (nullable)	number	If this argument contains a valid rate then it will be used as a fixing rate for all rates that have a reset date before or on the trade date. In other words, it is a way to bypass the need of having a history of rates (flt_fixings) and dates (flt_fixing_dates). Obviously, since this is a single number, the historical cash flows may not be correct. See section <Fixing rates>.
flt_fixing_as_fwd (nullable)	logical	If true the fixing data will be used also for reset dates beyond trade_date. If fixing data is missing the rate will be calculated in the usual way. Default = false.
flt_mtm_fx_spot_date	date	FX spot date for flt_mtm_fx_spot_rate.
flt_mtm_fx_spot_rate	number	FX spot rate. Quoted as: base currency = the currency of "this" leg and price currency = the currency of the "other" leg.
flt_mtm_fx_base_df	disc_func	Discount function for base currency (currency of "this" leg).
flt_mtm_fx_price_df	disc_func	Discount function for price currency (currency of "other" leg). The base currency and price currency discount functions are, together with flt_mtm_fx_spot_rate, used to calculate the forward fx rates.
flt_mtm_fx_notional		The notional for the "other" leg.

3.3 Plain vanilla fixed leg constructor

The “plain vanilla” fix/float contains the most common arguments relevant for a fixed-float swap instrument.

Argument	Data type	Description
ql_swap_fix_def	ql_swap_fix_def	
common:		
id	string	Id.
trade_date	date	Trade date.
settle_date (nullable)	date	Settlement date. If this date is not defined the settlecode as defined in the def will be used.
start_date	date	Start date.
maturity (nullable)	date	Maturity (unadjusted).
maturity_code (nullable)	tenor_code	Maturity code. If maturity is not defined this code will be used to calculate the maturity.
pay_fixed	logical	If true the fixed leg is paid.
fix-leg:		
fix_notional	number	Notional amount for the fixed leg. If the currency of the two legs are the same this amount must be the same for the two legs.
fix_first_cpn_date (nullable)	date	First coupon date after start date.
fix_last_reg_date (nullable)	date	Last regular coupon date i.e. the last coupon that falls on the regular coupon date schedule. This date and the first coupon date must always occur on the regular coupon date schedule. If maturity is after the last regular coupon date an end stub period is defined.
fix_coupon_rate	number	Coupon rate.
fix_df (nullable)	disc_func	Discount function.
fix_cpn_roll_day (nullable)	number	The roll day. If null, it will default to the day of unadjusted fix_last_reg_date, or if this date is null, the unadjusted maturity date.
fix_fx_rate (nullable)	number	Fx-rate. This should be the fx-rate as of the settlement date (or pv date). Default = 1.

3.4 Plain vanilla ois leg constructor

The “plain vanilla” ois leg constructor contains the most common arguments relevant for an ois leg.

Argument	Data type	Description
ql_swap_ois_def	ql_swap_ois_def	
common:		

id	string	Id.
trade_date	date	Trade date.
settle_date (nullable)	date	Settlement date. If this date is not defined the settlecode as defined in the def will be used.
start_date	date	Start date.
maturity (nullable)	date	Maturity(unadjusted).
maturity_code (nullable)	tenor_code	Maturity code. If maturity is not defined this code will be used to calculate the maturity.
pay_ois	logical	If true the ois leg is paid.
ois-leg:		
ois_notional	number	Notional amount for the ois leg. If the currency of the two legs are the same this amount must be the same for the two legs.
ois_first_cpn_date (nullable)	date	First coupon date after start date.
ois_last_reg_date (nullable)	date	Last regular coupon date i.e. the last coupon that falls on the regular coupon date schedule. This date and the first coupon date must always occur on the regular coupon date schedule. If maturity is after the last regular coupon date an end stub period is defined.
ois_df (nullable)	disc_func	Discount function.
ois_tenor_surface (nullable)	tenor_surface	Tenor surface for calculating implied forward rates.
ois_cpn_roll_day (nullable)	number	The roll day. If null, it will default to the day of unadjusted flt_last_reg_date, or if this date is null, the unadjusted maturity date.
ois_fx_rate (nullable)	number	Fx-rate. This should be the fx-rate as of the settlement date (or pv date). Default = 1.
ois_fixing_dates (nullable)	vector(date)	Fixing date vector.
ois_fixings (nullable)	vector(number)	Fixing rate vector.
ois_fixing_proxy (nullable)	number	If this argument contains a valid rate then it will be used as a fixing rate for all rates that have a reset date before or on the trade date. In other words, it is a way to bypass the need of having a history of rates (flt_fixings) and dates (flt_fixing_dates). Obviously, since this is a single number, the historical cash flows may not be correct. See section <Fixing rates>.
ois_fixing_as_fwd (nullable)	logical	if true the fixing data will be used also for reset dates beyond trade_date. If fixing data is missing the rate will be calculated in the usual way. Default = false.

3.5 Plain vanilla fix_zero leg constructor

The “plain vanilla” fix_zero leg constructor contains the most common arguments relevant for a fixzero-leg.

Argument	Data type	Description
ql_swap_fix_zero_def	ql_swap_fix_zero_def	
common:		
id	string	Id.
trade_date	date	Trade date.
settle_date (nullable)	date	Settlement date. If this date is not defined the settlecode as defined in the def will be used.
start_date	date	Start date.
maturity (nullable)	date	Maturity (unadjusted).
maturity_code (nullable)	tenor_code	Maturity code. If maturity is not defined this code will be used to calculate the maturity.
pay_fixed	logical	If true the fixed zero leg is paid.
fix_zero-leg:		
fix_notional	number	Notional amount for the fixed leg. If the currency of the two legs are the same this amount must be the same for the two legs.
fix_coupon_rate	number	Zero coupon rate.
fix_df (nullable)	disc_func	Discount function.
fix_fx_rate (nullable)	number	Fx-rate. This should be the fx-rate as of the settlement date (or pv date). Default = 1.

4 Instrument Constructors

A swap instrument is normally created by combining two defs with additional “market”-parameters. It is also possible to create a swap from two complete legs.

4.1 Basic Instrument Constructors

This is the most general constructor where the swap is created from two complete legs (i.e. legs that contain both a def and the fix market parameters). The swap legs have either been created independently or extracted from another swap.

Argument	Data type	Description
swap_leg	ql_swap_leg	Swapleg #1.
swap_leg	ql_swap_leg	Swapleg #2.
id	string	Id.
p (nullable)	number	Price
quote_style (nullable)	swap_quote_style	<work-in-progress>

no_check	logical	If the no_check argument is true then the swap will be created from the two legs without performing any checks as to whether the legs are appropriate for a "normal" swap. For example without any checks the legs can have different maturities, start dates, etc.
----------	---------	---

4.2 Extended Instrument Constructors

The extended instrument constructors includes all available arguments, hence they are usually only used if no specific instrument constructor exists.

4.2.1 Extended fix/float

The extended fix/float contains all arguments relevant for a fixed-float swap instrument.

Argument	Data type	Description
swap_fix_def	ql_swap_fix_def	
swapflt_def	ql_swap_float_def	
common:		
id	string	Id.
trade_date	date	Trade date.
settle_date (nullable)	date	Settlement date. If this date is not defined the settlecode as defined in the def will be used.
start_date	date	Start date.
maturity (nullable)	date	Maturity (unadjusted).
maturity_code (nullable)	tenor_code	Maturity code. If maturity is not defined this code will be used to calculate the maturity.
pay_fixed	logical	If true the fixed leg is paid and floating leg received and vice versa.
fix-leg:		
fix_notional	number	Notional amount for the fixed leg. If the currency of the two legs are the same this amount must be the same for the two legs.
fix_first_cpn_date (nullable)	date	First coupon date after start date.
fix_last_reg_date (nullable)	date	Last regular coupon date i.e. the last coupon that falls on the regular coupon date schedule. This date and the first coupon date must always occur on the regular coupon date schedule. If maturity is after the last regular coupon date an end stub period is defined.
fix_coupon_rate	number	Coupon rate.
fix_compounding_rate	number	Relevant for swaps that compounds the

(nullable)		cash flows to maturity. This rate is used as the compounding rate.
fix_df (nullable)	disc_func	Discount function.
fix_start_stub_rate (nullable)	number	If this is a valid number it will be used as coupon rate for start stub periods.
fix_end_stub_rate (nullable)	number	If this is a valid number it will be used as coupon rate for end stub periods.
fix_cpn_roll_day (nullable)	number	The roll day. If null, it will default to the day of unadjusted fix_last_reg_date, or if this date is null, the unadjusted maturity date.
fix_fx_rate (nullable)	number	Fx-rate. This should be the fx-rate as of the settlement date (or pv date). Default = 1.
fix_sink_fund_style (nullable)	sink_fund_style	Type of sinking fund schedule. See section <Sinking funds>.
fix_sink_fund_dates (nullable)	vector(date)	Sinking fund date vector. The dates should correspond to the unadjusted coupon dates. [Currently amortization on non-business days is not supported]. See section <Sinking funds>.
fix_sink_fund (nullable)	vector(number)	Sinking fund value vector. The sign convention is that negative numbers reduces the notional and positive numbers increases the notional i.e. the signs are independent of paying/receiving the cash flows of the swap leg with the sinking fund. See section <Sinking funds>.
fix_fee_dates (nullable)	vector(date)	Fee date vector.
fix_fees (nullable)	vector(number)	Fee vector.
float-leg:		
flt_notional	number	Notional amount for the float leg. If the currency of the two legs are the same this amount must be the same for the two legs.
flt_first_cpn_date (nullable)	date	First coupon date after start date.
flt_last_reg_date (nullable)	date	Last regular coupon date i.e. the last coupon that falls on the regular coupon date schedule. This date and the first coupon date must always occur on the regular coupon date schedule. If maturity is after the last regular coupon date an end stub period is defined.
flt_spread (nullable)	number	Float rate spread.
flt_comp_spread (nullable)	number	Relevant for swaps that compounds the cash flows to maturity or has reset frequency > coupon frequency (and compounding is specified). This spread is used in the compounding.

flt_df (nullable)	disc_func	Discount function.
flt_tenor_surface (nullable)	tenor_surface	Tenor surface for calculating implied forward rates.
flt_start_stub_spread (nullable)	number	If this argument contains a valid spread then it will be used as a spread for a start stub period. This spread will be used as is regardless of the value for start_stub_sprd_decomp. See section <Stub periods>.
flt_end_stub_spread (nullable)	number	If this argument contains a valid spread then it will be used as a spread for an end stub period. This spread will be used as is regardless of the value for end_stub_sprd_decomp. See section <Stub periods>.
flt_cpn_roll_day (nullable)	number	The roll day. If null, it will default to the day of unadjusted flt_last_reg_date, or if this date is null, the unadjusted maturity date.
flt_fx_rate (nullable)	number	Fx-rate. This should be the fx-rate as of the settlement date (or pv date).
flt_sink_fund_style (nullable)	sink_fund_style	Type of sinking fund schedule. See section <Sinking funds>.
flt_sink_fund_dates (nullable)	vector(date)	Sinking fund date vector. The dates should correspond to the unadjusted coupon dates. [Currently amortization on non-business days is not supported]. See section <Sinking funds>.
flt_sink_fund (nullable)	vector(number)	Sinking fund value vector. The sign convention is that negative numbers reduces the notional and positive numbers increases the notional i.e. the signs are independent of paying/receiving the cash flows of the swap leg with the sinking fund. See section <Sinking funds>.
flt_fee_dates (nullable)	vector(date)	Fee date vector.
flt_fees (nullable)	vector(number)	Fee vector.
flt_fixing_dates (nullable)	vector(date)	Fixing date vector.
flt_fixings (nullable)	vector(number)	Fixing rate vector.
flt_fixing_proxy (nullable)	number	If this argument contains a valid rate then it will be used as a fixing rate for all rates that have a reset date before or on the trade date. In other words, it is a way to bypass the need of having a history of rates (flt_fixings) and dates (flt_fixing_dates). Obviously, since this is a single number, the historical cash flows may not be

		correct. See section <Fixing rates>.
flt_fixing_start_stub (nullable)	number	If this argument contains a valid rate, it will be used as a fixing rate for a start stub period. It will override the flt_fixings vector, the flt_fixing_proxy, and any calculations as defined by fixing_stub_style. See section <Fixing rates>.
flt_fixing_end_stub (nullable)	number	If this argument contains a valid rate, it will be used as a fixing rate for an end stub period. It will override the flt_fixings vector, the flt_fixing_proxy, and any calculations as defined by fixing_stub_style. It is only applicable when the reset date for the last stub period is before or on the trade date, i.e. the settlement date is in the last stub period. See section <Fixing rates>.
flt_fixing_as_fwd (nullable)	logical	If true the fixing data will be used also for reset dates beyond trade_date. If fixing data is missing the rate will be calculated in the usual way. Default = false.
p (nullable)	number	Price
quote_style (nullable)	swap_quote_style	<work-in-progress>

4.2.2 Extended fix/fix

The extended fix/fix contains all arguments relevant for a fixed-fixed swap instrument. As 3.1.1 but with two fixed legs.

4.2.3 Extended float/float

The extended float/float contains all arguments relevant for a float-float swap instrument. As 3.1.1 but with two float legs.

4.2.4 Extended fix/ois

The extended fix/ois contains all arguments relevant for a fix-ois swap instrument.

Argument	Data type	Description
swap_fix_def	ql_swap_fix_def	
swap_ois_def	ql_swap_ois_def	
common:		
id	string	Id.
trade_date	date	Trade date.

settle_date (nullable)	date	Settlement date. If this date is not defined the settlecode as defined in the def will be used.
start_date	date	Start date.
maturity (nullable)	date	Maturity (unadjusted).
maturity_code (nullable)	tenor_code	Maturity code. If maturity is not defined this code will be used to calculate the maturity.
pay_fixed	logical	If true the fixed leg is paid and floating leg received and vice versa.
fix-leg:		
fix_notional	number	Notional amount for the fixed leg. If the currency of the two legs are the same this amount must be the same for the two legs.
fix_first_cpn_date (nullable)	date	First coupon date after start date.
fix_last_reg_date (nullable)	date	Last regular coupon date i.e. the last coupon that falls on the regular coupon date schedule. This date and the first coupon date must always occur on the regular coupon date schedule. If maturity is after the last regular coupon date an end stub period is defined.
fix_coupon_rate	number	Coupon rate.
fix_compounding_rate (nullable)	number	Relevant for swaps that compounds the cash flows to maturity. This rate is used as the compounding rate.
fix_df (nullable)	disc_func	Discount function.
fix_start_stub_rate (nullable)	number	If this is a valid number it will be used as coupon rate for start stub periods.
fix_end_stub_rate (nullable)	number	If this is a valid number it will be used as coupon rate for end stub periods.
fix_cpn_roll_day (nullable)	number	The roll day. If null, it will default to the day of unadjusted flt_last_reg_date, or if this date is null, the unadjusted maturity date.
fix_fx_rate (nullable)	number	Fx-rate. This should be the fx-rate as of the settlement date (or pv date). Default = 1.
fix_sink_fund_style (nullable)	sink_fund_style	Type of sinking fund schedule. See section <Sinking funds>.
fix_sink_fund_dates (nullable)	vector(date)	Sinking fund date vector. The dates should correspond to the unadjusted coupon dates. [Currently amortization on non-business days is not supported]. See section <Sinking funds>.
fix_sink_fund (nullable)	vector(number)	Sinking fund value vector. The sign convention is that negative numbers reduces the notional and positive numbers increases the notional i.e. the signs are independent of

		paying/receiving the cash flows of the swap leg with the sinking fund. See section <Sinking funds>.
fix_fee_dates (nullable)	vector(date)	Fee date vector.
fix_fees (nullable)	vector(number)	Fee vector.
ois-leg:		
ois_notional	number	Notional amount for the ois leg. If the currency of the two legs are the same this amount must be the same for the two legs.
ois_first_cpn_date (nullable)	date	First coupon date after start date.
ois_last_reg_date (nullable)	date	Last regular coupon date i.e. the last coupon that falls on the regular coupon date schedule. This date and the first coupon date must always occur on the regular coupon date schedule. If maturity is after the last regular coupon date an end stub period is defined.
ois_spread (nullable)	number	Ois rate spread. Default = 0.
ois_df (nullable)	disc_func	Discount function.
ois_tenor_surface (nullable)	tenor_surface	Tenor surface for calculating implied forward rates.
ois_cpn_roll_day (nullable)	number	The roll day. If null, it will default to the day of unadjusted flt_last_reg_date, or if this date is null, the unadjusted maturity date.
ois_fx_rate (nullable)	number	Fx-rate. This should be the fx-rate as of the settlement date (or pv date). Default = 1.
ois_sink_fund_style (nullable)	sink_fund_style	Type of sinking fund schedule. See section <Sinking funds>.
ois_sink_fund_dates (nullable)	vector(date)	Sinking fund date vector. The dates should correspond to the unadjusted coupon dates. [Currently amortization on non-business days is not supported]. See section <Sinking funds>.
ois_sink_fund (nullable)	vector(number)	Sinking fund value vector. The sign convention is that negative numbers reduces the notional and positive numbers increases the notional i.e. the signs are independent of paying/receiving the cash flows of the swap leg with the sinking fund. See section <Sinking funds>.
ois_fee_dates (nullable)	vector(date)	Fee date vector.
ois_fees (nullable)	vector(number)	Fee vector.
ois_fixing_dates (nullable)	vector(date)	Fixing date vector.
ois_fixings (nullable)	vector(number)	Fixing rate vector.
ois_fixing_proxy (nullable)	number	If this argument contains a valid rate then it will be used as a fixing

		rate for all rates that have a reset date before or on the trade date. In other words, it is a way to bypass the need of having a history of rates (flt_fixings) and dates (flt_fixing_dates). Obviously, since this is a single number, the historical cash flows may not be correct. See section <Fixing rates>.
ois_fixing_as_fwd (nullable)	logical	if true the fixing data will be used also for reset dates beyond trade_date. If fixing data is missing the rate will be calculated in the usual way. Default = false.
p (nullable)	number	Price
quote_style (nullable)	swap_quote_style	<work-in-progress>

4.2.5 Extended fix_zero/float

The extended fix_zero/float contains all arguments relevant for a fixzero-float swap instrument.

Argument	Data type	Description
swap_fix_zero_def	ql_swap_fix_zero_def	
swap_flt_def	ql_swap_flt_def	
common:		
id	string	Id.
trade_date	date	Trade date.
settle_date (nullable)	date	Settlement date. If this date is not defined the settlecode as defined in the def will be used.
start_date	date	Start date.
maturity (nullable)	date	Maturity (unadjusted).
maturity_code (nullable)	tenor_code	Maturity code. If maturity is not defined this code will be used to calculate the maturity.
pay_fixed	logical	If true the fixed zero leg is paid and floating leg received and vice versa.
fix_zero-leg:		
fix_notional	number	Notional amount for the fixed leg. If the currency of the two legs are the same this amount must be the same for the two legs.
fix_coupon_rate	number	Zero coupon rate.
fix_df (nullable)	disc_func	Discount function.
fix_fx_rate (nullable)	number	Fx-rate. This should be the fx-rate as of the settlement date (or pv date). Default = 1.
fix_sink_fund_style (nullable)	sink_fund_style	Type of sinking fund schedule. See section <Sinking funds>.
fix_sink_fund_dates (nullable)	vector(date)	Sinking fund date vector. The dates should correspond to the unadjusted coupon dates. [Currently

		amortization on non-business days is not supported]. See section <Sinking funds>.
fix_sink_fund (nullable)	vector(number)	Sinking fund value vector. The sign convention is that negative numbers reduces the notional and positive numbers increases the notional i.e. the signs are independent of paying/receiving the cash flows of the swap leg with the sinking fund. See section <Sinking funds>.
fix_fee_dates (nullable)	vector(date)	Fee date vector.
fix_fees (nullable)	vector(number)	Fee vector.
float-leg:		
flt_notional	number	Notional amount for the float leg. If the currency of the two legs are the same this amount must be the same for the two legs.
flt_first_cpn_date (nullable)	date	First coupon date after start date.
flt_last_reg_date (nullable)	date	Last regular coupon date i.e. the last coupon that falls on the regular coupon date schedule. This date and the first coupon date must always occur on the regular coupon date schedule. If maturity is after the last regular coupon date an end stub period is defined.
flt_spread (nullable)	number	Float rate spread. Default = 0.
flt_comp_spread (nullable)	number	Relevant for swaps that compounds the cash flows to maturity. This spread is used in the compounding.
flt_df (nullable)	disc func	Discount function.
flt_tenor_surface (nullable)	tenor_surface	Tenor surface for calculating implied forward rates.
flt_start_stub_spread (nullable)	number	If this argument contains a valid spread then it will be used as a spread for a start stub period. This spread will be used as is regardless of the value for start_stub_sprd_decomp. See section <Stub periods>.
flt_end_stub_spread (nullable)	number	If this argument contains a valid spread then it will be used as a spread for an end stub period. This spread will be used as is regardless of the value for end_stub_sprd_decomp. See section <Stub periods>.
flt_cpn_roll_day (nullable)	number	The roll day. If null, it will default to the day of unadjusted flt_last_reg_date, or if this date is null, the unadjusted maturity date.
flt_fx_rate	number	Fx-rate. This should be the fx-rate

(nullable)		as of the settlement date (or pv date). Default = 1.
flt_sink_fund_style (nullable)	sink_fund_style	Type of sinking fund schedule. See section <Sinking funds>.
flt_sink_fund_dates (nullable)	vector(date)	Sinking fund date vector. The dates should correspond to the unadjusted coupon dates. [Currently amortization on non-business days is not supported]. See section <Sinking funds>.
flt_sink_fund (nullable)	vector(number)	Sinking fund value vector. The sign convention is that negative numbers reduces the notional and positive numbers increases the notional i.e. the signs are independent of paying/receiving the cash flows of the swap leg with the sinking fund. See section <Sinking funds>.
flt_fee_dates (nullable)	vector(date)	Fee date vector.
flt_fees (nullable)	vector(number)	Fee vector.
flt_fixing_dates (nullable)	vector(date)	Fixing date vector.
flt_fixings (nullable)	vector(number)	Fixing rate vector.
flt_fixing_proxy (nullable)	number	If this argument contains a valid rate then it will be used as a fixing rate for all rates that have a reset date before or on the trade date. In other words, it is a way to bypass the need of having a history of rates (flt_fixings) and dates (flt_fixing_dates). Obviously, since this is a single number, the historical cash flows may not be correct. See section <Fixing rates>.
flt_fixing_start_stub (nullable)	number	If this argument contains a valid rate, it will be used as a fixing rate for a start stub period. It will override the flt_fixings vector, the flt_fixing_proxy, and any calculations as defined by fixing_stub_style. See section <Fixing rates>.
flt_fixing_end_stub (nullable)	number	If this argument contains a valid rate, it will be used as a fixing rate for an end stub period. It will override the flt_fixings vector, the flt_fixing_proxy, and any calculations as defined by fixing_stub_style. It is only applicable when the reset date for the last stub period is before or on the trade date, i.e. the settlement date is in the last stub

		period. See section <Fixing rates>.
flt_fixing_as_fwd (nullable)	logical	If true the fixing data will be used also for reset dates beyond trade_date. If fixing data is missing the rate will be calculated in the usual way. Default = false.
p (nullable)	number	Price
quote_style (nullable)	swap_quote_style	<work-in-progress>

4.3 Specific Instrument Constructors

A specific instrument contains a subset of the parameters in the extended instrument i.e. some parameters are set to appropriate default values depending on the type of instrument.

4.3.1 Plain vanilla fix/float

The “plain vanilla” fix/float contains the most common arguments relevant for a fixed-float swap instrument.

Argument	Data type	Description
swap fix def	ql swap fix def	
swap flt def	ql swap float def	
common:		
id	string	Id.
trade_date	date	Trade date.
settle_date (nullable)	date	Settlement date. If this date is not defined the settlecode as defined in the def will be used.
start_date	date	Start date.
maturity (nullable)	date	Maturity (unadjusted).
maturity_code (nullable)	tenor_code	Maturity code. If maturity is not defined this code will be used to calculate the maturity.
pay_fixed	logical	If true the fixed leg is paid and floating leg received and vice versa.
fix-leg:		
fix_notional	number	Notional amount for the fixed leg. If the currency of the two legs are the same this amount must be the same for the two legs.
fix_first_cpn_date (nullable)	date	First coupon date after start date.
fix_last_reg_date (nullable)	date	Last regular coupon date i.e. the last coupon that falls on the regular coupon date schedule. This date and the first coupon date must always occur on the regular coupon date schedule. If maturity is after the last regular coupon date an end stub period is defined.

fix_coupon_rate	number	Coupon rate.
fix_df (nullable)	disc_func	Discount function.
fix_cpn_roll_day (nullable)	number	The roll day. If null, it will default to the day of unadjusted fix_last_reg_date, or if this date is null, the unadjusted maturity date.
fix_fx_rate (nullable)	number	Fx-rate. This should be the fx-rate as of the settlement date (or pv date). Default = 1.
float-leg:		
flt_notional	number	Notional amount for the float leg. If the currency of the two legs are the same this amount must be the same for the two legs.
flt_first_cpn_date (nullable)	date	First coupon date after start date.
flt_last_reg_date (nullable)	date	Last regular coupon date i.e. the last coupon that falls on the regular coupon date schedule. This date and the first coupon date must always occur on the regular coupon date schedule. If maturity is after the last regular coupon date an end stub period is defined.
flt_spread (nullable)	number	Float rate spread. Default = 0.
flt_df (nullable)	disc_func	Discount function.
flt_tenor_surface (nullable)	tenor_surface	Tenor surface for calculating implied forward rates.
flt_cpn_roll_day (nullable)	number	The roll day. If null, it will default to the day of unadjusted flt_last_reg_date, or if this date is null, the unadjusted maturity date.
flt_fx_rate (nullable)	number	Fx-rate. This should be the fx-rate as of the settlement date (or pv date). Default = 1.
flt_fixing_dates (nullable)	vector(date)	Fixing date vector.
flt_fixings (nullable)	vector(number)	Fixing rate vector.
flt_fixing_proxy (nullable)	number	If this argument contains a valid rate then it will be used as a fixing rate for all rates that have a reset date before or on the trade date. In other words, it is a way to bypass the need of having a history of rates (flt_fixings) and dates (flt_fixing_dates). Obviously, since this is a single number, the historical cash flows may not be correct. See section <Fixing rates>.
flt_fixing_as_fwd (nullable)	logical	If true the fixing data will be used also for reset dates beyond trade_date. If fixing data is missing the rate will be calculated in the usual way. Default = false.

4.3.2 Plain vanilla fix/float with external coupon flows

The “plain vanilla” fix/float with external coupon flows contains the most common arguments relevant for a fixed-float swap instrument and which allows for externally defined coupon cashflow amounts.

Argument	Data type	Description
swap fix def	ql_swap_fix_def	
swap flt def	ql_swap_float_def	
common:		
id	string	Id.
trade_date	date	Trade date.
settle_date (nullable)	date	Settlement date. If this date is not defined the settlecode as defined in the def will be used.
start_date	date	Start date.
maturity (nullable)	date	Maturity (unadjusted).
maturity_code (nullable)	tenor_code	Maturity code. If maturity is not defined this code will be used to calculate the maturity.
pay_fixed	logical	If true the fixed leg is paid and floating leg received and vice versa.
fix-leg:		
fix_notional	number	Notional amount for the fixed leg. If the currency of the two legs are the same this amount must be the same for the two legs.
fix_first_cpn_date (nullable)	date	First coupon date after start date.
fix_last_reg_date (nullable)	date	Last regular coupon date i.e. the last coupon that falls on the regular coupon date schedule. This date and the first coupon date must always occur on the regular coupon date schedule. If maturity is after the last regular coupon date an end stub period is defined.
fix_coupon_flows	vector(number)	Coupon cashflows. The size of the vector should correspond to the number of coupon payments. The cashflow amounts should be input as positive amounts (assuming the underlying coupon rate is positive).
fix_df (nullable)	disc_func	Discount function.
fix_cpn_roll_day (nullable)	number	The roll day. If null, it will default to the day of unadjusted fix_last_reg_date, or if this date is null, the unadjusted maturity date.
fix_fx_rate (nullable)	number	Fx-rate. This should be the fx-rate as of the settlement date (or pv date). Default = 1.
float-leg:		
flt_notional	number	Notional amount for the float leg. If the currency of the two legs are the same this amount must be the same for

		the two legs.
flt_first_cpn_date (nullable)	date	First coupon date after start date.
flt_last_reg_date (nullable)	date	Last regular coupon date i.e. the last coupon that falls on the regular coupon date schedule. This date and the first coupon date must always occur on the regular coupon date schedule. If maturity is after the last regular coupon date an end stub period is defined.
flt_spread (nullable)	number	Float rate spread. Default = 0.
flt_df (nullable)	disc_func	Discount function.
flt_tenor_surface (nullable)	tenor_surface	Tenor surface for calculating implied forward rates.
flt_cpn_roll_day (nullable)	number	The roll day. If null, it will default to the day of unadjusted flt_last_reg_date, or if this date is null, the unadjusted maturity date.
flt_fx_rate (nullable)	number	Fx-rate. This should be the fx-rate as of the settlement date (or pv date). Default = 1.
flt_fixing_dates (nullable)	vector(date)	Fixing date vector.
flt_fixings (nullable)	vector(number)	Fixing rate vector.
flt_fixing_proxy (nullable)	number	If this argument contains a valid rate then it will be used as a fixing rate for all rates that have a reset date before or on the trade date. In other words, it is a way to bypass the need of having a history of rates (flt_fixings) and dates (flt_fixing_dates). Obviously, since this is a single number, the historical cash flows may not be correct. See section <Fixing rates>.
flt_fixing_as_fwd (nullable)	logical	If true the fixing data will be used also for reset dates beyond trade_date. If fixing data is missing the rate will be calculated in the usual way. Default = false.

4.3.3 Plain vanilla fix/float with sinkfunds and fees

The “plain vanilla” fix/float with sinkfunds and fees contains the most common arguments relevant for a fixed-float swap instrument with sinking funds and fees.

Argument	Data type	Description
swap_fix_def	ql_swap_fix_def	
swap_flt_def	ql_swap_float_def	
common:		
id	string	Id.
trade_date	date	Trade date.

settle_date (nullable)	date	Settlement date. If this date is not defined the settlecode as defined in the def will be used.
start_date	date	Start date.
maturity (nullable)	date	Maturity (unadjusted).
maturity_code (nullable)	tenor_code	Maturity code. If maturity is not defined this code will be used to calculate the maturity.
pay_fixed	logical	If true the fixed leg is paid and floating leg received and vice versa.
fix-leg:		
fix_notional	number	Notional amount for the fixed leg. If the currency of the two legs are the same this amount must be the same for the two legs.
fix_first_cpn_date (nullable)	date	First coupon date after start date.
fix_last_reg_date (nullable)	date	Last regular coupon date i.e. the last coupon that falls on the regular coupon date schedule. This date and the first coupon date must always occur on the regular coupon date schedule. If maturity is after the last regular coupon date an end stub period is defined.
fix_coupon_rate	number	Coupon rate.
fix_df (nullable)	disc_func	Discount function.
fix_cpn_roll_day (nullable)	number	The roll day. If null, it will default to the day of unadjusted fix_last_reg_date, or if this date is null, the unadjusted maturity date.
fix_fx_rate (nullable)	number	Fx-rate. This should be the fx-rate as of the settlement date (or pv date). Default = 1.
fix_sink_fund_style (nullable)	sink_fund_style	Type of sinking fund schedule. See section <Sinking funds>.
fix_sink_fund_dates (nullable)	vector(date)	Sinking fund date vector. The dates should correspond to the unadjusted coupon dates. [Currently amortization on non-business days is not supported]. See section <Sinking funds>.
fix_sink_fund (nullable)	vector(number)	Sinking fund value vector. The sign convention is that negative numbers reduces the notional and positive numbers increases the notional i.e. the signs are independent of paying/receiving the cash flows of the swap leg with the sinking fund. See section <Sinking funds>.
fix_fee_dates (nullable)	vector(date)	Fee date vector.
fix_fees (nullable)	vector(number)	Fee vector.
float-leg:		
flt_notional	number	Notional amount for the float leg. If the currency of the two legs are the

		same this amount must be the same for the two legs.
flt_first_cpn_date (nullable)	date	First coupon date after start date.
flt_last_reg_date (nullable)	date	Last regular coupon date i.e. the last coupon that falls on the regular coupon date schedule. This date and the first coupon date must always occur on the regular coupon date schedule. If maturity is after the last regular coupon date an end stub period is defined.
flt_spread (nullable)	number	Float rate spread. Default = 0.
flt_df (nullable)	disc_func	Discount function.
flt_tenor_surface (nullable)	tenor_surface	Teneor surface for calculating implied forward rates.
flt_cpn_roll_day (nullable)	number	The roll day. If null, it will default to the day of unadjusted flt_last_reg_date, or if this date is null, the unadjusted maturity date.
flt_fx_rate (nullable)	number	Fx-rate. This should be the fx-rate as of the settlement date (or pv date). Default = 1.
flt_sink_fund_style (nullable)	sink_fund_style	Type of sinking fund schedule. See section <Sinking funds>.
flt_sink_fund_dates (nullable)	vector(date)	Sinking fund date vector. The dates should correspond to the unadjusted coupon dates. [Currently amortization on non-business days is not supported]. See section <Sinking funds>.
flt_sink_fund (nullable)	vector(number)	Sinking fund value vector. The sign convention is that negative numbers reduces the notional and positive numbers increases the notional i.e. the signs are independent of paying/receiving the cash flows of the swap leg with the sinking fund. See section <Sinking funds>.
flt_fee_dates (nullable)	vector(date)	Fee date vector.
flt_fees (nullable)	vector(number)	Fee vector.
flt_fixing_dates (nullable)	vector(date)	Fixing date vector.
flt_fixings (nullable)	vector(number)	Fixing rate vector.
flt_fixing_proxy (nullable)	number	If this argument contains a valid rate then it will be used as a fixing rate for all rates that have a reset date before or on the trade date. In other words, it is a way to bypass the need of having a history of rates (flt_fixings) and dates (flt_fixing_dates). Obviously, since this is a single number, the historical cash flows may not be correct. See section <Fixing rates>.

flt_fixing_as_fwd (nullable)	logical	If true the fixing data will be used also for reset dates beyond trade_date. If fixing data is missing the rate will be calculated in the usual way. Default = false.
---------------------------------	---------	---

4.3.4 Plain vanilla fix/fix

The “plain vanilla” fix/fix contains the most common arguments relevant for a fixed-fixed swap instrument. As 4.3.1 but with two fixed legs.

4.3.5 Plain vanilla float/float

The “plain vanilla” float/float contains the most common arguments relevant for a float-float swap instrument. As 4.3.1 but with two float legs.

4.3.6 Plain vanilla float/float mtm

The “plain vanilla” float/float contains the most common arguments relevant for a float-float swap instrument with a mark-to-market feature, commonly used in currency basis swaps.

Argument	Data type	Description
swap_flt1_def	ql_swap_float_def	
swap_flt2_def	ql_swap_float_def	
common:		
id	string	Id.
trade_date	date	Trade date.
settle_date (nullable)	date	Settlement date. If this date is not defined the settlecode as defined in the def will be used.
start_date	date	Start date.
maturity (nullable)	date	Maturity (unadjusted).
maturity_code (nullable)	tenor_code	Maturity code. If maturity is not defined this code will be used to calculate the maturity.
pay_leg1	logical	If true the first float leg is paid and second floating leg received and vice versa.
float-leg1:		
flt1_notional	number	Notional amount for the float leg. If the currency of the two legs are the same this amount must be the same for the two legs.
flt1_first_cpn_date (nullable)	date	First coupon date after start date.
flt1_last_reg_date (nullable)	date	Last regular coupon date i.e. the last coupon that falls on the regular coupon date schedule. This date and the first coupon date must always

		occur on the regular coupon date schedule. If maturity is after the last regular coupon date an end stub period is defined.
flt1_spread (nullable)	number	Float rate spread. Default = 0.
flt1_df (nullable)	disc_func	Discount function.
flt1_tenor_surface (nullable)	tenor_surface	Tenor surface for calculating implied forward rates.
flt1_cpn_roll_day (nullable)	number	The roll day. If null, it will default to the day of unadjusted flt_last_reg_date, or if this date is null, the unadjusted maturity date.
flt1_fx_rate (nullable)	number	Fx-rate. This should be the fx-rate as of the settlement date (or pv date). Default = 1.
flt1_fixing_dates (nullable)	vector(date)	Fixing date vector.
flt1_fixings (nullable)	vector(number)	Fixing rate vector.
flt1_fixing_proxy (nullable)	number	If this argument contains a valid rate then it will be used as a fixing rate for all rates that have a reset date before or on the trade date. In other words, it is a way to bypass the need of having a history of rates (flt_fixings) and dates (flt_fixing_dates). Obviously, since this is a single number, the historical cash flows may not be correct. See section <Fixing rates>.
flt1_fixing_as_fwd (nullable)	logical	If true the fixing data will be used also for reset dates beyond trade_date. If fixing data is missing the rate will be calculated in the usual way. Default = false.
flt1_mtm_fx_spot_date	date	FX spot date for flt1_mtm_fx_spot_rate.
flt1_mtm_fx_spot_rate	number	FX spot rate. Quoted as: base currency = the currency of float leg 1 and price currency = the currency of float leg 2.
flt1_mtm_fx_base_df	disc_func	Discount function for base currency (currency of float leg 1).
flt1_mtm_fx_price_df	disc_func	Discount function for price currency (currency of float leg 2). The base currency and price currency discount functions are, together with flt1_mtm_fx_spot_rate, used to calculate the forward fx rates.
float-leg2:		
float-leg2 has the same arguments as in section 4.3.1		

4.3.7 Plain vanilla fix/ois

The “plain vanilla” fix/ois contains the most common arguments relevant for a fixed-ois swap instrument.

Argument	Data type	Description
swap_fix_def	ql_swap_fix_def	
swap_ois_def	ql_swap_ois_def	
common:		
id	string	Id.
trade_date	date	Trade date.
settle_date (nullable)	date	Settlement date. If this date is not defined the settlecode as defined in the def will be used.
start_date	date	Start date.
maturity (nullable)	date	Maturity(unadjusted).
maturity_code (nullable)	tenor_code	Maturity code. If maturity is not defined this code will be used to calculate the maturity.
pay_fixed	logical	If true the fixed leg is paid and floating leg received and vice versa.
fix-leg:		
fix_notional	number	Notional amount for the fixed leg. If the currency of the two legs are the same this amount must be the same for the two legs.
fix_first_cpn_date (nullable)	date	First coupon date after start date.
fix_last_reg_date (nullable)	date	Last regular coupon date i.e. the last coupon that falls on the regular coupon date schedule. This date and the first coupon date must always occur on the regular coupon date schedule. If maturity is after the last regular coupon date an end stub period is defined.
fix_coupon_rate	number	Coupon rate.
fix_df (nullable)	disc_func	Discount function.
fix_cpn_roll_day (nullable)	number	The roll day. If null, it will default to the day of unadjusted flt_last_reg_date, or if this date is null, the unadjusted maturity date.
fix_fx_rate (nullable)	number	Fx-rate. This should be the fx-rate as of the settlement date (or pv date). Default = 1.
ois-leg:		
ois_notional	number	Notional amount for the ois leg. If the currency of the two legs are the same this amount must be the same for the two legs.
ois_first_cpn_date (nullable)	date	First coupon date after start date.
ois_last_reg_date (nullable)	date	Last regular coupon date i.e. the last coupon that falls on the regular

		coupon date schedule. This date and the first coupon date must always occur on the regular coupon date schedule. If maturity is after the last regular coupon date an end stub period is defined.
ois_df (nullable)	disc_func	Discount function.
ois_tenor_surface (nullable)	tenor_surface	Tenor surface for calculating implied forward rates.
ois_cpn_roll_day (nullable)	number	The roll day. If null, it will default to the day of unadjusted flt_last_reg_date, or if this date is null, the unadjusted maturity date.
ois_fx_rate (nullable)	number	Fx-rate. This should be the fx-rate as of the settlement date (or pv date). Default = 1.
ois_fixing_dates (nullable)	vector(date)	Fixing date vector.
ois_fixings (nullable)	vector(number)	Fixing rate vector.
ois_fixing_proxy (nullable)	number	If this argument contains a valid rate then it will be used as a fixing rate for all rates that have a reset date before or on the trade date. In other words, it is a way to bypass the need of having a history of rates (flt_fixings) and dates (flt_fixing_dates). Obviously, since this is a single number, the historical cash flows may not be correct. See section <Fixing rates>.
ois_fixing_as_fwd (nullable)	logical	if true the fixing data will be used also for reset dates beyond trade_date. If fixing data is missing the rate will be calculated in the usual way. Default = false.

4.3.8 Plain vanilla fix_zero/float

The “plain vanilla” fix_zero/float contains the most common arguments relevant for a fixzero-float swap instrument.

Argument	Data type	Description
swap_fix_zero_def	ql_swap_fix_zero_def	
swap_flt_def	ql_swap_flt_def	
common:		
id	string	Id.
trade_date	date	Trade date.
settle_date (nullable)	date	Settlement date. If this date is not defined the settlecode as defined in the def will be used.
start_date	date	Start date.
maturity (nullable)	date	Maturity (unadjusted).
maturity_code	tenor_code	Maturity code. If maturity is not

(nullable)		defined this code will be used to calculate the maturity.
pay_fixed	logical	If true the fixed zero leg is paid and floating leg received and vice versa.
fix_zero-leg:		
fix_notional	number	Notional amount for the fixed leg. If the currency of the two legs are the same this amount must be the same for the two legs.
fix_coupon_rate	number	Zero coupon rate.
fix_df (nullable)	disc_func	Discount function.
fix_fx_rate (nullable)	number	Fx-rate. This should be the fx-rate as of the settlement date (or pv date). Default = 1.
float-leg:		
flt_notional	number	Notional amount for the float leg. If the currency of the two legs are the same this amount must be the same for the two legs.
flt_first_cpn_date (nullable)	date	First coupon date after start date.
flt_last_reg_date (nullable)	date	Last regular coupon date i.e. the last coupon that falls on the regular coupon date schedule. This date and the first coupon date must always occur on the regular coupon date schedule. If maturity is after the last regular coupon date an end stub period is defined.
flt_spread (nullable)	number	Float rate spread. Default = 0.
flt_df (nullable)	disc_func	Discount function.
flt_tenor_surface (nullable)	tenor_surface	tenor surface for calculating implied forward rates.
flt_cpn_roll_day (nullable)	number	The roll day. If null, it will default to the day of unadjusted flt_last_reg_date, or if this date is null, the unadjusted maturity date.
flt_fx_rate (nullable)	number	Fx-rate. This should be the fx-rate as of the settlement date (or pv date). Default = 1.
flt_fixing_dates (nullable)	vector(date)	Fixing date vector.
flt_fixings (nullable)	vector(number)	Fixing rate vector.
flt_fixing_proxy (nullable)	number	If this argument contains a valid rate then it will be used as a fixing rate for all rates that have a reset date before or on the trade date. In other words, it is a way to bypass the need of having a history of rates (flt_fixings) and dates (flt_fixing_dates). Obviously, since this is a single number, the historical cash flows

		may not be correct. See section <Fixing rates>.
flt_fixing_as_fwd (nullable)	logical	If true the fixing data will be used also for reset dates beyond trade_date. If fixing data is missing the rate will be calculated in the usual way. Default = false.

5 Data types

5.1 Interest calculation rule

Argument	Data type	Location	Description
interest_rule	interest_rule	ql_swap_float_def, ql_swap_fix_def	Rule for calculating interest.

interest_rule	Description
IR_ADJ	Interest is calculated up to the period end date as adjusted by the businessday convention.
IR_NOADJ	Interest is calculated up to the unadjusted period end date. (if the unadjusted day occurs on a non-business day, interest will be calculated up to this day but will be paid on the adjusted period end date).
IR_MAT_NOADJ	Interest is calculated up to the period end date as adjusted by the businessday convention except for the last period where interest is calculated up to the unadjusted maturity date. This is only applicable when maturity is on a non-business day. This is the default.

For all cases above the first interest accrual period starts on the start date as it is input i.e. it is not adjusted to be a businessday. Normally, the start date is always a businessday.

5.2 The implied forward rate

The periods will be defined by the roll method.

Argument	Data type	Location	Description
roll_method	flt_roll_method	ql_swap_float_def	Determines the coupon dates.

Flt_roll_method	Description
FLR_STRAIGHT	

FLR_IMM	
FLR_CHAIN	
FLR_ZERO	

The length of the period can be of two types.

Argument	Data type	Location	Description
fwd_rate_style	flt_fwd_style	ql_swap_float_def	Determines start and end date for the implied forward rate calculation.

flt_fwd_style	Description
FF_IDX	The end date for the implied forward rate is calculated by adding a period according to the floating rate tenor. This means that in many cases the end date of the period used for calculating the forward rate is not the same as end date of the coupon period (this is so even in the case that the coupon frequency is the same as the index rate frequency). This is the default.
FF_CPN_DATES	The implied forward rate is calculated for the period between the coupon dates i.e. the start and end date corresponds to the coupon dates.

The reset date for the rate and hence the rate period can be of different types as well.

reset_style	flt_reset_style	ql_swap_float_def	Type of forward rate.
-------------	-----------------	-------------------	-----------------------

flt_reset_style	Description
FRS_NORM	Normal reset meaning that that the rate is reset in the beginning of the payment period. This is the default.
FRS_ARR	Reset in arrears meaning that that the rate is reset at the end of the payment period.
FRS_ADV	Advance reset meaning that that the rate is reset in the beginning of the period before the payment period. The first period, however, is always of the normal style (FRS_NORM).

The type of reset rate can also be defined and does not have to follow the same description used when calculating the cash flow for the floating leg.

Argument	Data type	Location	Description
index_tenor	date_code	ql_swap_float_def	The tenor for the floating rate index. If this is not defined it will default to the tenor

			implied by the coupon frequency or the reset frequency (if this is more frequent)
index_rate_type	flt_index_rate_type	ql_swap_float_def	Type of rate for the index.
index_cpn_freq	coupon_freq	ql_swap_float_def	Coupon frequency of the index rate. Only applicable when index_rate_type is of type swap rate.
index_daycount	day_count_method	ql_swap_float_def	Daycount method of the index rate.
index_bus_day	bd_convention	ql_swap_float_def	Businessday convention of the index rate.
index_calendar	calendar	ql_swap_float_def	Calendar for the index.
index_eom	logical	ql_swap_float_def	End-of-month setting for the index.

Flt index rate type	Description
IRT SIMPLE	Simple rate.
IRT SWAP	Swap rate. Used for CMS.

In addition, the rate can be converted.

Argument	Data type	Location	Description
sub_factor	number	ql_swap_float_def	The rate will be calculated as sub_factor (C) minus reset rate (r) i.e. C-r.
multi_factor	number	ql_swap_float_def	The rate will be calculated as multi_factor (M) times the reset rate (r) i.e. M*r.
power_factor	number	ql_swap_float_def	The rate will be calculated as the reset rate (r) raised to the power = power_factor (P) i.e. r^P.

All three types of rate manipulations can be used at the same time in which case the reset rate will be

$$\text{Rate} = C - M * r^P.$$

5.3 The fixed rate

The periods will be defined by the roll method.

Argument	Data type	Location	Description
roll_method	fix_roll_method	ql_swap_fix_def	Determines the coupon dates.

fix_roll_method	Description
FIR_STRAIGHT	
FIR_IMM	
FIR_ZERO	

Argument	Data type	Location	Description
compound_freq	compound_freq	ql_swap_fix_def	Determines the compounding frequency of the swap coupon rate.

compound_freq	Description
FIC_FREQ	The compounding frequency is the same as coupon frequency.
FIC_ANNUAL	The compounding frequency is annual regardless of coupon frequency.

5.4 Reset frequency greater than coupon frequency

In this case, the calculations on the reset will be either of a compounding type or of an averaging type. When compounding is used, the amounts calculated on the reset dates will be compounded to the next payment date and when averaging is used, the reset rates will be averaged and the average will be used for calculation of the interest on the payment date.

Argument	Data type	Location	Description
comp_avg	flt_comp_avg_type	ql_swap_float_def, ql_swap_ois_def	Type of method.
spread_comp_method	flt_sprd_comp_method	ql_swap_float_def, ql_swap_ois_def	Type of compounding method.
avg_method	flt_avg_method	ql_swap_float_def, ql_swap_ois_def	Type of averaging method.

flt_comp_avg_type	Description
CAT_COMP	Compounding according to flt_sprd_comp_method will be used
CAT_AVG	Averaging according to flt_avg_method will be used

flt_sprd_comp_method	Description
SCM_SPRD_INCL	ISDA-method. The reset rate plus the spread is the effective rate for compounding. Spread amounts are compounded. This convention is called "straight compounding" in ISDA terminology.
SCM_SPRD_EXCL	The flat reset rate is the effective rate for compounding. Spread amounts are not compounded.

	This convention is called "compounding with simple spread" in ISDA terminology but is currently (2009) not defined by ISDA.
SCM_FLAT	ISDA-method. The flat reset rate is the effective rate for compounding. Spread amounts are compounded. This convention is called "flat compounding" in ISDA terminology.

flt_avg_method	Description
AVGM_UNWEIGHTED	An arithmetic mean of the reset rates during the period.
AVGM_WEIGHTED	Each reset rate is weighted by the length of the period it is in effect.
AVGM_SIMPLE	An arithmetic mean of the reset rates that have a reset date from the start date of the period (exclusive) to the end date of the period (exclusive). Compared to AVGM_WEIGHTED there is one less reset rate included in the average.
AVGM_BUS_WEIGHTED	Each reset rate during a period is weighted by the number of business days in the period.

5.5 Stub periods

Argument	Data type	Location	Description
fwd_stub_rate_style	flt_stub_fwd_style	ql_swap_float_def	Determines start and end date for the implied forward rate calculation in a stub (start or end) period.
start_stub_sprd_decomp	logical	ql_swap_float_def	If true, the floating rate spread will be decompounded for a start stub period, otherwise the spread is used as input.
end_stub_sprd_decomp	logical	ql_swap_float_def	If true, the floating rate spread will be decompounded for an end stub period, otherwise the spread is used as input.
flt_start_stub_spread	number	Instrument	If this argument contains a valid spread then it will be used as a spread for a start stub period. This spread will be used as is regardless of the value for start_stub_sprd_decomp.
flt_end_stub_spread	number	Instrument	If this argument contains a valid spread

			then it will be used as a spread for an end stub period. This spread will be used as is regardless of the value for end_stub_sprd_decomp.
--	--	--	---

flt_stub_fwd_style	Description
FSF_IDX	The implied forward rate is calculated by adding a period according to the floating rate tenor. This means that the end date of the period used for calculating the forward rate is not the same as end date of the coupon period. This is the default if flt_fwd_style is not equal to FF_CPN_DATES.
FSF_CPN_DATES	The implied forward rate is calculated between the coupon dates i.e. the start and end date corresponds to the coupon dates. This is the default if flt_fwd_style is equal to FF_CPN_DATES.
FSF_IDX_DECOMP	The implied forward rate is the same as for FSF_IDX but is decomposed to the period length for the stub period.

5.6 Fixing rates

Argument	Data type	Location	Description
flt_fixing_dates	vector(date)	Instrument	Fixing date vector. The dates needs to match the reset dates i.e. normally 2 days before the payment date. The reset dates can be retrieved with the function reset_dates() (see below).
flt_fixings	vector(number)	Instrument	Fixing rate vector.
flt_fixing_proxy	number	Instrument	If this argument contains a valid rate then it will be used as a fixing rate for all rates that have a reset date before or on the trade date. In other words, it is a way to bypass the need of having a history of rates (flt_fixings) and dates (flt_fixing_dates). Obviously, since this

			is a single number, the historical cash flows may not be correct.
flt_fixing_start_stub	number	Instrument	If this argument contains a valid rate, it will be used as a fixing rate for a start stub period. It will override the flt_fixings vector, the flt_fixing_proxy, and any calculations as defined by fixing_stub_style.
flt_fixing_end_stub	number	Instrument	If this argument contains a valid rate, it will be used as a fixing rate for an end stub period. It will override the flt_fixings vector, the flt_fixing_proxy, and any calculations as defined by fixing_stub_style. It is only applicable when the reset date for the last stub period is before or on the trade date, i.e. the settlement date is in the last stub period.
flt_fixing_as_fwd	logical	Instrument	If true the fixing data will be used also for reset dates beyond trade_date. If fixing data is missing the rate will be calculated in the usual way. This allows overriding of the calculation of implied forward rates via a discount function.
fixing_stub_style	fixing_stub_style	ql_swap_float_def	Determines how fixing rates are handled for stub periods. Note, values entered for flt_fixing_start_stub or flt_fixing_end_stub will not be subject to this setting.

fixing stub style	Description
FSC_IDX	The fixing rate is used as is. This is the default.
FSC_IDX_IP	The fixing rate is interpolated from the relevant fixing indexes. Ex. if the stub period is 8 months it may be interpolated between a 6-month and a 9-month index. [Currently not implemented]
FSC_IDX_DECOMP	The fixing rate is decomposed to the period length for the stub period.

5.7 Tenor surface

The tenor surface contains the forward curves for relevant tenors.

5.8 Exchange of notional

Applicable for most types of swap legs.

Argument	Data type	Location	Description
notional_exchange	notional_exchg_style	ql_swap_float_def, ql_swap_fix_def, ql_swap_ois_def, ql_swap_zero_def	Type of notional exchange.

notional_exchg_style	Description
NE_NONE	No notional amounts will be exchanged.
NE_START	The notional amount on the start date only is exchanged.
NE_FINAL	The notional amount on maturity date only is exchanged. For amortizing swaps, all notional amounts after start date will be exchanged.
NE_BOTH	All notional amounts will be exchanged.

5.9 Sinking funds

Applicable for most types of swap legs.

Argument	Data type	Location	Description
sink_fund_style	sink_fund_style	Instrument	Type of sinking fund schedule.

sink_fund_dates	vector(date)	Instrument	Sinking fund date vector. The dates should correspond to the unadjusted coupon dates. The unadjusted dates can be retrieved with the function unadj_dates() (see below). [Currently amortization on non-business days is not supported]
sink_fund	vector(number)	Instrument	Sinking fund value vector. The sign convention is that negative numbers reduces the notional and positive numbers increases the notional i.e. the signs are independent of paying/receiving the cash flows of the swap leg with the sinking fund.
average_notional	logical	ql_swap_float_def, ql_swap_fix_def	If a sinking fund schedule has a greater frequency than the payment frequency then the notional will be either a weighted average (weighted by the length of the period in the sinking fund schedule) or be the notional as of the beginning of the coupon period, depending on the value for average_notional.

sink_fund_style	Description
SF_NONE	No sinking funds. Neither the sinking fund date vector nor the sinking fund value vector is used.
SF_FIXED	A fixed amount of amortization for all dates or, optionally, between a start date and end date. At maturity, the remaining of the notional is amortized and this sum may or may not be the same as the fixed amount. The fixed amount is given as the first element in the sinking fund value vector. If the sinking fund value vector is larger than one element, an error will be returned. Amount < 0 reduces the notional and amount > 0 increases the notional. The optional start date and end date should be entered as the first two elements in the sinking fund date vector. If the sinking fund date vector is not defined (null or zero

	elements) the fixed amount amortization will occur for all dates. If the sinking fund date vector has a size > 2, an error will be returned.
SF_FIXED_PCNT	A fixed percentage (as a decimal number) of outstanding notional for all dates or, optionally, between a start date and end date. At maturity, the remaining of the notional is amortized. The fixed percentage is given as the first element in the sinking fund value vector. If the sinking fund value vector is larger than one element, an error will be returned. A number < 0 reduces the notional and a number > 0 increases the notional. The optional start date and end date should be entered as the first two elements in the sinking fund date vector. If the sinking fund date vector is not defined (null or zero elements) the fixed amount amortization will occur for all dates. If the sinking fund date vector has a size > 2, an error will be returned.
SF_FIXED_END	The amortization finishes with a final amortization (amount) at maturity, in addition to the calculated amount. The rest of the amortization amount is divided evenly between all dates. The final amortization amount is given as the first element in the sinking fund value vector. If the sinking fund value vector is larger than one element, an error will be returned. For example, if the amount is equal to zero the amortization is the same for all dates including maturity.
SF_ANNUITY	Annuity. In this case, neither the sinking fund date vector nor the sinking fund value vector is used. The sinking fund is calculated from the restriction that the interest payment and amortization is a constant amount for all payment dates (this is so even if there are stub periods and uneven coupon periods). Annuity is not supported for floating legs.
SF_SAME	The sinking fund schedule is the same as for the other leg in the swap. Hence, only one of the legs can have this setting. If both swap legs have this setting, an error will be returned.
SF_SAME_FXADJ	The same as SF_SAME but the schedule is adjusted for relative exchange rates.
SF_CUSTOM	A custom schedule. If the sinking fund date vector is not defined then the sinking fund value vector must have a size equal to the number of coupon dates on either of the swap legs. If the sinking fund date vector is defined, then amortization will occur on the dates specified and consequently the sinking fund value vector has to be of equal size. Amounts < 0 reduce the notional and amounts > 0 increases the notional.

5.10 Zero rate compounding

Argument	Data type	Location	Description
comp_style	zero_rate_comp_style	ql_swap_fix_zero_def	Determines how the coupon rate will be interpreted. It can either be a rate

			with a certain compounding frequency, a simple rate or a discount rate.
--	--	--	---

zero_rate_comp_style	Description
ZC_ANNUAL	Annual compounding.
ZC_SEMI_ANN	Semi-annual compounding.
ZC_QUARTERLY	Quarterly compounding
ZC_MONTHLY	Monthly compounding.
ZC_SIMPLE	Simple rate (no compounding)
ZC_DISC	Discount rate, $(1 - \text{coupon rate}) * \text{notional} = \text{interest}$. This is a "period-less" style.

6 Member functions and other functions

6.1 *ql_fixed_income_swap* members

Function	Location	Description
accrual_data	ql_fixed_income_swap	returns accrued start date, accrued end date and the accrued period. The cut_off argument is a date that prevents all dates up to this date from being returned.
accrued	ql_fixed_income_swap	returns accrued interest of each leg and the net accrued interest.
cashflow	ql_fixed_income_swap	returns a vector of cashflow objects for each leg. The cashflow object contains information on: <ol style="list-style-type: none"> 1) amount 2) type of cashflow 3) currency 4) date 5) id
clean_value	ql_fixed_income_swap	returns pv excl. accrued of each leg and the net clean value.
convexity <forthcoming>	ql_fixed_income_swap	convexity <forthcoming>
disc_func_leg1	ql_fixed_income_swap	returns the disc_func for leg 1.
disc_func_leg2	ql_fixed_income_swap	returns the disc_func for leg 2.
duration <forthcoming>	ql_fixed_income_swap	duration <forthcoming>
fix_leg	ql_fixed_income_swap	returns a vector of fix legs in the swap. If the swap has no fix legs a null vector will be returned.
fix_zero_leg	ql_fixed_income_swap	returns a vector of fixzero legs in the swap. If the swap has no fixzero legs a null vector will be returned.
float_leg	ql_fixed_income_swap	returns a vector of floating legs in the swap. If the swap has no floating legs a null vector will be returned.
id	ql_fixed_income_swap	returns id of the swap.
start_date	ql_fixed_income_swap	returns start date of the swap.
maturity	ql_fixed_income_swap	returns maturity of the swap
notional	ql_fixed_income_swap	returns notional of each leg of the swap.
ois_leg	ql_fixed_income_swap	returns a vector of ois legs

		in the swap. If the swap has no ois legs a null vector will be returned.
payment_data	ql_fixed_income_swap	returns payment date, coupon rate, cashflow at payment date, present value of the cashflow and the notional. If trade_date_pv is true then the present values will be calculated as of the trade date. The cut_off argument is a date that prevents all payment dates up to this date from being returned.
payment_dates	ql_fixed_income_swap	returns the payment dates of each leg.
present_value	ql_fixed_income_swap	returns pv of each leg and the net pv.
pvpb <forthcoming>	ql_fixed_income_swap	pvpb <forthcoming>
set_disc_func_leg1	ql_fixed_income_swap	sets the disc_func for leg 1.
set_disc_func_leg2	ql_fixed_income_swap	sets the disc_func for leg 2.
set_fixing_leg1	ql_fixed_income_swap	sets fixing data for leg 1.
set_fixing_leg2	ql_fixed_income_swap	sets fixing data for leg 2.
set_settle_date	ql_fixed_income_swap	sets the settlement date.
set_surface_leg1	ql_fixed_income_swap	sets the tenor_surface for leg 1.
set_surface_leg2	ql_fixed_income_swap	sets the tenor_surface for leg 2.
set_trade_date	ql_fixed_income_swap	sets the trade date.
settlement_date	ql_fixed_income_swap	returns settlement date of the swap.
solver	ql_fixed_income_swap	given a pv the following values can be solved for: 1) fix rate 2) float spread 3) frontend fee 4) backend fee
swap_leg	ql_fixed_income_swap	returns a vector of swap legs in the swap.
trade_date	ql_fixed_income_swap	returns trade date of the swap.
type	ql_fixed_income_swap	returns the type of the swap.
unadj_dates	ql_fixed_income_swap	returns the unadjusted dates of each leg.
upfront_carry	ql_fixed_income_swap	This is the difference between what is received and what paid over the investment horizon. Carry can be determined by today's market prices.
upfront_rolldown	ql_fixed_income_swap	This is the difference between what is received and what paid over the investment horizon. The roll-down of a swap makes an assumption that

		the shape of the curve at the horizon is the same as today. Note, possible fixings between today and the horizon date will also assume an unchanged curve.
--	--	--

6.2 ql_swap_leg members

Function	Location	Description
disc_func	ql_swap_leg	returns the disc_func of a swap leg.
id	ql_swap_leg	returns the id of a swap leg.
present value	ql_swap_leg	returns pv of a swap leg.
set_disc_func	ql_swap_leg	sets the disc_func of a swap leg.
type	ql_swap_leg	returns the type of the swap leg.

6.3 ql_float_leg members

Function	Location	Description
accrual_data	ql_float_leg	returns accrued start date, accrued end date and the accrued period. The cut_off argument is a date that prevents all dates up to this date from being returned.
accrued	ql_float_leg	returns accrued interest of a float leg.
base_rates	ql_float_leg	returns the rate excluding any spread, convexity and formula adjustments but includes any decomposing, compounding and averaging effects.
cashflow	ql_floatleg	returns a vector of cashflow objects for a float leg. The cashflow object contains information on: 6) amount 7) type of cashflow 8) currency 9) date 10) id
cashflow	ql_float_leg	returns a vector of cashflow objects for a float leg. The cashflow object contains information on:

		<p>11) amount 12) type of cashflow 13) currency 14) date 15) id</p>
clean_value	ql_float_leg	returns pv excl. accrued of a float leg. If trade_date_pv is true then the present values will be calculated as of the trade date.
current_coupon_rate	ql_float_leg	returns the coupon rate of a float leg for the current period.
current_fixing_rate	ql_float_leg	returns the fixing rate of a float leg for the current period.
current_spread	ql_float_leg	returns the spread of a float leg for the current period. Normally this is the same as spread but in case decomposing is used in stubperiods it can be different.
disc_func	ql_float_leg	returns the disc_func of a float leg.
dyn_cast_float_leg	ql_float_leg	casts a swap_leg to a float leg.
fixing_data	ql_float_leg	returns reset date, fixing rate and a flag indicating whether the rate is fixed or calculated from the curve. The cut_off argument is a date that prevents all reset dates up to this date from being returned. If only_fixed is true, only the rates that are fixed will be returned.
fwd_curve_disc_func	ql_float_leg	returns the disc_func for the tenor. Only applicable if the tenor surface is populated with disc func.
fwd_curve_fwd_func	ql_float_leg	returns the fwd_func for the tenor. Only applicable if the tenor surface is populated with fwd func.
id	ql_float_leg	returns id of a float leg.
index_tenor	ql_float_leg	returns the index tenor of a float leg.
is_reg_first_period	ql_float_leg	returns true if the first period is regular.
is_reg_last_period	ql_float_leg	returns true if the last period is regular.
notional	ql_float_leg	returns notional of a float leg.
payment_data	ql_float_leg	returns payment date, coupon rate, cashflow at payment

		date, present value of the cashflow and the notional. If trade_date_pv is true then the present values will be calculated as of the trade date. The cut_off argument is a date that prevents all payment dates up to this date from being returned.
payment_dates	ql_float_leg	returns the payment dates of a float leg.
present_value	ql_float_leg	returns pv of a float leg. If trade_date_pv is true then the present values will be calculated as of the trade date.
rates	ql_float_leg	returns the rate excluding spread but includes all other adjustments.
reset_dates	ql_float_leg	returns the reset dates of a float leg.
set_disc_func	ql_float_leg	sets the disc_func of a float leg.
set_fixing	ql_float_leg	sets the fixing rates of a float leg.
set_tenor_surface	ql_float_leg	sets the tenor surface of a float leg.
spread	ql_float_leg	returns the spread of a float leg.
tenor_surface	ql_float_leg	returns the tenor surface of a float leg.
type	ql_float_leg	returns the type.
unadj_dates	ql_float_leg	returns the unadjusted dates of float leg.

6.4 ql_fix_leg members

Function	Location	Description
accrual_data	ql_fixleg	returns accrued start date, accrued end date and the accrued period. The cut_off argument is a date that prevents all dates up to this date from being returned.
accrued	ql_fix_leg	returns accrued interest of a fixed leg.
cashflow	ql_fix_leg	returns a vector of cashflow objects for a fixed leg. The cashflow object contains information on: 16) amount

		17) type of cashflow 18) currency 19) date 20) id
clean_value	ql_fix_leg	returns pv excl. accrued of a fixed leg. If trade_date_pv is true then the present values will be calculated as of the trade date.
coupon_rate	ql_fix_leg	returns the coupon rate of a fixed leg.
current_coupon_rate	ql_fix_leg	returns the coupon rate of a fixed leg for the current period. Normally this is the same as coupon_rate but in case compounding is used in stubperiods it can be different.
disc_func	ql_fix_leg	returns the disc_func of a fixed leg.
dyn_cast_fix_leg	ql_fix_leg	casts a swap_leg to a fixed leg.
id	ql_fix_leg	returns id of a fixed leg.
is_reg_first_period	ql_fix_leg	returns true if the first period is regular.
is_reg_last_period	ql_fix_leg	returns true if the last period is regular.
notional	ql_fix_leg	returns notional of a fixed leg.
payment_data	ql_fix_leg	returns payment date, coupon rate, cashflow at payment date, present value of the cashflow and the notional. If trade_date_pv is true then the present values will be calculated as of the trade date. The cut_off argument is a date that prevents all payment dates up to this date from being returned.
payment_dates	ql_fix_leg	returns the payment dates of a fixed leg.
present_value	ql_fix_leg	returns pv of a fixed leg. If trade_date_pv is true then the present values will be calculated as of the trade date.
pvpb_cpn_rate	ql_fix_leg	returns the value of lbp change of the fixed coupon rate.
set_disc_func	ql_fix_leg	sets the disc_func of a fixed leg.
type	ql_fix_leg	returns the type.
unadj_dates	ql_fix_leg	returns the unadjusted dates a fixed leg.

6.5 ql_ois_leg members

Function	Location	Description
accrual_data	ql_ois_leg	returns accrued start date, accrued end date and the accrued period. The cut_off argument is a date that prevents all dates up to this date from being returned.
accrued	ql_ois_leg	returns accrued interest of an ois leg.
base_rates	ql_ois_leg	returns the rate excluding any spread, convexity and formula adjustments but includes any decomposing, compounding and averaging effects.
cashflow	ql_ois_leg	returns a vector of cashflow objects for an ois leg. The cashflow object contains information on: 21) amount 22) type of cashflow 23) currency 24) date 25) id
clean_value	ql_ois_leg	returns pv excl. accrued of an ois leg. If trade_date_pv is true then the present values will be calculated as of the trade date.
disc_func	ql_ois_leg	returns the disc_func of an ois leg.
dyn_cast_ois_leg	ql_ois_leg	casts a swap_leg to an ois leg.
fixing_data	ql_ois_leg	returns reset date, fixing rate and a flag indicating whether the rate is fixed or calculated from the curve. The cut_off argument is a date that prevents all reset dates up to this date from being returned. If only_fixed is true, only the rates that are fixed will be returned.
id	ql_ois_leg	returns id of an ois leg.
index_tenor	ql_ois_leg	returns the index tenor of an ois leg.
is_reg_first_period	ql_ois_leg	returns true if the first period is regular.
is_reg_last_period	ql_ois_leg	returns true if the last

		period is regular.
notional	ql_ois_leg	returns notional of an ois leg.
payment_data	ql_ois_leg	returns payment date, coupon rate, cashflow at payment date, present value of the cashflow and the notional. If trade_date_pv is true then the present values will be calculated as of the trade date. The cut_off argument is a date that prevents all payment dates up to this date from being returned.
payment_dates	ql_ois_leg	returns the payment dates of an ois leg.
present_value	ql_ois_leg	returns pv of an ois leg. If trade_date_pv is true then the present values will be calculated as of the trade date.
rates	ql_ois_leg	returns the rate excluding spread but includes all other adjustments.
reset_dates	ql_ois_leg	returns the reset dates of an ois leg.
set_disc_func	ql_ois_leg	sets the disc_func of an ois leg.
set_fixing	ql_ois_leg	sets the fixing rates of an ois leg.
set_tenor_surface	ql_ois_leg	sets the tenor surface of an ois leg.
tenor_surface	ql_ois_leg	returns the tenor surface of an ois leg.
type	ql_ ois _leg	returns the type.
unadj_dates	ql_ois_leg	returns the unadjusted dates an ois leg.

6.6 ql_fix_zero_leg members

Function	Location	Description
accrual_data	ql_fix_zero_leg	returns accrued start date, accrued end date and the accrued period. The cut_off argument is a date that prevents all dates up to this date from being returned.
accrued	ql_fix_zero_leg	returns accrued interest of a fixzero leg.
cashflow	ql_fix_zero_leg	returns a vector of cashflow objects for a fixzero leg.

		The cashflow object contains information on: 1) amount 2) type of cashflow 3) currency 4) date 5) id
disc_func	ql_fix_zero_leg	returns the disc_func of a fixzero leg.
dyn_cast_fix_zero_leg	ql_fix_zero_leg	casts a swap_leg to an fixzero leg.
id	ql_fix_zero_leg	returns id of a fixzero leg.
notional	ql_fix_zero_leg	returns notional of a fixzero leg.
payment_data	ql_fix_zero_leg	returns payment date, coupon rate, cashflow at payment date, present value of the cashflow and the notional. If trade_date_pv is true then the present values will be calculated as of the trade date. The cut_off argument is a date that prevents all payment dates up to this date from being returned.
payment_dates	ql_fix_zero_leg	returns the payment dates of a fixzero leg.
present_value	ql_fix_zero_leg	returns pv of a fixzero leg. If trade_date_pv is true then the present values will be calculated as of the trade date.
set_disc_func	ql_fix_zero_leg	sets the disc_func of a fixzero leg.
unadj_dates	ql_fix_zero_leg	returns the unadjusted dates a fixzero leg.

6.7 other functions

Function		Description
par_fix_rate		returns the par fix rate.
par_float_spread		returns the par float spread.
par_fix_spread		returns the coupon rate spread between two identical fixed legs, implied from two float legs. This spread is commonly quoted for single currency tenor basis swaps.