6 Indices: Revised

In this chapter, we discuss a very general concept in F3: indices.

We will introduce the concept of indices with examples to develop your intuition about working with indices. We will also introduce F3's powerful tool of constructing and manipulating indices, the F3 Index Expression Language. Finally, we will explain how you can extract the fixing date from a referenced index.

6.1 Introducing Indices

An index in F3 is a random variable, whose value is subject to variations. The possible values that an index can take on represent the possible outcomes of some yet-to-occur events. More specifically, an index in F3 is an object that specifies a random amount.

A useful working definition of an index is "any observable that is sufficiently well-defined to determine an amount, when required, without ambiguity". In F3, the task of defining a quantity is accomplished with the use of an idnex. .

Here are some examples of indices:

- Constant number: An index that only takes a constant value, regardless of any event that occurs. For example, UnitConstant is an index whose observable value is always one. ZeroConstant is an index whose observable value is always zero.
- Equity index: An index that represents the price of an equity asset. In an equity trade, the index is one or more underlying stocks.
- Foreign exchange Index: An index that represents a foreign exchange rate in a multi-currency trade. In a multi-currency trade using GBPUSD, the index is GBPUSD.
- Inflation index: An index that represents the inflation level for a given currency.
- Interest rate index: An index that represents the floating interest rate payable or receivable in a product. In a floating leg of a vanilla interest rate swap with 3-month payment frequency, the index is often LiborUSD3m.
- Derivative index: An index that represents a stochastic amount in a derivative, such as an index that describes an option payoff:. The following are two examples:
    - In the case of a European option, the index describes the payoff of such an option, which is given by

$$payoff = \max\left(\eta\left(S_T - K\right), 0\right),$$

where

- S_T denotes the spot prices at maturity
- K denotes the strike price
- $\eta = \pm 1$ indicates a call and a put option, respectively

- In the case of an Asian option, the index describes the payoff of such an option, which is given by

$$payoff = \max\left(\eta\left(S_{ave} - K\right), 0\right),$$

where

- S_{ave} denotes the average price of the underlying asset over a pre-determined time period
- K denotes the strike price
- $\eta = \pm 1$ indicates a call and a put option, respectively

In each one of the above examples, the index describes an amount in finance. Sometimes, an index is a constant number, such as in the case of a fixed leg of a swap. However, frequently, an index represents a stochastic quantity, such as LiborUSD3m. In such case, the amount is only "fixed" on the fixing date (or reference point). Thus, if an index describes a stochastic quantity before it is fixed, it must not only provide the definition of the amount, but also the prescription of how to quantify this amount precisely.

6.2 Finding the Value of an Index

In this section, we use **LiborUSD3m** as an example.

6.2.1 Use ValueProduct

To find the value of LiborUSD3m, you use **ImpliedCashflows** *request* of the ValueProduct function.

In the Fundamentals Tutorial chapter, you constructed a vanilla interest rate swap. We will now utilize this product after making a few slight modifications to this swap, so that it now has a trade date of **August 1, 2015**. This change is illustrated in Fig. 6.1.

CreateEmptyModel	
ModelName	EmptyModel
BaseDate	2015-08-01
ValuationDate	Default
	EmptyModel

Fig. 6.1. Call to CreateEmptyModel

AddImpliedRateCurveToModel	
ModelName	ImpliedModel
BaseModel	DiscountModel
Index	LiborUSD3m
CollateralAgreement	ZeroCollateral
	ImpliedModel

Fig. 6.2. Call to AddImpliedRateCurveToModel

We also modified the fixed leg so that it has a coupon rate of 5.25%, and a maturity of 1 year, using the market convention of **SwapUSDSemi**. See Fig. 6.3.

CreateSingleCurrencyFixedLeg	
ProductName	FixedLeg1
RollSchedule	2015-08-05
FixedCoupon	5.25%
Notional	10 mio
Currency	USD
PayRec	Receive
Index	UnitConstant
	FixedLeg1

Fig. 6.3. Call to CreateSingleCurrencyFixedLeg: FixedLeg1

The floating leg is now modified so that it uses the floating rate index of **LiborUSD3m**, and the market convention of **SwapUSD3m**, which is done in Fig. 6.4

.

CreateSingleCurrencyFloatingLeg			
ProductName	FloatingLeg1		
RollSchedule	2015-08-05	1y	SwapUSD3m
FloatingRateIndex	LiborUSD3m		
Margin	0		
Notional	10 mio		
Currency	USD		
PayRec	Pay		
Index	UnitConstant		
	FloatingLeg1		

Fig. 6.4. Call to CreateSingleCurrencyFloatingLeg: FloatingLeg1

Construct a swap with **FixedLeg1** and **FloatingLeg1**, which is done in Fig. 6.5.

CreateGenericSwap	
ProductName	VanillaSwap1
CouponLeg	FixedLeg1
OffsettingLeg	FloatingLeg1
ExtraProducts	
	VanillaSwap1

Fig. 6.5. Call to CreateGenericSwap: VanillaSwap1

To extract the value of the index, call the function ValueProduct, using the **ImpliedCashflows** *request*. This is illustrated in Fig. 6.6.

ValueProduct	
Model	ImpliedModel
Product	VanillaSwap1
ValuationMethod	Default
Requests	ImpliedCashflowsLabels
	ImpliedCashflows

PaymentCurrency	PaymentDate	PayReceiveBuySell	Notional	AccrualFraction	ImpliedAmount	IsFixedAmount	ImpliedCashflow
USD	5-Feb-16	Receive-Buy	10000000	0.511111111	0.0525	TRUE	268333.3333
USD	5-Aug-16	Receive-Buy	10000000	0.505555556	0.0525	TRUE	265416.6667
USD	5-Nov-15	Pay-Sell	10000000	0.255555556	0.048418918	FALSE	-123737.2362
USD	5-Feb-16	Pay-Sell	10000000	0.255555556	0.048418918	FALSE	-123737.2362
USD	5-May-16	Pay-Sell	10000000	0.25	0.048412434	FALSE	-121031.0839
USD	5-Aug-16	Pay-Sell	10000000	0.255555556	0.048418918	FALSE	-123737.2362

Fig. 6.6. Call to ValueProduct: VanillaSwap1

> In the output of the function call, the **ImpliedAmount** column denotes the observed value of the **index**.

6.2.2 Use ValueIndex

In some cases, you can use the debugging tool, ValueIndex, to extract the value of an index. We show how you can do this with LiborUSD3m in Fig. 6.7.

ValueIndex				
Model	ImpliedModel			
Index	LiborUSD3m			
RefSpec	05-Nov-15	0.255556	05-Aug-15	05-Nov-2015
ValuationMethod	Default			
ReturnRisk	FALSE			
SortLexically	TRUE			
EvaluationPoint				
	0.04841891850			

Fig. 6.7. Call to ValueIndex: LiborUSD3m

The arguments of ValueIndex are:

- *Model*: Model to use for the valuation

- *Index*: We use **LiborUSD3m** in this example
- *RefSpec*: Reference point specification. This input could be either a date, or a roll, although you should always supply a roll for the input. For more information on reference point and referencing an index, see Finer Details of Indices: Reference Point, Referenced Index and Referencer.
- *ValuationMethod*: We use **Default** in this example.
- *ReturnRisk*: Optional. Flag to indicate if risk exposures of the index are returned in addition to index value. Default is FALSE.
- *SortLexically*: Optional. Flag to indicate if the risk report is ordered lexically. Default is TRUE.
- *EvaluationPoint*: Optional. Index evaluation point.

> Often, the ValueIndex function computes the expected value of an index in the risk-neutral pricing measure which ValueProduct uses. However, in some cases, it uses a different measure from the risk-neutral pricing measure. Therefore, you should only use ValueIndex as a debugging tool. You should avoid comparing results of the values of different indices with ValueIndex.

> When using ValueIndex as a debugging tool, you should always supply a roll as an input to *RefSpec*. This is because when a date is supplied, ValueIndex treats the supplied date as the fixing date, and uses that date as the observation point for the index. This sometimes gives unrealistic results. For example, an inflation index is usually observed either at the end or the beginning of the month. If a date is supplied, ValueIndex then treats the date as the fixing date. To correctly use ValueIndex, you should always supply a roll.

6.3 Creating New Indices

In this section, we introduce the types of built-in indices, and the types of indicies that you can construct in F3, so that you can solidify your intuition about working with indicies.

Before you construct new indices, refer to Index repository in the F3 Reference Manual to view all built-in indices. The most basic built-in indices are UnitConstant and ZeroConstant, which are indices that takes on the values of one and zero, respectively, regardless of any event that occurs. The pre-built LIBOR indices in F3 are examples of indices that are more complex.

Although there are some prebuilt indices in F3, you are likely to construct new indices to suit your needs. Index constructors are functions in F3 that are used to create new indices. These constructors can also incorporate the F3 Index Expression Language (IEL).

We discuss each index type in the following sections.

6.3.1 Built-in Indicies

In the Index repository, you find all the prebuilt indicies such as constant indices, LIBOR-type indices, cross-currency indices, and compound indices, among others. The pre-built indices can be summarized as follows:

- Constant index:

  There are two indices of this type: UnitConstant index, and ZeroConstant index.

  UnitConstant is an index whose observable value is always one.

  ZeroConstant is an index whose observable value is always zero.

- Fixed Coupon index:

  This is an index whose observable specifies the fixed coupon of a trade.

- Overnight indices:

  Overnight rate indices for various currencies.

  There are built-in overnight rate indicies for the following currencies:

  - GBP: Pound Sterling
  - USD: US Dollar
  - EUR: Euro
  - JPY: Japanese Yen
  - CHF: Swiss Franc
  - CAD: Canadian Dollar
  - AUD: Australia Dollar
  - BRL: Brazillian Real
  - RUB: Russian Ruble

- Compounded indices:

  Indices of this type are compounded rates based on the overnight indices.

- LIBOR indices:

  Indices of this type have the form **"Libor"** + **"Currency Name"** + **"Nm"**, where N is the rate tenor, and takes an integer value from 1 to 12. An example is LIborUSD3m.

  There are built-in LIBOR indices for the following currencies:

  - GBP: Pound Sterling

    - USD: US Dollar
    - JPY: Japanese Yen
    - CHF: Swiss Franc
    - CAD: Canadian Dollar
    - AUD: Australia Dollar
    - DKK: Danish Kroner
    - SEK: Swedish Krona
    - NZD: New Zealand Dollar
- EURIBOR indices:

  Indices of this type have the form **"Euribor"** + **"Nm"**, where N is the rate tenor, and takes an integer value from 1 to 12. An example is Euribor3m.

- Bank bill swap indices:

  Indices of this type are for the Australian financial market.

- Bank bill benchmark rate indices:

  Indices of this type are for the New Zealand financial market.

- Cross-currency index:

  Indices of this type have the form **"A"** + **"B"**, where A is the asset currency, and B is the numeraire currency. An example is EURUSD.

- Canadian Dealer Offered Rate indices:

  Indices of this type have tenors of 1m, 2m, 3m, 6m, and 12m.

- Russian MosPrime Rate indices:

  Indices of this type have tenors of 1w, 2w, 1m, 2m, 3m, and 6m.

- Johannesburg interbank average rate indices:

  Indices of this type have tenors of 1m, 3m, 6m, 9m, 12m.

6.3.2 Fundamental Indicies

A fundamental index can best be understood as an index that does not require an underlying index in its construction. Equity indices, inflation indices, interest rate indices, commodity

indices, foreign exchange indices, credit indices, futures indices, swap futures indices are all fundamental indices.

6.3.2.1 Create an Equity Index

You can create an equity index in two different ways. The more basic way is to use CreateSimpleEquityAssetIndex, which is a convenience function that merge several F3 functions into one function at the cost of reduced flexibility. This convenience function constructs a new index that represents the cash settlement price of an equity asset.

The arguments of CreateSimpleEquityAssetIndex are:

- *IndexName*: Name to use for the new index, **FincadStockPriceIndex**
- *Currency*: Trading currency of the equity, **USD**
- *MarketConventions*: Market conventions to be used for settlement, **NewYorkDaily**

Alternatively, you can first create the entity with CreateEntity function, then create the equity entity associated with the asset with CreateEquityEntity function. Next, you create the equity asset index with CreateEquityAssetIndex function.

The arguments of CreateEquityAssetIndex are:

- *IndexName*
- *EquityEntity*: Equity entity associated with the underlying equity asset
- *MarketConventions*: Market convention to be used for settlement, **NewYorkDaily**
- *IndexReferencer*: **UnmodifiedPaymentDate**. This index referencer uses the payment date of a roll as the reference specification, and no specific time of the day.

Fig. 6.8 summarizes the above two different methods. On the left, the diagram shows the workflow for the convenience function CreateSimpleEquityAssetIndex. On the right, the workflow using is shown, which gives you maximum flexibility.

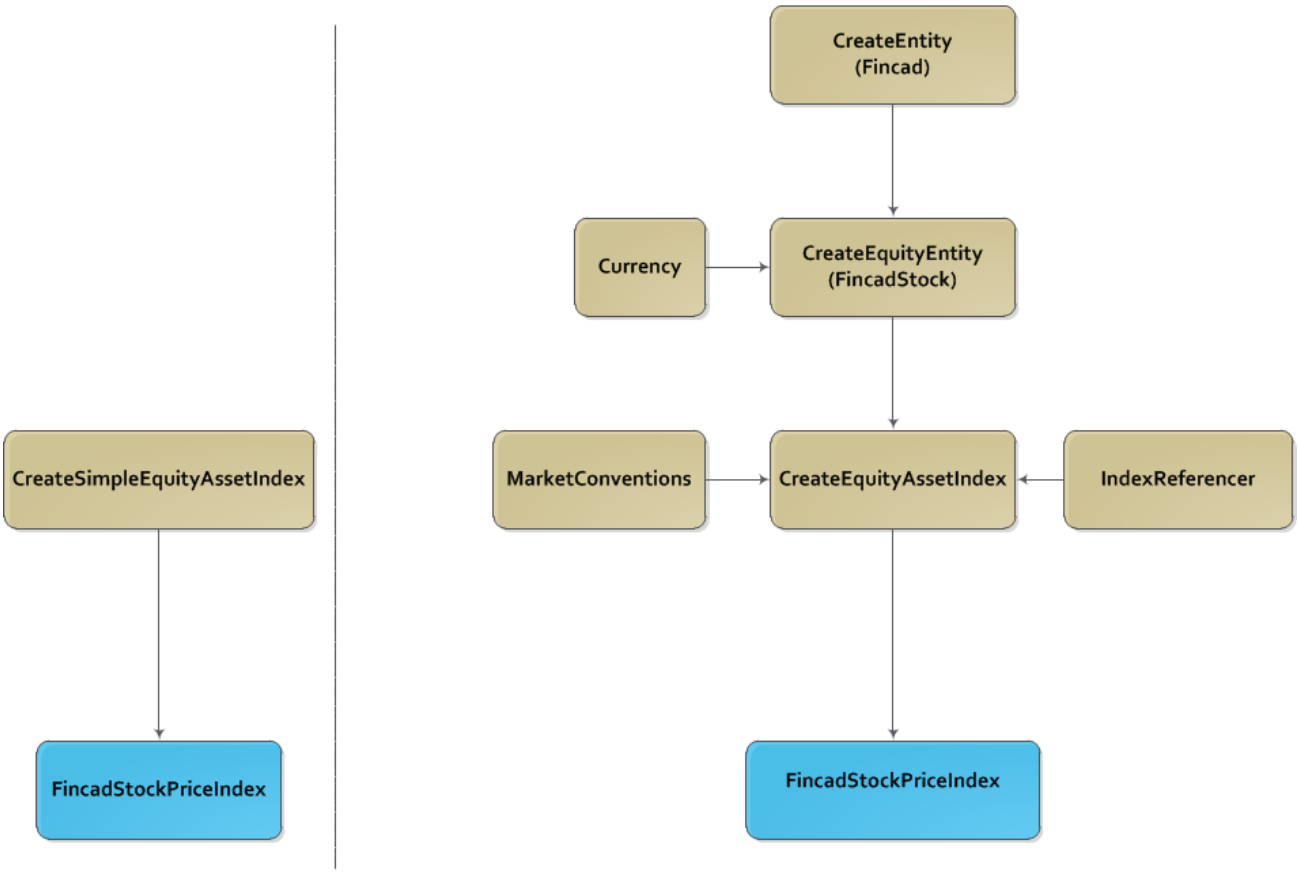

Fig. 6.8. Creating an Equity Index With Two Methods: On the left, an equity index is created with the convenience function CreateSimpleEquityAssetIndex. On the right, an equity index is created without using the convenience function.

> When you use convenience functions such as CreateSimpleEquityAssetIndex, it accomplishes all of the above in one single step. However, you lose the flexibility to create several different equities based on the same entity. To have full flexibility, you must use the alternate way of constructing an equity index to take advantage of the flexibility that F3 affords you.

6.3.2.2 Create an Inflation Index

To create an inflation index, see Fig. 6.9.

CreateRPIInflationIndex	
IndexName	InflationIndex
Currency	USD
SettlementConventions	CashUSD
Referencer	MonthEndEndDate
OptionalCurveReferenceName	
	InflationIndex

Fig. 6.9. Call to CreateRPIInflationIndex

The arguments of CreateRPIInflationIndex are:

- *IndexName*: Name to use for the new index
- *Currency*: Currency underlying the index
- *MarketConventions*: Market conventions used for transaction settlement
- *Referencer*: Optional. Name of the Index referencer. Default: **MonthEndEndDate**.
- *OptionalCurveReferenceName*: Optional. Name for the curve associated with the index.

6.3.2.3 Create an Interest Rate Index

In addition to the built-in interest rate indicies, you can also construct new rate indices, such as one for a currency not found in the built-in Index repository.

To construct a new LIBOR index, use the function CreateLiborIndex. To create a new overnight rate index, you use either the convenience function FormBasicOvernightRateIndex, or CreateOvernightRateIndex.

The arguments of CreateLiborIndex are:

- *IndexName*: Name to use for the new index
- *Currency*: Underlying currency for the LIBOR rate
- *MarketConventions*: Market conventions to use for the LIBOR rate
- *FixingReferenccer*: Optional. Index referencer.
- *OptionalFixingTableName*: Optional. Name for the fixings table associated with the index
- *OptionalCurveReferenceName*: Optional. Name for the curve associated with the index

The arguments for FormBasicOvernightRateIndex are:

- *IndexName*: Name to use for the new index
- *Currency*: Underlying currency for the overnight rate
- *OvernightRateSettlementDelay*: Settlement delay for the overnight rate
- *OvernightRateDayCountType*: Day count convention for daily compounding
- *PaymentHolidays*: Holiday convention used to specify valid business days
- *FixingHolidays*: Holiday convention used to specify valid fixing days

> To create a compounded rate index based on the overnight rate, use CreateCompoundingRateIndex. This compounded index is an example of a derived index, which we will disucss in Derived Indicies.

6.3.2.4 Create a Foreign Exchange Index

6.3.2.5 Create a Commodity Index

6.3.2.6 Create a Credit Index

6.3.2.7 Create a Futures Index

6.3.2.8 Create a Swap Futures Index

6.3.3 Derived Indicies

6.3.3.1 Create a European Option Index

A European call option gives the holder the right to buy the underlying asset for a certain price on the expiration date. A European put option gives the holder the right to sell the underlying asset for a certain price on the expiration date. The price in the contract is known as the exercise price, or the strike price.

In this section, suppose we want to construct a European-style option, written on the underlying asset of Fincad stocks. This European option can be represented by the following term sheet:

Term Sheet

Start Date	July 17, 2015
Spot Price	100
Currency	USD
Market Convention	NewYorkDaily
Maturity	2y
Strike Price	100
Underlying	FINCAD Stock
Notional	1
Option Type	Call
Position	Short

In this section, we will construct a European call option by using the function CreateSingleCashflowProduct. This function consumes an *index* input, which can either be an index object constructed by CreateEuropeanOptionIndex, or an index described by F3's Index Expression Language. See F3 Index Expression Language for more details.

The index of a European option describes the payoff of such an option, and is given by

$$payoff = \max\left(\eta\left(S_T - K\right), 0\right),$$

where

- S_T denotes the spot prices at maturity
- K denotes the strike price
- $\eta = \pm 1$ indicates a call and a put option, respectively

To construct a European option index, see Fig. 6.10.

CreateEuropeanOptionIndex	
IndexName	EuroOptionIndex
UnderlyingIndex	FincadStockPriceIndex
Strike	100
Payoff	Call
	EuroOptionIndex

Fig. 6.10. Call to CreateEuropeanOptionIndex

6.3.3.2 Create a Summation Index

An Asian option is a special type of European option contract. The payoff of an Asian option is determined by the average underlying price over some pre-determined period of time. This is different from a standard European option, where the payoff of the option contract depends on the spot price of the underlying instrument at maturity. Thus, an Asian option is often considered a form of exotic option.

There are many types of Asian options. Here, we focus on the average price call and the average price put Asian option.

The payoff of an average price Asian option is given by

$$payoff = \max\left(\eta\left(S_{ave} - K\right), 0\right),$$

where

- S_{ave} denotes the average price of the underlying asset over a pre-determined time period
- K denotes the strike price
- $\eta = \pm 1$ indicates a call and a put option, respectively

When constructing the product, the biggest difference between an Asian option and a European option is that you use a different underlying index.

Consider an Asian option with the following term sheet:

Term Sheet

Start Date	July 17, 2014
Notional	100
Underlying	FINCAD stock

Market Convention	NewYorkDaily
Maturity	6m
Strike Price	90
Option Type	Put
Position	Long
Observation Dates for Calculating Average Stock Price	1m, 2m, 3m 4m 5m and 6m

Before we can create an Asian option, we first need to create an index that computes the arithmetic mean of the underlying index. You can do this by hand with CreateSummationIndex, or you can do this using the convenience function CreateAsianOptionIndex.

To create the index describing the arithmetic mean by hand, perform the following steps:

1. Call the function CreateIndexWithDifferentReferencer. This is illustrated in Fig. 6.11.

	CreateIndexWithDifferentReferencer				
Maturity	*IndexName*	*UnderlyingIndex*	*Referencer*	*NewIndex*	*Weights*
1m	=FincadStockPriceIndex&":"&TEXT(2014-08-18,"dd-mmm-yyyy")	FincadStockPriceIndex	=MaturityDate("2014-07-17", "1m", "NewYorkDaily")	**FincadStockPriceIndex:2014-08-18**	0.16666667
2m	=FincadStockPriceIndex&":"&TEXT(2014-09-17,"dd-mmm-yyyy")	FincadStockPriceIndex	=MaturityDate("2014-07-17", "2m", "NewYorkDaily")	**FincadStockPriceIndex:2014-09-17**	0.16666667
3m	=FincadStockPriceIndex&":"&TEXT(2014-10-17,"dd-mmm-yyyy")	FincadStockPriceIndex	=MaturityDate("2014-10-17", "3m", "NewYorkDaily")	**FincadStockPriceIndex:2014-10-17**	0.16666667
4m	=FincadStockPriceIndex&":"&TEXT(2014-11-17,"dd-mmm-yyyy")	FincadStockPriceIndex	=MaturityDate("2014-11-17", "4m", "NewYorkDaily")	**FincadStockPriceIndex:2014-11-17**	0.16666667
5m	=FincadStockPriceIndex&":"&TEXT(2014-12-17,"dd-mmm-yyyy")	FincadStockPriceIndex	=MaturityDate("2014-12-17", "5m", "NewYorkDaily")	**FincadStockPriceIndex:2014-12-17**	0.16666667
6m	=FincadStockPriceIndex&":"&TEXT(2015-01-20,"dd-mmm-yyyy")	FincadStockPriceIndex	=MaturityDate("2015-01-20", "6m", "NewYorkDaily")	**FincadStockPriceIndex:2015-01-20**	0.16666667

Fig. 6.11. Call to CreateIndexWithDifferentReferencer: Index to Describe Arithmetic Mean

CreateIndexWithDifferentReferencer takes 3 arguments:

- *IndexName*: Name to use for the index
- *UnderlyingIndex*: Underlying index to use
- *Referencer*: Index referencer mapping to use. Here we use the maturity dates calculated using the maturity descriptor (such as **1m** and the market convention of **NewYorkDaily**. For more information on index referencer, see Concept of Index Referencer.

2. Call the function CreateSummationIndex, as in Fig. 6.12.

CreateSummationIndex						
IndexName	AsianIndex					
ConstituentIndices	FincadStockPriceIndex:2014-08-18	FincadStockPriceIndex:2014-09-17	FincadStockPriceIndex:2014-10-17	FincadStockPriceIndex:2014-11-17	FincadStockPriceIndex:2014-12-17	FincadStockPriceIndex:2015-01-20
IndexWeights	0.16666667	0.16666667	0.16666667	0.16666667	0.16666667	0.16666667
	AsianIndex					

Fig. 6.12. Call to CreateSummationIndex

This function constructs a new index

$$I = \prod_{t=1}^{6} w_i I_i,$$

where $I_i$denotes the constituent indices, and $w_i$denotes the associated weights.

Now that you have created AsianIndex, it can later be used with IEL to compactly describe the payout of an Asian option.

To create an Asian option index using the convenience function CreateAsianOptionIndex, see Fig. 6.23.

CreateAsianOptionIndex	
IndexName	AsianIndex
UnderlyingIndex	FincadStockPriceIndex
ObservationScheduleGenerator	SwapUSD1m
Strike	90
Payoff	Put
	AsianIndex1

Fig. 6.23. Call to CreateAsianOptionIndex

6.3.3.3 Create an Indicator Index and a Product Index

We now introduce the index constructors CreateLessThanIndex using an example of a European barrier option.

A barrier option is an option contract where the payoff depends on whether the underlying asset's price reaches a pre-determined level (the barrier) over the term of the option contract.

There are four basic types of barrier options:

Type	Description
Up-and-out	The spot price starts below the barrier level. The option is knocked out, or ceases to exist, if the spot price moves above the barrier level.

Down-and-out	The spot price starts above the barrier level. The option is knocked out if the spot price moves below the barrier level.
Up-and-in	The spot price starts below the barrier level. The option is knocked in, or comes into existence, if the spot price moves above the barrier level.
Down-and-in	The spot price starts above the barrier level. The option is knocked in if the spot price moves below the barrier level.

For example, suppose we have a European barrier call option, with a spot price of 85, and a knock-out barrier of 100. This option behaves similarly to a European option, except that when the spot price moves above the barrier level of 100, the option is knocked out of existence. The option will not be reactivated if the spot price subsequently moves below the barrier level.

Barrier options are path-dependent derivatives. A path-dependent derivative (or history-dependent derivative) is a derivative where the payoff depends on the path followed by the price of the underlying asset, not just its final value. An Asian option is also an example of path-dependent derivatives. This is because the payoff from an Asian option depends on the average price of the underlying asset. See Asian Option for more details.

In this section, we construct a one-year knock-out put option with a barrier of 100 and strike of 85. For simplicity, suppose that we observe the spot price at 12 intermediate dates. The product has the following term sheet:

Term Sheet

Start Date	July 17, 2014
Currency	USD
Market Convention	NewYorkDaily
Maturity	12m
Notional	1
Underlying	FINCAD Stock
Strike Price	85
Barrier Level	100
Option Type	Put
Position	Long

Potential Knockout Dates	1m, 2m, 3m, ... ,12m

The high-level procedure is as follows:

- Create indices to observe the underlying stock price at intermediate dates
- Create an index to check if the barrier level is breached at intermediate dates
- Create an index to indicate whether the option has been knocked out or not
- Create an European option index to represent the payoff of the option if it is not knocked out
- Create the barrier option index that is the product of the barrier index and the European option index

We illustrate this example in detail in the following sections.

Step 1: Observe the underlying at intermediate dates

Before we can check if the barrier has been breached at the observation time points, we need to construct indices to observe the value of the stock price at these time points.

- Create 12 indices using the function CreateIndexWithDifferentReferencer, each using FincadStockPriceIndex as the underlying index, with the indicated property as shown in Fig. 6.14.

CreateIndexWithDifferentReferencer

Maturity	*IndexName*	*UnderlyingIndex*	*Referencer*	*Product*
1m	=FincadStockPriceIndex&":"&TEXT(2014-08-18,"dd-mmm-yyyy")	FincadStockPriceIndex	=MaturityDate("2014-07-17", "1m", "NewYorkDaily")	**FincadStockPriceIndex:2014-08-18**
2m	=FincadStockPriceIndex&":"&TEXT(2014-09-17,"dd-mmm-yyyy")	FincadStockPriceIndex	=MaturityDate("2014-07-17", "2m", "NewYorkDaily")	**FincadStockPriceIndex:2014-09-17**
3m	=FincadStockPriceIndex&":"&TEXT(2014-10-17,"dd-mmm-yyyy")	FincadStockPriceIndex	=MaturityDate("2014-07-17", "3m", "NewYorkDaily")	**FincadStockPriceIndex:2014-10-17**
4m	=FincadStockPriceIndex&":"&TEXT(2014-11-17,"dd-mmm-yyyy")	FincadStockPriceIndex	=MaturityDate("2014-07-17", "4m", "NewYorkDaily")	**FincadStockPriceIndex:2014-11-17**
5m	=FincadStockPriceIndex&":"&TEXT(2014-12-17,"dd-mmm-yyyy")	FincadStockPriceIndex	=MaturityDate("2014-07-17", "5m", "NewYorkDaily")	**FincadStockPriceIndex:2014-12-17**
6m	=FincadStockPriceIndex&":"&TEXT(2015-01-20,"dd-mmm-yyyy")	FincadStockPriceIndex	=MaturityDate("2014-07-17", "6m", "NewYorkDaily")	**FincadStockPriceIndex:2015-01-20**
6m	=FincadStockPriceIndex&":"&TEXT(2015-02-17,"dd-mmm-yyyy")	FincadStockPriceIndex	=MaturityDate("2014-07-17", "7m", "NewYorkDaily")	**FincadStockPriceIndex:2015-02-17**
6m	=FincadStockPriceIndex&":"&TEXT(2015-03-17,"dd-mmm-yyyy")	FincadStockPriceIndex	=MaturityDate("2014-07-17", "8m", "NewYorkDaily")	**FincadStockPriceIndex:2015-03-17**
6m	=FincadStockPriceIndex&":"&TEXT(2015-04-17,"dd-mmm-yyyy")	FincadStockPriceIndex	=MaturityDate("2014-07-17", "9m", "NewYorkDaily")	**FincadStockPriceIndex:2015-04-17**
6m	=FincadStockPriceIndex&":"&TEXT(2015-05-18,"dd-mmm-yyyy")	FincadStockPriceIndex	=MaturityDate("2014-07-17", "10m", "NewYorkDaily")	**FincadStockPriceIndex:2015-05-18**
6m	=FincadStockPriceIndex&":"&TEXT(2015-06-17,"dd-mmm-yyyy")	FincadStockPriceIndex	=MaturityDate("2014-07-17", "11m", "NewYorkDaily")	**FincadStockPriceIndex:2015-06-17**
6m	=FincadStockPriceIndex&":"&TEXT(2015-07-17,"dd-mmm-yyyy")	FincadStockPriceIndex	=MaturityDate("2014-07-17", "12m", "NewYorkDaily")	**FincadStockPriceIndex:2015-07-17**

Fig. 6.14. Call to CreateIndexWithDifferentReferencer: Observe the Underlying at Intermediate Dates

6.3.3.3.1 CreateLessThanIndex

Step 2: Check if the barrier level has been breached at intermediate dates

Now that we have indices that tell us the value of the underlying at the observation dates, we need to compare those values to the barrier. We use the function CreateLessThanIndex to do this.

CreateLessThanIndex takes three inputs:

- *IndexName*: Name to use for the index
- *FirstIndex*: First index in comparison
- *SecondIndex*: Second index in comparison

CreateLessThanIndex compares the values of two indices, and constructs a new index. The new index is expressed as

$$I = \begin{cases} 1, & \text{if } I_1 \not\geq I_2 \\ 0, & \text{if otherwise,} \end{cases}$$

where I_1 is the first index used in the comparison, and I_2 is the second index in comparison.

We now construct a new index, with the property that it gives a value of 1 at all time points when the barrier level is not breached, and a value of 0 otherwise. This is illustrated in Fig. 6.15.

CreateLessThanIndex		
IndexName	*FirstIndex*	*SecondIndex*
=BarrierIndex&":"&TEXT(2014-08-18,"dd-mmm-yyyy")	FincadStockPriceIndex:2014-08-18	100 * UnitConstant
=BarrierIndex&":"&TEXT(2014-09-17,"dd-mmm-yyyy")	FincadStockPriceIndex:2014-09-17	100 * UnitConstant
=BarrierIndex&":"&TEXT(2014-10-17,"dd-mmm-yyyy")	FincadStockPriceIndex:2014-10-17	100 * UnitConstant
=BarrierIndex&":"&TEXT(2014-11-17,"dd-mmm-yyyy")	FincadStockPriceIndex:2014-11-17	100 * UnitConstant
=BarrierIndex&":"&TEXT(2014-12-17,"dd-mmm-yyyy")	FincadStockPriceIndex:2014-12-17	100 * UnitConstant
=BarrierIndex&":"&TEXT(2015-01-20,"dd-mmm-yyyy")	FincadStockPriceIndex:2015-01-20	100 * UnitConstant
=BarrierIndex&":"&TEXT(2015-02-17,"dd-mmm-yyyy")	FincadStockPriceIndex:2015-02-17	100 * UnitConstant
=BarrierIndex&":"&TEXT(2015-03-17,"dd-mmm-yyyy")	FincadStockPriceIndex:2015-03-17	100 * UnitConstant
=BarrierIndex&":"&TEXT(2015-04-17,"dd-mmm-yyyy")	FincadStockPriceIndex:2015-04-17	100 * UnitConstant
=BarrierIndex&":"&TEXT(2015-05-18,"dd-mmm-yyyy")	FincadStockPriceIndex:2015-05-18	100 * UnitConstant
=BarrierIndex&":"&TEXT(2015-06-17,"dd-mmm-yyyy")	FincadStockPriceIndex:2015-06-17	100 * UnitConstant
=BarrierIndex&":"&TEXT(2015-07-17,"dd-mmm-yyyy")	FincadStockPriceIndex:2015-07-17	100 * UnitConstant

Fig. 6.15. Call to CreateLessThanIndex

What we end up with are 12 indices with a value of either 1 or 0. Each of these indices is itself a product of two indices, FincadStockPriceIndex and the 100*UnitConstant index, at the observation dates.

6.3.3.3.2 CreateProductIndex

Step 3: Create an index to indicate whether the option has been knocked out

In this step, we construct an index that indicates if the option has been knocked out. We do this by using the function CreateProductIndex.

This function constructs a new index

$$I = \prod_i I_i,$$

where I_i denotes the constituent indices.

For each observation date, the CreateLessThanIndex function returns 1 if the barrier has not been breached, and 0 otherwise. Multiplying the above 12 indices using **CreateProductIndex** ensures that if the barrier has been breached in any of the observation times, then it will return 0. This is done in Fig. 6.16.

CreateProductIndex	
IndexName	Survived_or_Knocked_out
ConstituentIndices	BarrierIndex:2014-08-18
	BarrierIndex:2014-09-17
	BarrierIndex:2014-10-17
	BarrierIndex:2014-11-17
	BarrierIndex:2014-12-17
	BarrierIndex:2015-01-20
	BarrierIndex:2015-02-17
	BarrierIndex:2015-03-17
	BarrierIndex:2015-04-17
	BarrierIndex:2015-05-18
	BarrierIndex:2015-06-17
	BarrierIndex:2015-07-17
	Survived_or_Knocked_out

Fig. 6.16. Call to CreateProductIndex

Step 4: Create a European option index

In this step, we use the function CreateEuropeanOptionIndex to create an index to represent the payoff of the option if it has not been knocked out. See Fig. 6.17.

CreateEuropeanOptionIndex	
IndexName	FincadPutOptionIndex
UnderlyingIndex	FincadStockPriceIndex
Strike	85
Payoff	Put
	FincadPutOptionIndex

Fig. 6.17. Call to CreateEuropeanOptionIndex: FincadPutOptionIndex

Step 5: Create a product index that describes the Barrier option payoff

See Fig. 6.18.

CreateProductIndex	
IndexName	BarrierOptionIndex
ConstituentIndices	Survived_or_Knocked_out
	FincadPutOptionIndex
	BarrierOptionIndex

Fig. 6.18. Call to CreateProductIndex: BarrierOptionIndex

At the end of step 5, you have created **BarrierOptionIndex**. This index fully describes the payoff of a European barrier put option, which is given by

$$payoff = \max(K - S_T, 0) \cdot I$$

where the quantity $\max(K - S_T, 0)$is represented by the index FincadPutOptionIndex constructed in step 4, and $I$is represented by the index Survived_or_Knocked_out constructed in step 3, with the property that

$$I = \prod_{t=1}^{12} I_{AB_t},$$

where $I_{AB} = I_A \cdot I_B$, $I_A$is the FincadStockPriceIndex constructed in step 1, and $I_B$is the 100*UnitConstant index constructed in step 2.

6.4 F3 Index Expression Language

This section introduces F3's Index Expression Language, which is a tool to create complex indices by directly applying certain simple operators on pre-built or user-constructed indices.

This means that when you create an index, you can either create it by the index constructors, or you can create it by the Index Expression Language (IEL). An index created by IEL can be used throughout F3 wherever an index is an input to a function.

You have seen many examples of index constructors in the preceding section. We will now introduce the F3 Index Expression Language.

The advantage of the index expression language is that it allows the description of almost any financial payoff explicitly, and cuts down the number of steps required to construct an object.

6.4.1 Operators Used in Index Expression Language

Formally, the F3 Index Expression Language (IEL) is an expression that defines an index as a function of one or more other indices, constants, and operators.

Informally, IEL is F3's method of creating a complex indices through direct manipulation of underlying indices. See Index Argument Type in F3 Reference Manual for a complete list of operators that can be used to specify a new index.

IEL uses the following mathematical operators to describe the product in a similar way to how it would be written on a term sheet:

- **Arithmetic operators**

  IEL uses basic mathematical operators such as **+**, **-**, *****, **/** to describe the payoff of a product intuitively.

  - Example 1: The index describing the payoff of portfolio product of 3 stocks, Stock1, Stock2, and Stock3, is written in IEL as **(Stock1 + Stock2 + Stcok3)/3**, where Stock1, Stock2, and Stock3 are indices which bear the same name as the stocks.
  - Example 2: Further, instead of using an index constructor to create a new index which is 90% of LiborUSD3m, you can use index expression language to compactly describe the new index as **0.9 * LiborUSD3m**.
  - Example 3: Similarly, you can use an expression such as **LiborUSD3m * LiborUSD3m** to build a new index easily.

- **Binary min and max functions**

The max or min functions used in the F3 IEL takes two arguments, each of which must be an index.

Since the max or min function takes only two arguments, when comparing three or more indices, you need to choose use the same function iteratively.

- Example 4: To find the maximum among three indices, A, B, and C, use the expression **max(A, max(B,C))**.

- **Option strategies**

  You can use **call**, **put** and **straddle** to create a new index to describe the option payoff. Call, put, and straddle all take an underlying index as a first argument, and strike as a second argument.

  - Example 5: **Call(A,100)** describes the payoff of a call option written on index **A**, with a strike of 100.

- **Boolean operators**

  You can use <, >, <=, >=, **and**, **or** and **not()**. These result in boolean-values index expressions, which may form the input to **if**, a conditional operator described below.

- **Conditional operator**

  if(*Condition, Left, Right*) evaluates to ***Left*** if ***Condition*** is true, otherwise to ***Right***, where ***Condition*** is a boolean-valued index expression, and ***Left***, ***Right*** are any index expressions.

- **Bind function**

  The **bind** function takes an index and date string as arguments. This function causes the specified index to be referenced, or bound to the date provided. Note that the specified index can be given in the form of IEL.

  - Example 6: You can bind an index to a specific date by using the express **bind(LiborUSD3m * LiborUSD3m, June 18, 2014)**. Note that the index can be expressed using IEL.

- **Parentheses**

  Parentheses **()** are used to indicate precedence in expressions.

- **Scaling operators**

bp or **bps** (basis points)	0.0001
%	0.01
m (milla)	1,000
mm or **mio**>	1,000,000

6.4.2 Input Methods: Using IEL

When constructing indices using IEL, the cell contents must be entered as a string. In Excel, this is done by putting all text in quotation marks and by separating text from links using the **&** symbol.

Mathematical function such as max, min, and option strategies such as call and put must be entered in Excel as strings to avoid confusion with similar Excel functions. These strings can then be used together with other strings, commas, and brackets to describe a trade.

See the following for the input methods for the above Example 1 to 6:

Example: IEL	**Direct input methods**	**Input methods with cell links**
(Stock1 + Stock2 + Stock3)/3	="(Stock1 + Stock2 + Stock3)/3"	="(" & cell &" + "& cell &" + "& cell &")/3"
0.9 * LiborUSD3m	="0.9 * LiborUSD3m"	=" 0.9 * "& cell &" "
LiborUSD3m * LiborUSD3m	="LiborUSD3m * LiborUSD3m" You can also use CreateProductIndex to create this index	=" "& cell &" * "& cell &" "
max(A, max(B,C))	="max("A", max("B", "C"))"	="max("& cell &", max("& cell & ", " & cell &"))"
call(A,100)	="call(A,100)"	="call("& cell &",100)"
bind(LiborUSD3m * LiborUSD3m, June 18, 2014)	="bind(LiborUSD3m * LiborUSD3m, 2014-06-18)"	="bind("& cell &"*"& cell &", 2014-06-18)"

6.4.3 Three ways of Creating a European Option

In this example, we compare and contrast the three different ways of creating a European option.

1. Use IEL

   This is illustrated in Fig. 6.19.

CreateSingleCashflowProduct	
ProductName	EUOption1
RollDates	=MaturityDate("2014-07-17", "2y", "NewYorkDaily")
Index	="call(FincadStockPriceIndex, 100)"
Notional	1
Currency	USD
PayRec	Pay
	EUOption1

Fig. 6.19. Call to CreateSingleCashflowProduct: EUOption1

2. Use CreateSingleCashflowProduct and EuroOptionIndex that you constructed earlier

   To pass the European option index as an argument in CreateSingleCashflowProduct so that you can create a European option, see Fig. 6.20.

CreateSingleCashflowProduct	
ProductName	EUOption2
RollDates	=MaturityDate("2014-07-17", "2y", "NewYorkDaily")
Index	EuroOptionIndex
Notional	1
Currency	USD
PayRec	Pay
	EUOption2

Fig. 6.20. Call to CreateSingleCashflow Product: EUOption2

   CreateSingleCashflowProduct takes six arguments:

   - *ProductName*: Name to use for the created product, **EUoption2**
   - *RollDates*: Dates roll for the cash flow, **=MaturityDate("2014-07-17", "2y", "NewYorkDaily")**.
   - *Index*: Index describing the cash flow payments, **EuroOptionIndex**.
   - *Notional*: Notional amount of the cash flow.
   - *Currency*: Currency of the notional amount
   - *PayRec*: Flag indicating payment or receipt of the cash flow, **Pay**. This indicates you hold the short position, and have to pay out if the option expires in the money.

3. Use the convenience function CreateEuropeanOption

The function call to CreateEuropeanOption is shown in Fig. 6.21.

CreateEuropeanOption	
ProductName	EUOption3
Underlying	FincadStockPriceIndex
Expiry	=MaturityDate("2014-07-17", "2y", "NewYorkDaily")
Notional	1
Strike	100
PayRec	Call
BuySell	Pay
	EUOption3

Fig. 6.21. Call to CreateEuropeanOption: EUOption3

In summary, we constructed EUOption1, EUOption2, and EUOption3. Even though they are created using different approaches, they have the same value. This makes sense because even though CreateEuropeanOption references FincadStockPriceAsset index, whereas CreateSingleCashflowProduct references EuroOptionIndex, these two indices look to the same equity entity object, namely, FincadStock, to identify the underlying asset.

6.4.4 Two Ways of Creating an Asian Option

In this example, we compare and contrast the two different ways of creating an Asian option.

1. Use IEL

   In the Useful Function for Asian Options: CreateSummationIndex section, you have constructed the AsianIndex by using the CreateSummationIndex function. Now that we have introduced IEL, we will complete the example started in that section.

   This is illustrated in Fig. 6.22. The advantage of using IEL is that you can quickly construct an index for an Asian option with minimum amount of steps. We will contrast this method with the other method, which require more intermediate steps to construct an Asian option product.

CreateSingleCashflowProduct	
ProductName	AsianPutOption1
RollDates	=MaturityDate("2014-07-17", "6m", "NewYorkDaily")
Index	="put(AsianIndex, 90)"
Notional	100
Currency	USD
PayRec	Rec
	AsianPutOption1

Fig. 6.22. Call to CreateSingleCashflowProduct: AsianOption1

With one more application of IEL, you can express **AsianIndex** as "(FincadStockPriceIndex:18-Aug-2014 + FincadStockPriceIndex:17-Sep-2014 + FincadStockPriceIndex:17-Oct-2014 + FincadStockPriceIndex:17-Nov-2014 + FincadStockPriceIndex:17-Dec-2014 + FincadStockPriceIndex:20-Dec-2015)/6". This will further cut out the step of creating AsianIndex.

2. Use CreateAsianOptionIndex and CreateSingleCashflowProduct

   This is demonstrated in Fig. 6.23 and Fig. 6.24.

CreateAsianOptionIndex	
IndexName	AsianIndex2
Underlying	FincadStockPriceIndex
ObservationScheduleGenerator	SwapUSD1m
Strike	90
BuySell	Put
	AsianIndex2

Fig. 6.23. Call to CreateAsianOptionIndex

CreateSingleCashflowProduct				
ProductName	AsianPutOption2			
RollDates	2015-01-20	0.094444444	2014-12-17	2015-01-20
Index	AsianIndex2			
Notional	100			
Currency	USD			
PayRec	Rec			
	AsianPutOption2			

Fig. 6.24. Call to CreateSingleCashflowProduct: AsianOption2

> In summary, we constructed AsianPutOption1 and AsianPutOption2. They are constructed using different approaches, but they have the same value.

6.4.5 Three Ways of Creating a Barrier Option

In this example, we compare and contrast the three different ways of creating a Barrier option.

1. Use IEL

   This is demonstrated in Fig. 6.25. Note that using only simple operators, you can quickly construct an index for this option with minimum amount of steps. We will contrast this method with other methods, which require more intermediate steps to construct the Barrier option product.

CreateSingleCashflowProduct	
ProductName	BarrierOption1
RollDates	=MaturityDate("2014-07-17", "12m", "NewYorkDaily")
Index	="if(Survived_or_Knocked_out <1, ZeroConstant, put(FincadStockPriceIndex, 85))"
Notional	1
Currency	USD
PayRec	Rec
	BarrierOption1

Fig. 6.25. Call to CreateSingleCashflowProduct: BarrierOption1

2. Use CreateSingleCashflowProduct with BarrierOptionIndex you constructed earlier

   This is illustrated in Fig. 6.26. With this method, you need to have created all of the indices in Subsubsec. 6.3.3.3 in the first place.

CreateSingleCashflowProduct	
ProductName	BarrierOption2
RollDates	=MaturityDate("2014-07-17", "12m", "NewYorkDaily")
Index	BarrierOptionIndex
Notional	1
Currency	USD
PayRec	Rec
	BarrierOption2

Fig. 6.26. Call to CreateSingleCashflowProduct: BarrierOption2

3. Use the convenience function CreateSingleBarrierEuropeanOption

   This is illustrated in Fig. 6.27.

CreateSingleBarrierEuropeanOption	
ProductName	BarrierOption3
Underlying	FincadStockPriceIndex
Start	2014-07-17
Expiry	=MaturityDate("2014-07-17", "12m", "NewY
Notional	1
BarrierLevel	100
UpOrDown	Up
InOrOut	Out
BarrierObservation	SwapUSD1m
Strike	85
Payoff	Put
Buysell	Rec
	BarrierOption3

Fig. 6.27. Call to CreateSingleBarrierEuropeanOption: BarrierOption3

In summary, we constructed BarrierOption1, BarrierOption2, and BarrierOption3. They are constructed using different approaches, but they have the same value.

6.5 Finer Details of Indices: Reference Point, Referenced Index and Referencer

In order to quantify precisely an amount in a legal contract, the index must contain all necessary information to make an observation of this amount. Sometimes, an index is a constant number (such as UnitConstant or ZeroConstant). However, an index typically represents a stochastic amount, whose value is only "fixed" upon observation.

In F3, the fixing, or observation time, is called the **reference point**. The act of binding an index to a reference point is called **referencing the index**. An observed index is called a **referenced index**, which is also an index-reference point pair. Furthermore, the act of referencing an index is performed by the **IndexReferencer** objects.

Let us consider an example of a floating rate bond using the index of LiborUSD3m, with the coupon payment date of June 30, 2015:

- The referenced point, or fixing point, denoted by t_f, is March 26, 2015. It is the date when LiborUSD3m is observed.
- The referenced index, LiborUSD3m on March 26, 2015, is $LiborUSD3m(t_f)$.
- The referencer, denoted R, is a mapping from the roll to the reference point, namely,

  $$R : rolls \to dates.$$

  It calculates the reference point for a given roll. Then the reference point is given by

  $$t_f = R(roll).$$

Next, we discuss how you can extract the reference point from a referenced index. We will also discuss how you can create a different index using a different mapping (referencer).

6.5.1 Find Out the Fixing Date of an Index

To obtain the fixing date, or the reference point of a referenced index, use ReferencedIndexInfo.

ReferencedIndexInfo takes 4 arguments:

- *Index*: Name of index whose fixing date you want to query
- *RefSpec*: Reference point specification

  This input encodes the information with which a referencer mapping, or R, uses to form a reference point. A valid input must be either a date, or a roll.

  When a date is specified, most referencer mappings will use such a date as the observation point, and treat it as the fixing date.

  As you will see in the subsequent section, you can create an index with a different referencer. Therefore, when a roll is specified, then the referencer can use different time point specified in the roll, such as the payment date, the roll start date or end date, as the fixing date of the index.
- *Requests*
- *EvaluationPoint*: This input is optional.

In the following example, we suppose a floating leg of an interest rate swap (with the floating leg index of LiborUSD3m), start date of December 23, 2014 and maturity of one year.

We first construct the roll schedule for the swap, as shown in Fig. 6.28. For a review of the RollSchedule function, see Use the RollSchedule Function.

RollSchedule				
StartDate	2014-12-23			
Maturity	1y			
MarketConvention	SwapUSD3m			
	23-Mar-15	**0.25**	**23-Dec-14**	**23-Mar-15**
	23-Jun-15	**0.255555556**	**23-Mar-15**	**23-Jun-15**
	23-Sep-15	**0.255555556**	**23-Jun-14**	**23-Sep-15**
	23-Dec-15	**0.252777778**	**23-Sep-14**	**23-Dec-15**

Fig. 6.28. Call to RollSchedule

To extract the reference point of LiborUSD3m with coupon payment on December 23, 2015, see Fig. 6.29.

ReferencedIndexInfo				
Index	LiborUSD3m			
RefSpec	23-Dec-15	0.252777778	23-Sep-15	23-Dec-15
Requests	ReferenceDate			
EvaluationPoint				
	21-Sep-15			

Fig. 6.29. Call to ReferencedIndexInfo: LiborUSD3m

Let's take a step back and make sure we understand what has happened so far. Given a floating leg of interest rate swap with one-year maturity, and a start date of December 23, 2014, using a floating leg index of LiborUSD3m, we want to find out when LiborUSD3m, a stochastic quantity, is "fixed", or "observed", for the coupon payment on December 23, 2015. By invoking the function ReferencedIndexInfo, you can extract the reference point of such an index given a roll that specifies the payment date of June 30, 2015.

> When the LiborUSD3m is passed into this function, the function returns a fixing date that is two business days before the start date of the roll, not two business days before the payment date. However, you can create an index such that the fixing date is two business days before the payment date, or the end date. We will discuss this next.

6.5.2 Concept of Index Referencers

In the Referencing Indices: Reference Point, Referenced Index and Referencer section, you have seen that an index referencer is a mapping from rolls to reference points. It specifies the method that binds an index to the reference point. Keep in mind that an index represents the definition of an entity being observed, and is kept distinct from the act of actually observing the entity's value at a given point in time.

Because an index referencer is a mapping, namely, $R : rolls \rightarrow dates$,with $t_f = R(roll)$, it is a rule for forming a reference point. That is, an index referencer specifies the method used to select the reference point.

Valid input consists of any member of the IndexReferencer repository. Additionally, it can also be one of the following inputs:

- A date.
- The array **[S,M,H]**, where **S** is one of the strings described below, **M** is a market conventions object, and **H** is an optional holiday convention.

The list of valid strings **S** is:

- **PaymentOffset**
- **PaymentOffsetLatest**, which is the same as PaymentOffset
- **PaymentOffsetEarliest**
- **StartOffset**
- **StartOffsetLatest**, which is the same as StartOffset
- **StartOffsetEarliest**
- **EndOffset**
- **EndOffsetLatest**, which is the same as EndOffset
- **EndOffsetEarliest**

All referencer mapping formed by using the array **[S,M,H]** will base its calculation on a date **d** in the roll, where **d** is either the **payment date**, **start date**, or **end date** of the roll. Then the referencer uses the function **TradeDatesBoundingStartDate** to find the date range that, under the settlement delay of M, bounds **d**. For more details on the TradeDatesBoundingStartDate, see Useful Function: TradeDatesBoundingStartDate. The date range is between a and b, where under the market convention **M**, a is the latest day that will give a date before **d**, and b is the earliest day that will give a date after **d**.

When **S** ends with "Earliest", the referencer increments **a** forward one day, until it finds a valid business day within the date rage **a:b** according to the holiday convention **H**.

When **S** ends with "Latest", the referencer increments **b** backward one day, until it finds a valid business day according to **H** within the date rage.

Where no **H** is supplied, then F3 automatically **forms H from M**.

It is possible that the above algorithm fails to find a reference point based on **d**, with the implicit settlement delay of **M**. In this case, the referencer finds the latest valid business day before but closest to **d**.

These referencers do not consider any specific time of day when forming an observation point.

6.5.2.1 Creating Indicies With Different Fixing Rules

In the Useful Function: ReferencedIndexInfo section, you have seen a floating leg of an interest rate swap with one-year maturity, and a start date of December 23, 2014, using index of

LiborUSD3m. In that example, we have extracted the fixing date that is two business days before the **start date** of the roll.

We now introduce an example where you can create a new index that has a fixing date that is two business days before the **payment date**. You can achieve this by using a different referencer mapping. But first, you need to create a different index, which is illustrated in Fig. 6.30.

CreateIndexWithDifferentReferencer		
IndexName	LiborUSD3m:PaymentOffset	
UnderlyingIndex	LiborUSD3m	
Referencer	PaymentOffset	SwapUSD3m
	LiborUSD3m:PaymentOffset	

Fig. 6.30. Call to ReferencedIndexInfo: LiborUSD3m:PaymentOffset

Recall from the preceding section, an valid input for *Referencer* can be of the format **[S, M, H]**, where **H** is optional. The input:

PaymentOffset	SwapUSD3m

satisfies this format. This referencer forms the fixing date based on the **payment date** of the roll, which is in turn based on the market convention SwapuSD3m.

After the index with the new fixing rule is created, you can call the function ReferencedIndexInfo to extract the fixing date. This is demonstrated in Fig. 6.31.

ReferencedIndexInfo				
Index	LiborUSD3m:PaymentOffset			
RefSpec	23-Dec-15	0.252777778	23-Sep-15	23-Dec-
Requests	ReferenceDate			
EvaluationPoint				
	21-Dec-15			

Fig. 6.31. Call to ReferencedIndexInfo: LiborUSD3m:PaymentOffset

CreateIndexWithDifferentReferencer takes 3 arguments:

- *IndexName*: Name to use for the index
- *UnderlyingIndex*: Underlying index to use
- *Referencer*: Index referencer mapping to use

After the function call, you will get the reference date of December 21, 2015, which is two business days before the coupon payment date.

<< PREVIOUS CHAPTER CONTENTS NEXT CHAPTER >>

Intellectual Property