FpML Version 2.0

Working Draft 17 October 2001

This version:

http://www.fpml.org/spec/2001/wd-fpml-2-0-2001-10-17

Latest version:

http://www.fpml.org/spec/fpml-2-0

Previous version:

http://www.fpml.org/spec/2001/wd-fpml-2-0-2001-08-23

Status of this Document:

This is the FpML Version 2.0 Working Draft for review by the public and by FpML.org members and working groups. It is a draft document and may be updated, replaced or obsoleted by other documents at any time. It is inappropriate to use FpML Working Drafts as reference material or to cite them as other than "work in progress". This is work in progress and does not imply endorsement by the FpML.org Consortium.

The Interest Rate Derivatives (IRD) Products Working Group encourage reviewing organizations to provide feedback as early as possible. Comments on this document should be sent via e-mail to fpml-issues@yahoogroups.com. An archive of the comments is available at http://groups.yahoo.com/group/fpml-issues/messages.

Public discussion of FpML takes place on the FpML Discussion List at http://groups.yahoo.com/group/fpml-discuss.

A list of current FpML Recommendations and other technical documents can be found at http://www.fpml.org/spec.

While implementation experience reports are welcomed, the IRD Products Working Group will not allow early implementation to constrain its ability to make changes to this specification prior to final release.

This document has been produced as part of the FpML Version 2.0 activity and is part of the Standards Approval Process. This Activity was initiated by the FpML Board of Directors in October 2000 to extend the FpML Version 1.0 product definitions, which covered interest rate swaps and FRAs, to include further interest rate derivatives products and features.

Working Group Members and Acknowledgements:

This document was produced in the IRD Products Working Group, which comprises of the following members:

- Steven Lord (UBS Warburg), chair
- Guy Gurden (SwapsWire), previous chair
- Andrew Addison (Merrill Lynch)
- Ariane Athalie (BNP Paribas)
- Owen Bugge (Blackbird)
- Marisol Collazo (Mizuho Capital Markets)
- Marie-Paule Dumont (S.W.I.F.T.)
- Alexander Ernst (RiskTrak Financial)
- Tom Fahy (Goldman Sachs)
- Doug Gallager (Reval.com)
- Mark Golding (JPMorgan)
- David Gorans (BNP Paribas)
- Paul Hoskins (Barclays Capital)
- Saleem Huda (Algorithmics)
- Vlad Iordanov (FinTrack Systems)
- Keri Jackson (Cygnifi)

- Sathy Kovvali (Citigroup)
- Philippe Negri (SunGard Trading & Risk Systems)
- Michael North (Reuters)
- Henry Teng (UBS Warburg)
- Ian Thomas(Credit Suisse First Boston)
- Patrick Treanor (Wall Street Systems)
- Olga Urrutia (S.W.I.F.T.)
- James Williams (Deutsche Bank)
- Barry Witkow (Treasury Connect)
- Chuck Witter (Bank of America)

Working Group Guests

- Nicholas Davies (Cygnifi)

Editor

- Steven Lord (UBS Warburg)

TABLE OF CONTENTS

1 INTRODUCTION

The Financial Products Markup Language (FpML) is a protocol enabling e-commerce activities in the field of financial derivatives. The development of the standard, controlled by FpML.org, will ultimately allow the electronic integration of a range of services, from electronic trading and confirmations to portfolio specification for risk analysis. All types of over-the-counter (OTC) derivatives will, over time, be incorporated into the standard, although the current focus of FpML Version 2.0 is interest rate derivatives.

FpML is an *application* of XML, an internet standard language for describing data shared between computer applications.

2 SCOPE

2.1 Scope

The scope of the IRD Products Working Group, with respect to extending the FpML 1.0 product definitions, is to complete definitions for the following new products and features:

- Interest Rate Cap
- Interest Rate Floor
- Interest Rate Swaption (European, Bermudan and American Styles; Cash and Physical Settlement)
- Extendible and Cancelable Interest Rate Swap Provisions
- Mandatory and Optional Early Termination Provisions for Interest Rate Swaps
- FX Resetable Cross-Currency Swap

Current capabilities of the FpML Version 1.0 specification to support Basis Swaps will also be reviewed.

Outside the scope of the Working Group are the following:

- Definition of business processes that might result in the trade content defined here being transmitted between parties. The definition of these processes and resulting messages is expected to be covered by the work of other FpML Working Groups.
- Definition of reference data related to the counterparty such as settlement instructions, location and contact details. It was agreed that this static data did not belong in each instance of an FpML document and would most likely be stored in central or distributed repositories and referenced from within the document. Specification or design of such repositories is also beyond the scope of the Working Group. Since identification of parties is an essential requirement of a trade content definition, the FpML Consortium has decided, to continue in this release, to identify parties using the ISO standard bank identifier code (BIC). S.W.I.F.T. is the designated registration authority for the assignment of BIC codes. Although this is the recommended identification scheme for parties wishing to use FpML for inter-firm communication , the FpML architecture supports the use and identification of alternative coding schemes through the Schemes mechanism.

2.2 Architecture Framework

The Products Working Group has developed FpML 2.0 within the FpML Architecture Version 1.0 framework defined by the Architecture Working Group. Their recommendations covered:

- XML tools for editing and parsing
- XML namespace usage within FpML
- FpML versioning methodology
- FpML content model - a new style for representing the FpML Document Type Definition (DTD)
- FpML referencing methodology, including guidelines for referencing coding schemes.

2.3 DTD Structure

FpML 1.0 Recommendation and FpML 2.0 Working Draft have been publised as a single DTD. With the addition of other asset classes (FX and Equities) it is intended to split the DTD into parts:

1. A Shared components DTD
2. A Asset class specific DTD
3. A main dtd which links the other dtds into the FpML standard.

These changes will be put into place for FpML 3.0. This release will also incorporate any recommendations from the FpML Architecture Working Group about moving to Schema.

3 PRODUCT ARCHITECTURE OVERVIEW

3.1 Introduction

FpML incorporates a significant level of structure, rather than being a 'flat' representation of data. This structuring is achieved through the grouping of related elements describing particular features of a trade into components. Components can both contain, and be contained by, other components.

An alternative approach would have been to collect all the required elements in a single large component representing a product or trade. A flat structure of this kind would capture all the relevant information concisely but would also constrain the model in two important respects, namely, ease of implementation and extensibility.

Grouping related elements into components makes it easier to validate that the model is correct, that it is complete and that it doesn't contain redundancy. This is true, both from the perspective of readability to the human eye, and also from the perspective of processing services. Processing services that do not need all the information in a trade definition can isolate components and be sure that the complete set of elements required, and only the elements required, is available for the particular process in hand.

Components additionally serve as the building blocks for a flexible and extensible model. Generally speaking, the complexity of financial products is a result of combining a few simple ideas in a variety of different ways. The component structure supports a trade content definition that is flexible enough to represent the wide variation of features found in traded financial instruments.

It should be noted that the application of the guiding principles of extensibility and ease of use has resulted in a different approach with regard to the forward rate agreement. Because this product is straightforward, commoditized and unlikely to develop further, the advantage to be gained from the extensive use of components is outweighed by the concision of a single component.

3.2 Component Framework

The optimum level of granularity is important to FpML. FpML separates the elements which collectively describe a feature of a product or trade into a separate component with each component serving a particular semantic purpose. Every grouping of elements in FpML is regarded as a component and each component is regarded as a container for the elements that describe that component. In the majority of cases each component will contain a mixture of other components and primitive elements, e.g. a date or string, that collectively describe the features of the component. Components are typically represented in the FpML Document Type Definition (DTD) as XML entities.

Generally speaking, the lower level a component is, the more re-usable it will be. FpML makes use of a number of primitive entity components that describe the basic building blocks of financial products, for example, `FpML_Money`, `FpML_AdjustableDate`, `FpML_BusinessCenters`, `FpML_Interval`, `FpML_BusinessDayAdjustments` etc. These primitive components are re-used in different business contexts.

Primitive components are contained in higher level components that describe the features of particular products. For this reason these higher level components will tend not to be re-usable to the same extent. Examples within the definition of `swapStream` are the components required to construct schedules of

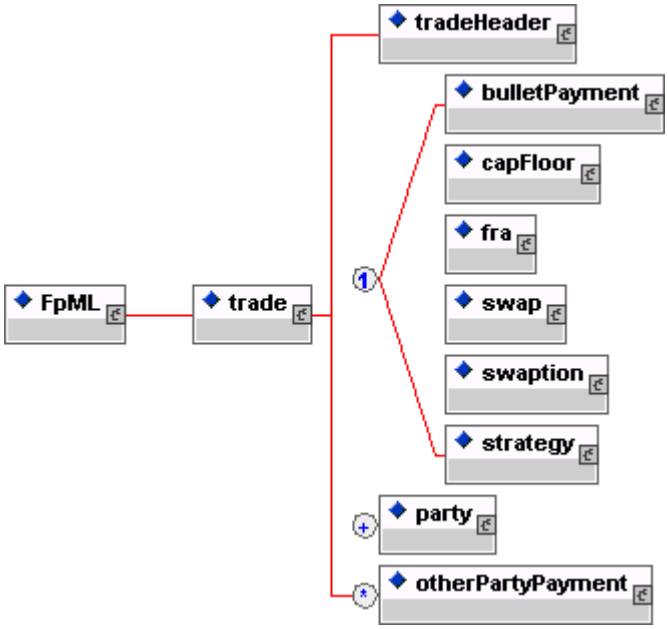
dates such as `calculationPeriodDates`, `resetDates` and `paymentDates`. However, it should not be inferred from this that any fundamental distinction is drawn between components in usage or structure.

3.3 Overview of Core Trade Components

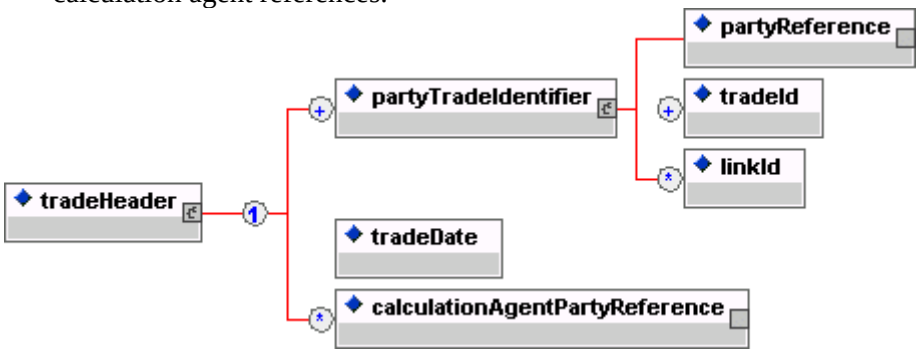
3.3.1 The Trade Component

The `trade` is the top-level component within the root element `FpML`. A trade is an agreement between two parties to enter into a financial contract and the `trade` component in FpML contains the economic information necessary to execute and confirm that trade. A `trade` contains four components: `tradeHeader`, `product` (an abstract concept) , `party` (two or more instances) and `otherPartyPayment` (zero or more instances).

(See Section 4.1, Interpreting the Diagrams, for an explanation of the graphical DTD representation shown in the following schematics)



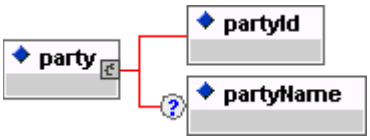
- **`tradeHeader`** - The information within `tradeHeader` will be common across all types of trade regardless of product. In FpML 2.0 this contains the trade date, party trade identifiers and any calculation agent references.



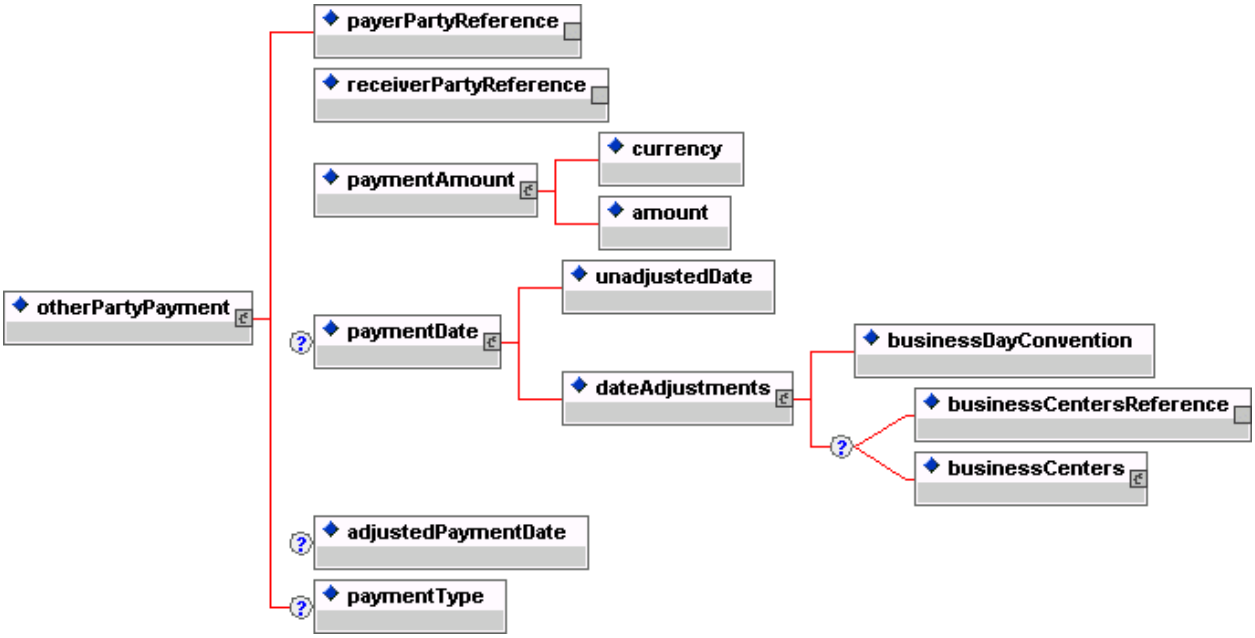
- **`product`** – Product is an abstract concept in fpml, the product element does not appear. Instead one of the fpml products (bulletPayment, capFloor, fra, swap, swaption or strategy) will appear directly under trade. This is currently restricted in FpML 2.0 to interest rate products. A strategy product has been introduced which allows for multiple products to be included in a trade to allow for structured trades. These contain the economics of the trade and are described section later.

- **`party`** - The `party` component holds information about a party involved in the trade. The parties involved will be the principals to the trade and potentially additional third parties such as a broker. For this release, this component is restricted to party identification.

  It should be noted that an FpML document is not 'written' from the perspective of one particular party, i.e. it is symmetrical with respect to the principal parties. The particular role that a party plays in the trade, e.g. buyer, seller, stream payer/receiver, fee payer/receiver, is modeled via the use of references from the component where the role is identified to the `party` component.



- **`otherPartyPayment`** – This component contains additional payments such as brokerage paid to third parties which are not part of the economics of a trade itself.



3.3.2 The Product Component

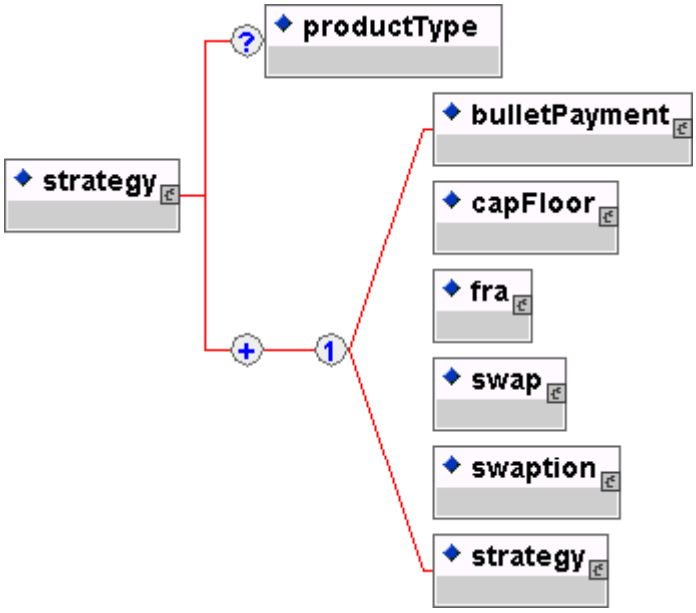
The product component specifies the financial instrument being traded. This component captures the economic details of the trade. Because of the complexity of the OTC Interest Rate Derivatives domain

that FpML 2.0 covers, composing these products from various building blocks is a key aspect of the design approach.

FpML 1.0 focused on the instrument definitions for interest rate swaps (including cross currency swaps) and forward rate agreements. For that initial release, a `trade` was restricted to containing only a single product definition. In FpML 2.0 product strategy has been introduced which allows more than one product within a trade. In FpML 2.0 the instrument definition has been extended to include options.

3.3.3 The Strategy Component

The strategy component is itself a product. This component allows the structuring of trades by combining any number of products within a strategy. This component makes use of a composition pattern since strategy itself is a product. This means that strategies can themselves contain strategies.



3.4 Coding Schemes

A necessary feature of a portable data standard is both an agreed set of elements and an agreed set of permissible values (the value domain) for those elements. An FpML document exchanged between two parties would not be mutually understandable if either or both of the parties used internal or proprietary coding schemes to populate elements.

Reference data can originate from various sources and the range of permitted values may be more or less extensive. The `dayCountFraction` is an example of an element with a limited set of permissible values with well-defined meanings. The range of permitted values comes from several sources including ISDA and AFB definitions. However, the `currency` element is an example of where the list of permitted values is more extensive and the coding scheme reference is to a well-known standard, in this case ISO 4217.

In FpML 1.0 the recommended domain for party identification is a valid bank identifier code (BIC). The BIC is an ISO standard, ISO 9362. S.W.I.F.T. is the designated registration authority for the assignment of BIC codes.

One possible means of identifying value domains would have been to include the domain of permitted values within the DTD. This solution has been rejected for two reasons. Firstly, in many cases the scope of permitted values is extensive, most obviously with party identifiers, and this would make the standard unnecessarily bulky. Secondly, although there are varying degrees of stability, all value domains are subject to change and including them in the DTD would have necessitated a new version of FpML each time a value domain changed.

For these reasons, FpML uses Schemes to identify the permitted values for an element. In each case, the reference Scheme will be identified by a URI. The URI will either identify a well-known external standard such as ISO 4217, or where no well-established standard exists, an FpML standard. FpML 1.0 includes provision for a default Scheme and the facility to override the default Scheme at an element level. In both cases, no values are included for the URI in the DTD in order to avoid coding either particular Schemes, or particular versions of Schemes, into FpML. For the same reason, the URI quoted in an FpML document for a Scheme that is FpML controlled will include a date and version in order to identify the particular version referenced.

It should be noted that the Scheme approach adopted by FpML does not allow validation of the values within the DTD. It will be the responsibility of the applications that implement FpML to validate that the contents of an element conform to the specified Scheme.

For further details on the architectural framework behind Schemes, refer to the FpML Architecture Version 1.0 document.

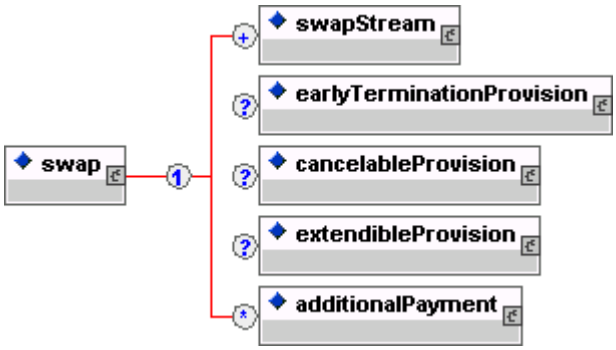
4 INTEREST RATE DERIVATIVE PRODUCT ARCHITECTURE

4.1 Interest Rate Swap

A `swap` component contains one or more instances of the `swapStream` component, zero or more instances of the `additionalPayment` component together with an optional `cancelableProvision` component,an optional `extendibleProvision` component and an optional `earlyTerminationProvision` component. A `swapStream` contains the elements required to define an individual swap leg.

Within an FpML swap there is no concept of a swap header. Details of payment flows are defined within `swapStreams` which each contain their own independent parameters. There can also be `additionalPayment` elements that contain fees. The `additionalPayment` component is identical to the `otherPartyPayment` component shown above.

FpML 2.0 adds option related features. These include cancelable, extendible swaps and early termination provisions. Combining these together with swaptions into a single component was considered but rejected in favour of identifying the different option types with their own components. This provided more clarity and allowed for easier combination of the different options into a single trade. As such a swap can contain a `cancelableProvision`, `extendibleProvision` and an `earlyTerminationProvision`. All these components are very similar (and similar to the `swaption` component), re-use is achieved by using shared entities within each of the components.



FpML supports two representations of a swap stream; a parametric representation, and a cashflows representation. The parametric representation is designed to capture all the economic information required regarding dates, amounts and rates to allow trade execution and confirmation. The parametric representation is mandatory. The cashflows representation specifies an optional additional description of the same stream. The main purpose of this is to allow the inclusion of adjusted dates within an FpML representation of a trade. It can also be used to represent adhoc trades not achievable by easy manipulation of the parameters of a stream (i.e. by changing the adjusted dates). This would lead to the cashflows not matching those generated by the parameters (see more discussion later) and would also render the representation of the trade unsuitable for a confirmation. The spirit of FpML is that such manipulation of cashflows would be achieved by splitting a single stream into a number of streams though it is recognized that this may be impractical in some systems.

The cashflows representation is not self contained as it relies on certain information contained within the stream's parametric definition. The elements required from the parametric definition to complete the cashflows representation are:

- The following elements and their sub-elements within the `calculationPeriodAmount` element:

- `floatingRateIndex`
- `indexTenor`
- `rateTreatment`
- `finalRateRounding`
- `averagingMethod`
- `negativeInterestRateTreatment`
- `dayCountFraction`
- `discounting`
- `compoundingMethod.`

- The following elements and their sub-elements within the `stubCalculationPeriodAmount` element:

- `floatingRateIndex`
- `indexTenor.`

The inclusion of the cashflows representation is intended to support application integration. For example, a financial institution may have one application that captures trade parameters and constructs the trade schedules and then publishes the result for use by other applications. In this case it may be either undesirable, or impossible, for each of the subscribing applications to store and calculate schedules.

The flexibility of the cashflows representation also allows payment and calculation schedules which can not be fully represented by the parametric description. If this situation arises, the mandatory parametric data should still be included in the document and the flag `cashflowsMatchParameters` should contain the value `false` to indicate that it is not possible to generate the cashflows from this parametric data. The setting of this flag to `true` means that the cashflows can be regenerated at any time without loss of information.

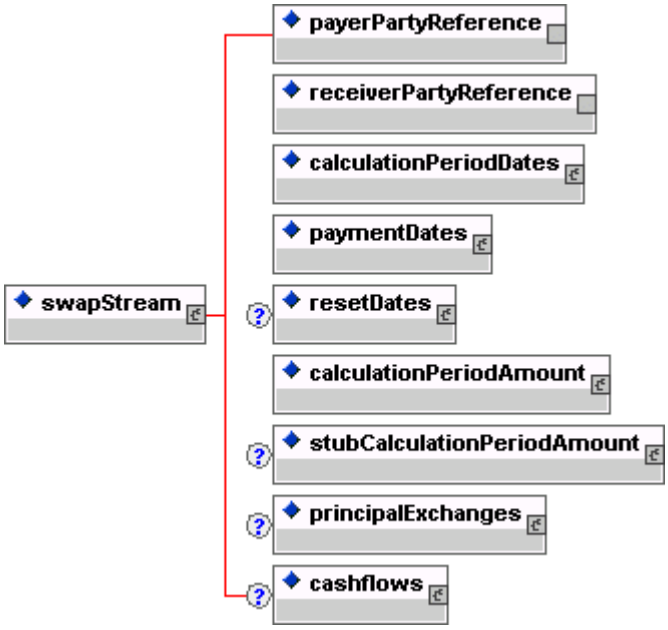
Parties wishing to take advantage of the facility for specifying cashflows which are inconsistent with the parametric representation will need to specify additional rules for how the parametric representation should be processed. This applies to both the creation of the parametric data as well as its interpretation.

The cashflows representation specifies adjusted dates, that is, dates that have already been adjusted for the relevant business day convention using the relevant set of business day calendars (lists of valid business days for each business center). The FpML standard does not specify the source of these business day calendars. This may lead applications to generate differing cashflow representations from the same parametric representation if they use different business day calendars. The use of adjusted dates also produces schedules that are only valid at a particular instance of time. Additional holidays for a business center may subsequently be introduced that would result in changes to the adjusted dates, which would not be reflected in the cashflows representation.

Analogous to cashflows being used to represent adjusted dates, with the addition of options it was important to be able to represented the adjusted dates associated with an option. Thus, where appropriate, a component includes an optional element to represent a schedule of adjusted dates for the option. Such a schedule would include details of adjusted dates such as adjusted exercise dates and cash settlement dates.

In general, an interest rate swap will be a swap with a fixed leg and a floating leg, two floating legs, or two fixed legs. However, certain types of trades may contain more than two legs. FpML does not restrict the number of legs that may be defined. From a modeling perspective, FpML does not distinguish between a swap leg referencing a fixed rate and a swap leg referencing a floating rate, the difference being indicated by the existence, for example, of the `resetDates` component in a floating rate leg.

The structure of a `swapStream` is shown diagrammatically below:



The components within a `swapStream` cannot be randomly combined and cannot be thought of as existing in their own right; they only make sense in the given context and in relationship to other components within the `swapStream` container.

In FpML, the schedule of dates within a `swapStream` is based around the `calculationPeriodDates` component. The definition of a calculation period in FpML differs in some respects from the International Swaps and Derivatives Association (ISDA) definition of Calculation Period. In the case of a trade involving compounding, ISDA introduces the concept of a Compounding Period, with several Compounding Periods contributing to a single Calculation Period. The FpML calculation period is equivalent to the ISDA definition of Compounding Period when compounding is applicable, i.e. the calculation period frequency will correspond to the compounding frequency. An FpML calculation period is directly comparable to the ISDA defined Calculation Period when only one calculation period contributes to a payment.

The other date components within `swapStream` are related to the `calculationPeriodDates` component. The `paymentDates` and `resetDates` components contain the information necessary to construct a schedule of payment and reset dates relative to the calculation period dates.

FpML uses the ISDA Floating Rate Option to specify the floating rate being observed. This scheme was used rather than attempting to parameterize into elements because although most floating rate indices are defined fully by a standard set of parameters (namely index, currency and fixing source) there are sometimes other details including fixing offsets and formulas. This approach allows for more flexibility in

adding new floating rate indices without having to introduce new elements, although this comes at the expense of a self contained definition within the standard.

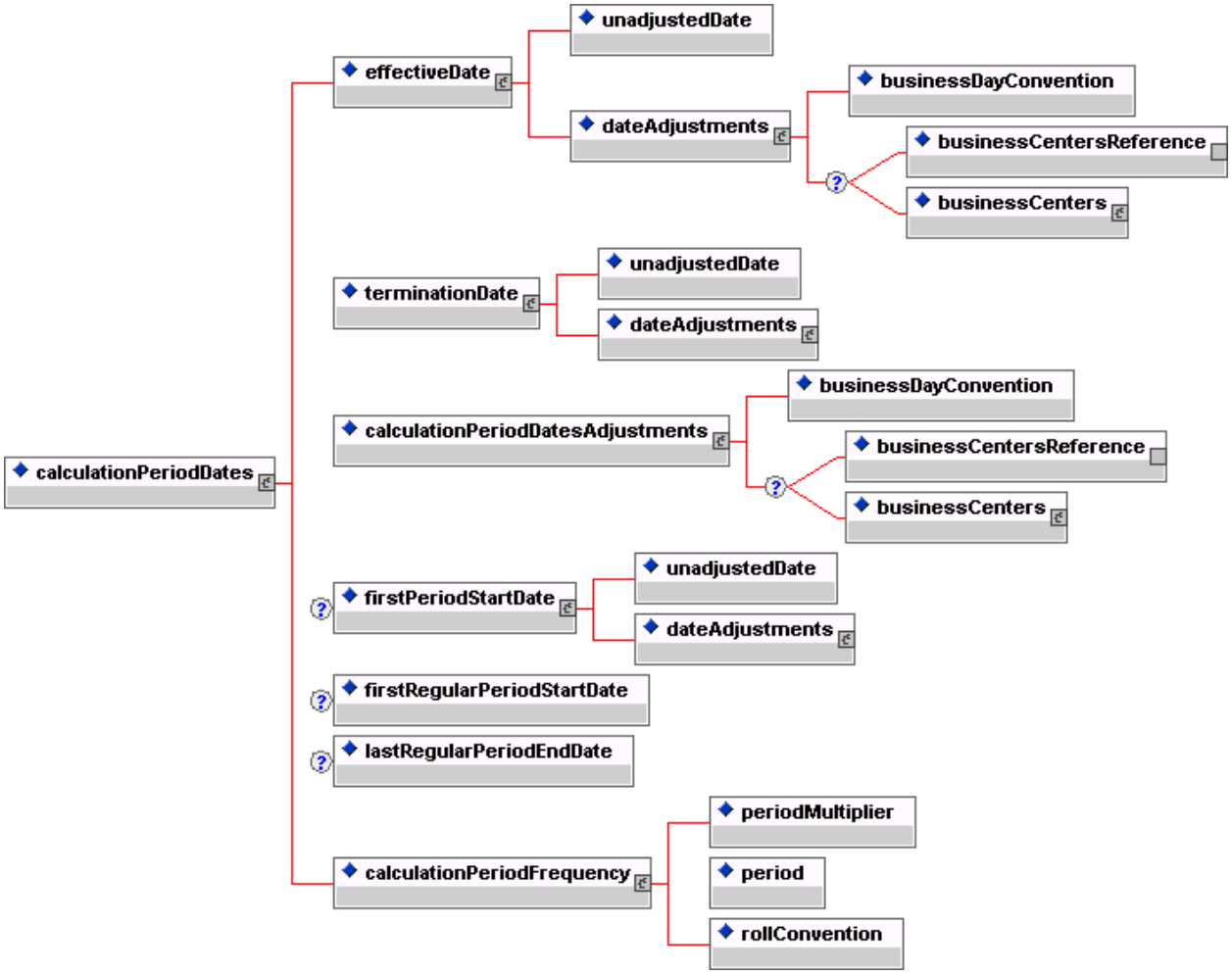
The information relating to amounts and rates is collected in the `calculationPeriodAmount` and `stubCalculationPeriodAmount` components. FpML 2.0 has introduced `fxLinkedNotionalSchedule` as an alternative to `notionalSchedule` for defining notionals. This allows for the definition of FX Resetable trades by allowing for the notional of a stream to be linked to notionals from another stream by way of the spot fx rate.

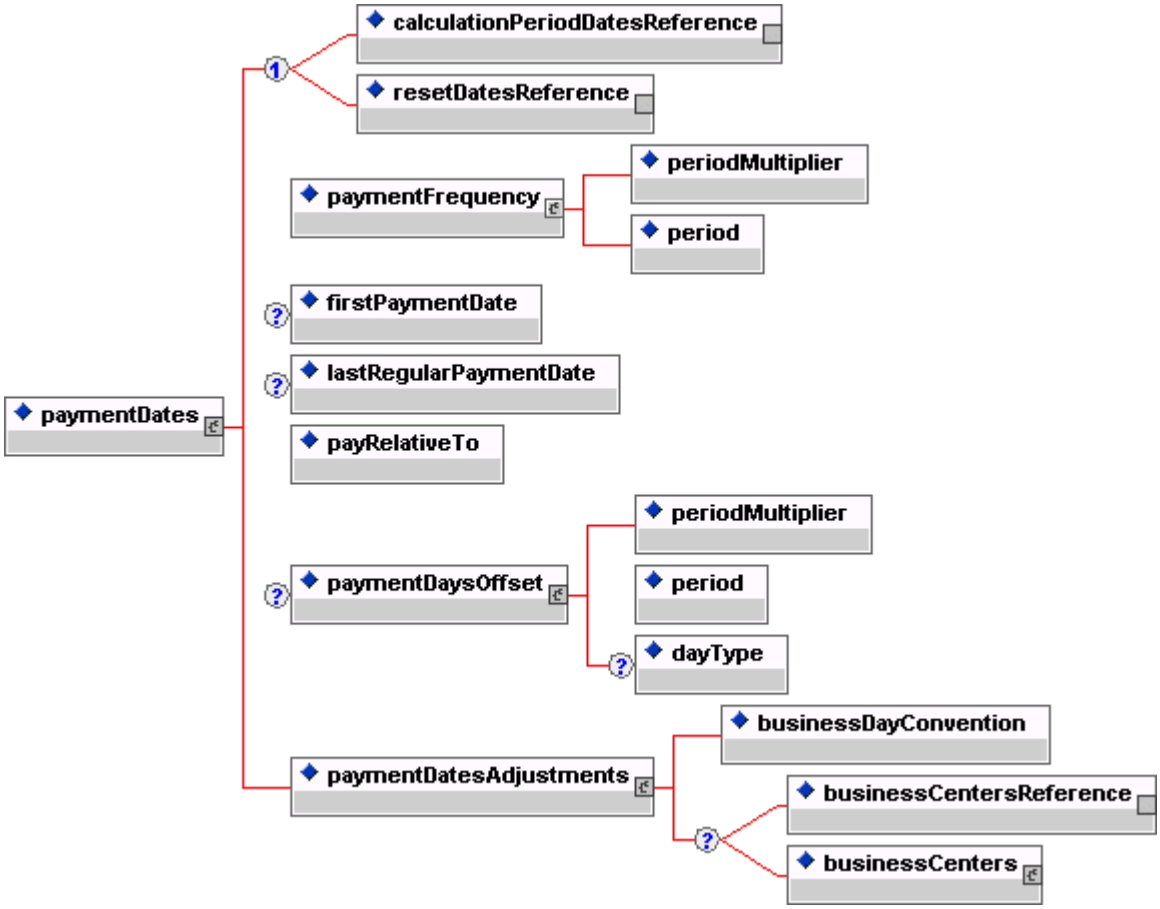
Certain `swapStream` components are designated as being optional (although it would be more accurate to say that they are conditional). Thus a fixed rate stream never includes a `resetDates` component, but this is required for a floating rate stream. Similarly, the `stubCalculationPeriodAmount` component will be required if the swap leg has either an initial or final stub, or indeed both, but should otherwise not be specified. The `principalExchanges` component is required in the case of cross currency swaps or other types of swap involving exchanges of principal amounts.

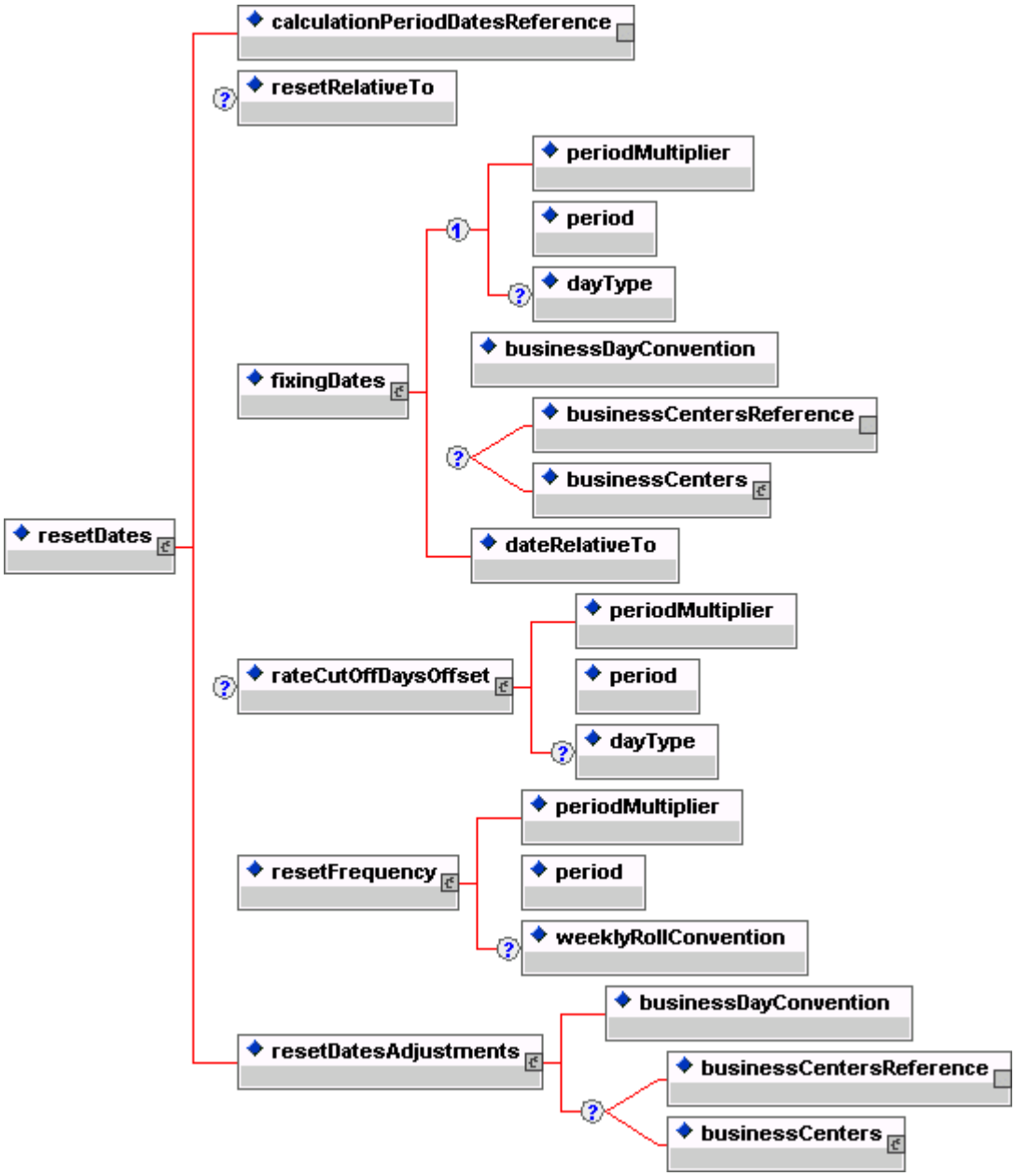
The `payerPartyReference` and `receiverPartyReference` elements indicate which party is paying and which receiving the stream payments. This is done by referencing the appropriate party within the `party` component.

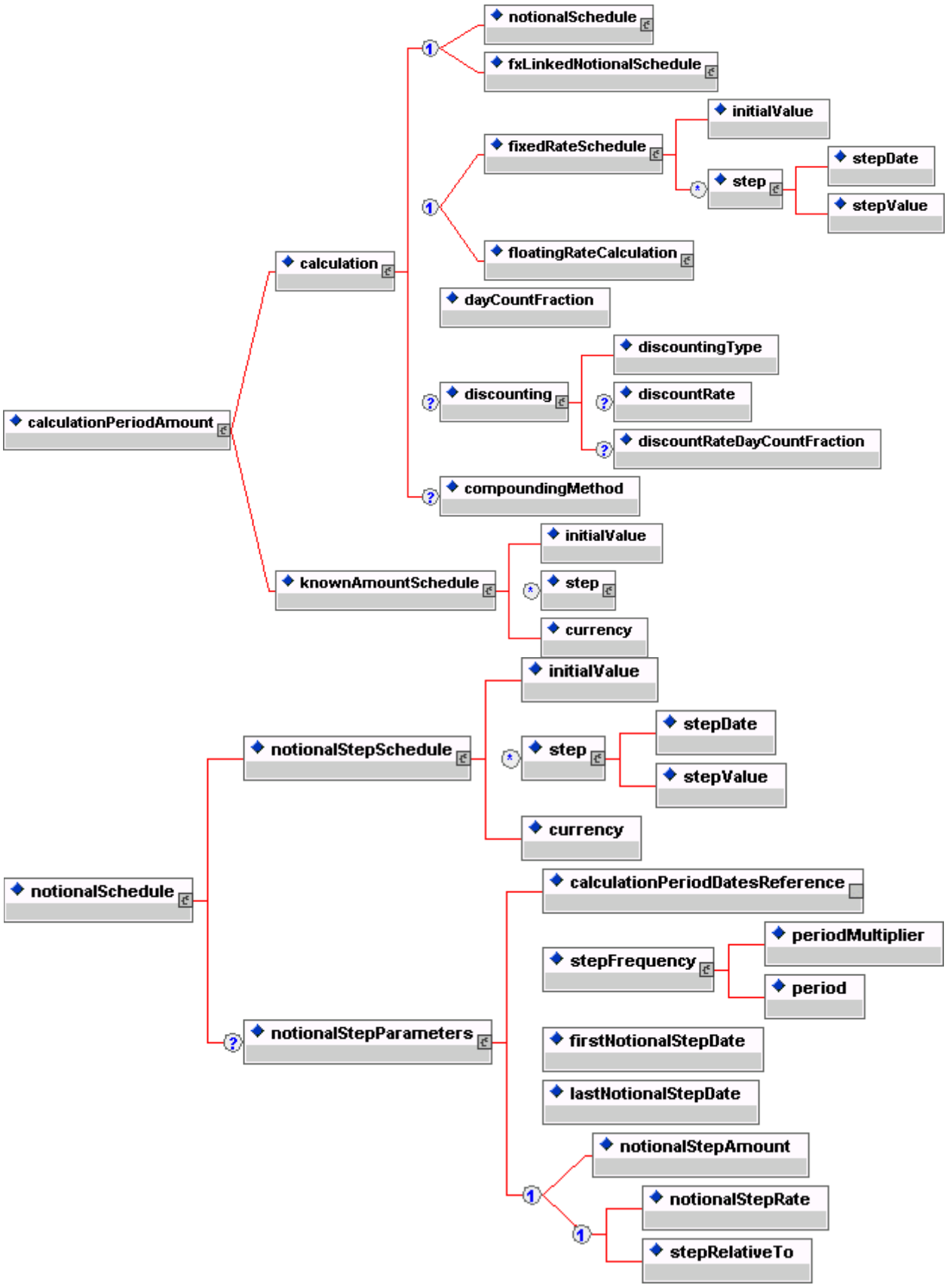
The detailed structures within the `swapStream` are shown diagrammatically below:

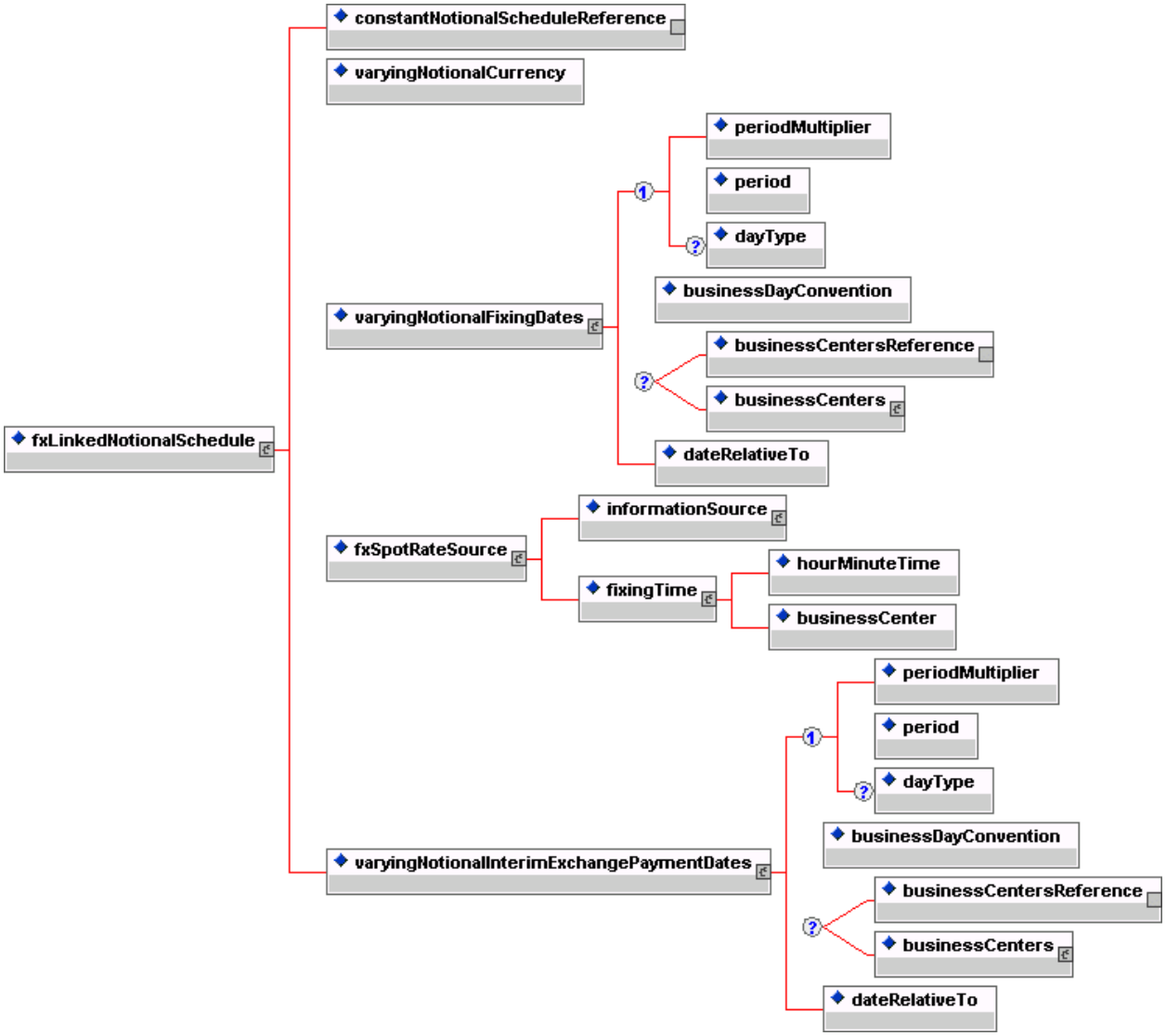
4.1.1 Swap Stream Diagrams

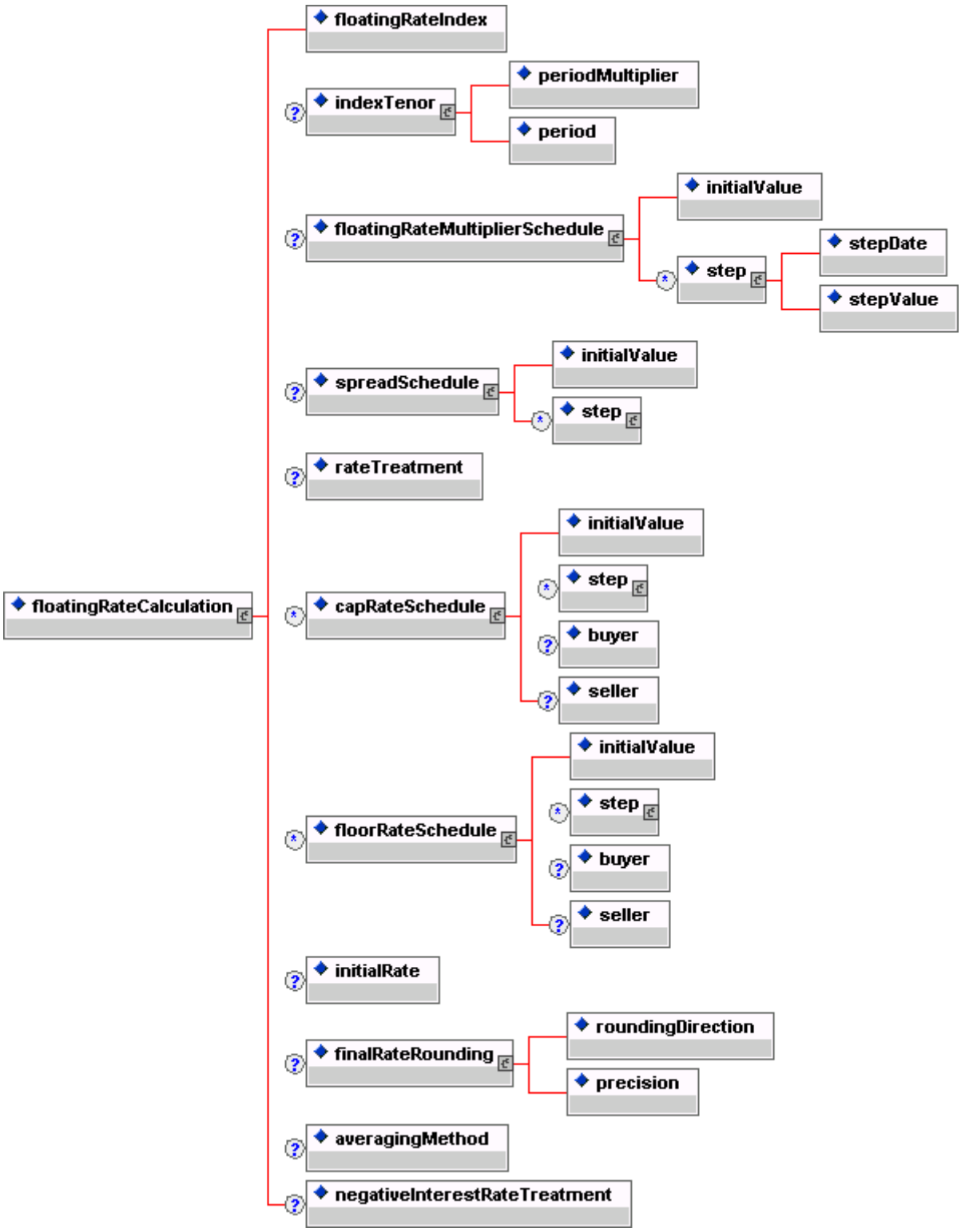


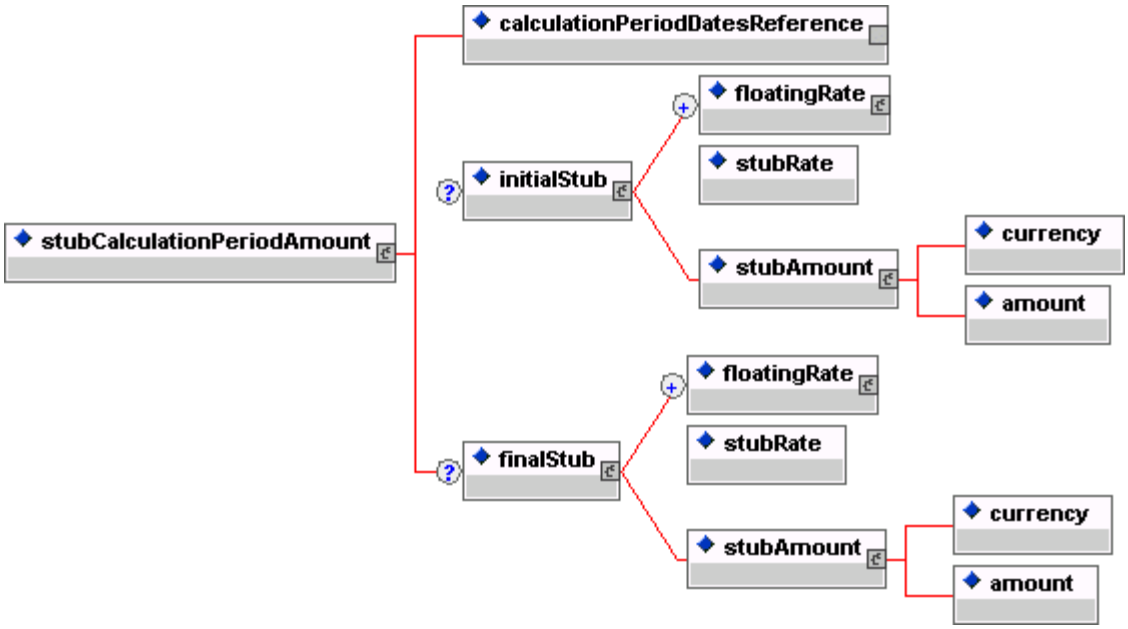

paymentDates
calculationPeriodDatesReference
resetDatesReference
paymentFrequency
periodMultiplier
period
firstPaymentDate
lastRegularPaymentDate
payRelativeTo
paymentDaysOffset
periodMultiplier
period
dayType
paymentDatesAdjustments
businessDayConvention
businessCentersReference
businessCenters

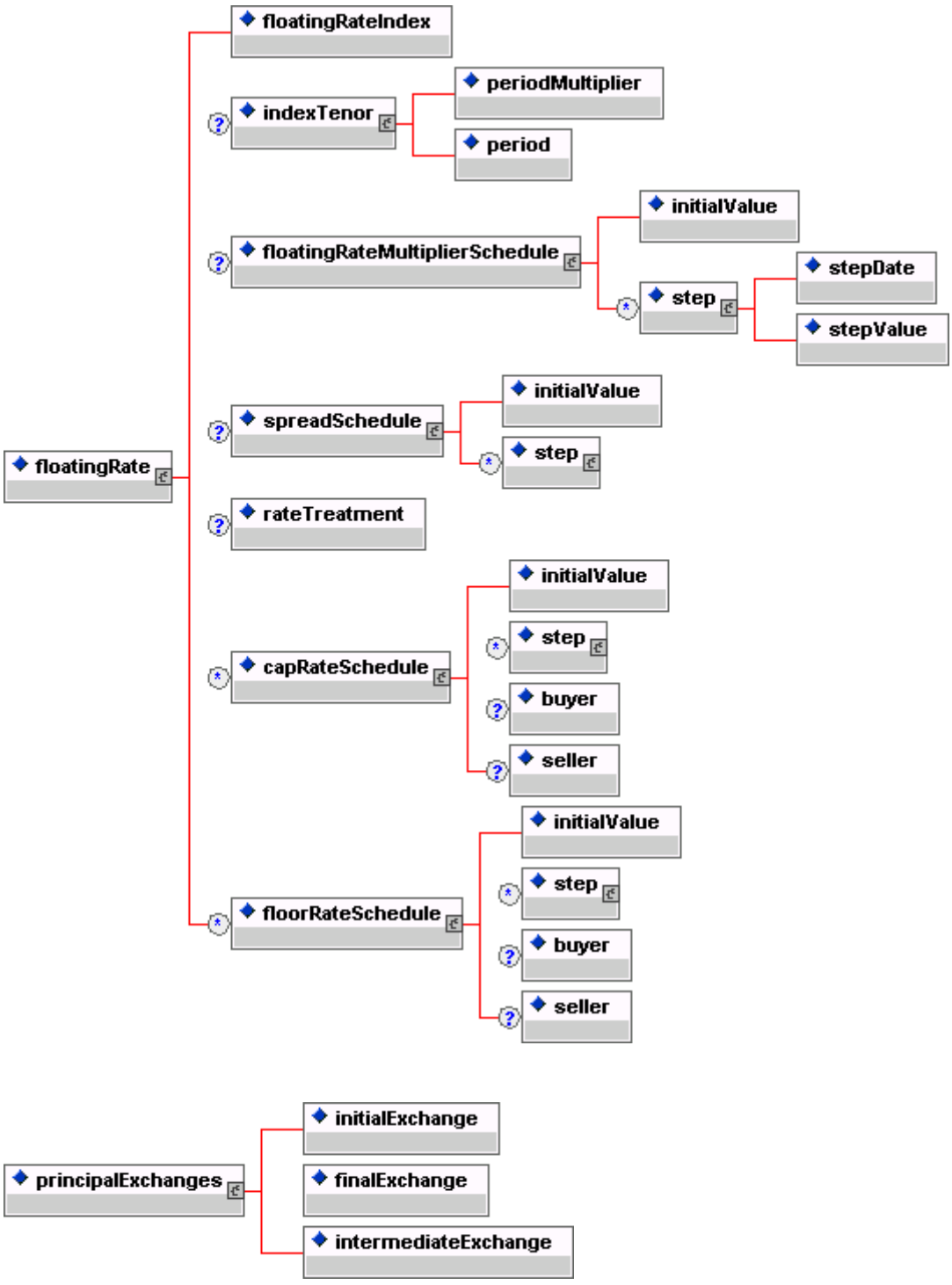

calculationPeriodDatesReference
resetRelativeTo
periodMultiplier
period
dayType
businessDayConvention
fixingDates
businessCentersReference
businessCenters
resetDates
dateRelativeTo
periodMultiplier
rateCutOffDaysOffset
period
dayType
periodMultiplier
resetFrequency
period
weeklyRollConvention
businessDayConvention
resetDatesAdjustments
businessCentersReference
businessCenters

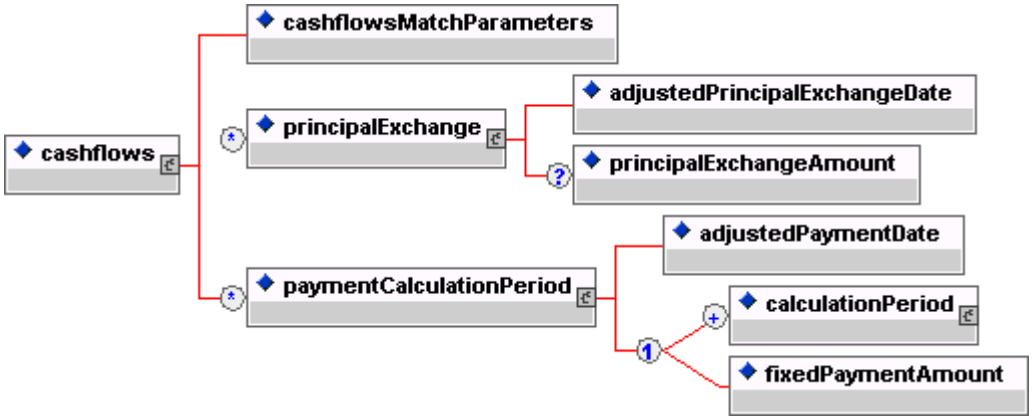

notionalSchedule
fxLinkedNotionalSchedule
initialValue
fixedRateSchedule
step
stepDate
stepValue
floatingRateCalculation
calculation
dayCountFraction
discountingType
discounting
discountRate
discountRateDayCountFraction
calculationPeriodAmount
compoundingMethod
initialValue
knownAmountSchedule
step
currency
initialValue
notionalStepSchedule
step
stepDate
stepValue
currency
calculationPeriodDatesReference
notionalSchedule
periodMultiplier
stepFrequency
period
notionalStepParameters
firstNotionalStepDate
lastNotionalStepDate
notionalStepAmount
notionalStepRate
stepRelativeTo

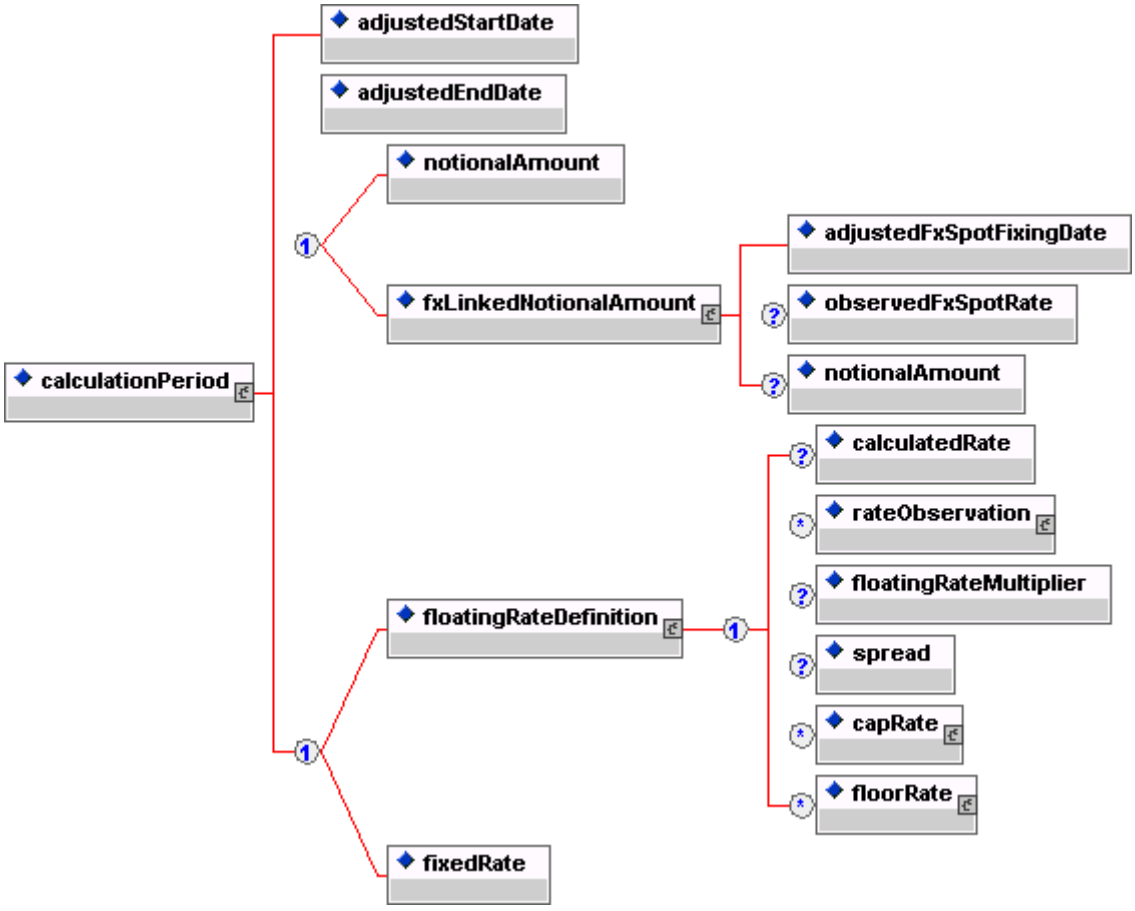

fxLinkedNotionalSchedule
constantNotionalScheduleReference
varyingNotionalCurrency
varyingNotionalFixingDates
periodMultiplier
period
dayType
businessDayConvention
businessCentersReference
businessCenters
dateRelativeTo
fxSpotRateSource
informationSource
fixingTime
hourMinuteTime
businessCenter
varyingNotionalInterimExchangePaymentDates
periodMultiplier
period
dayType
businessDayConvention
businessCentersReference
businessCenters
dateRelativeTo

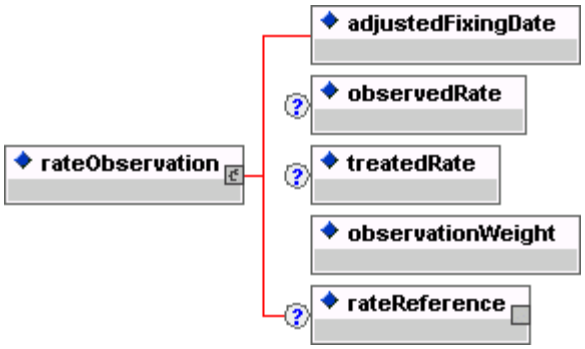

floatingRateCalculation
floatingRateIndex
indexTenor
periodMultiplier
period
floatingRateMultiplierSchedule
initialValue
step
stepDate
stepValue
spreadSchedule
initialValue
step
rateTreatment
capRateSchedule
initialValue
step
buyer
seller
floorRateSchedule
initialValue
step
buyer
seller
initialRate
finalRateRounding
roundingDirection
precision
averagingMethod
negativeInterestRateTreatment


stubCalculationPeriodAmount
calculationPeriodDatesReference
initialStub
floatingRate
stubRate
stubAmount
currency
amount
finalStub
floatingRate
stubRate
stubAmount
currency
amount


floatingRate
floatingRateIndex
indexTenor
periodMultiplier
period
floatingRateMultiplierSchedule
initialValue
step
stepDate
stepValue
spreadSchedule
initialValue
step
rateTreatment
capRateSchedule
initialValue
step
buyer
seller
floorRateSchedule
initialValue
step
buyer
seller
principalExchanges
initialExchange
finalExchange
intermediateExchange


cashflows
cashflowsMatchParameters
principalExchange
adjustedPrincipalExchangeDate
principalExchangeAmount
paymentCalculationPeriod
adjustedPaymentDate
calculationPeriod
fixedPaymentAmount

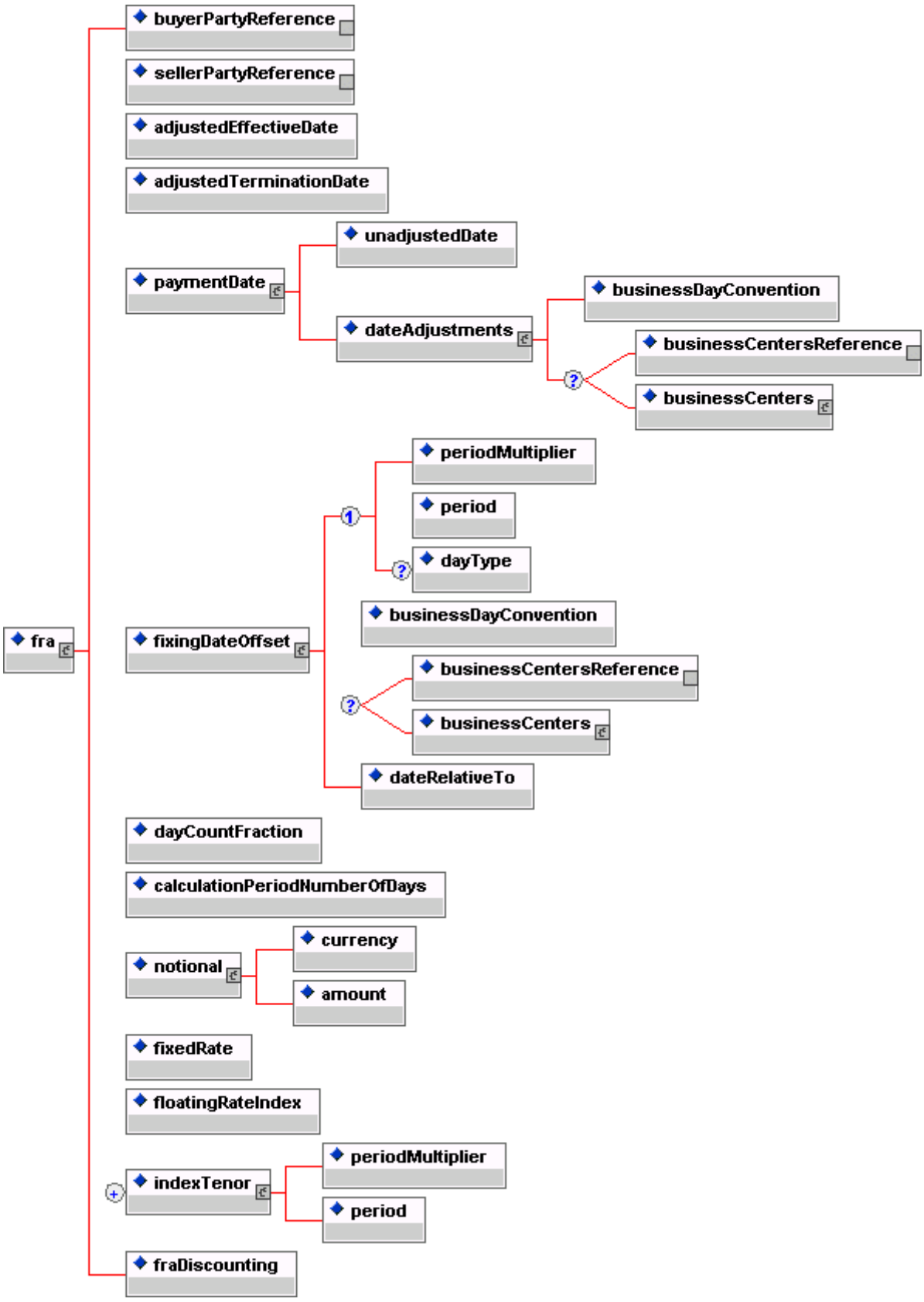

calculationPeriod
adjustedStartDate
adjustedEndDate
notionalAmount
fxLinkedNotionalAmount
adjustedFxSpotFixingDate
observedFxSpotRate
notionalAmount
floatingRateDefinition
calculatedRate
rateObservation
floatingRateMultiplier
spread
capRate
floorRate
fixedRate



4.2 Forward Rate Agreement

As noted above, the definition of a forward rate agreement trade is contained within a single component. A forward rate agreement is a simple and commoditized product. This means there is no variation in the product traded and it is not expected to become more complex in the future.

The structure of the `fra` component is shown diagrammatically below:


fra
buyerPartyReference
sellerPartyReference
adjustedEffectiveDate
adjustedTerminationDate
paymentDate
unadjustedDate
dateAdjustments
businessDayConvention
businessCentersReference
businessCenters
fixingDateOffset
periodMultiplier
period
dayType
businessDayConvention
businessCentersReference
businessCenters
dateRelativeTo
dayCountFraction
calculationPeriodNumberOfDays
notional
currency
amount
fixedRate
floatingRateIndex
indexTenor
periodMultiplier
period
fraDiscounting

4.3 Option Components

FpML 2.0 has introduced interest rate options. The components introduced are:

- Early Termination Provision (Optional or Mandatory)
- Cancelable Provision
- Extendible Provision
- Swaption
- Cap / Floor

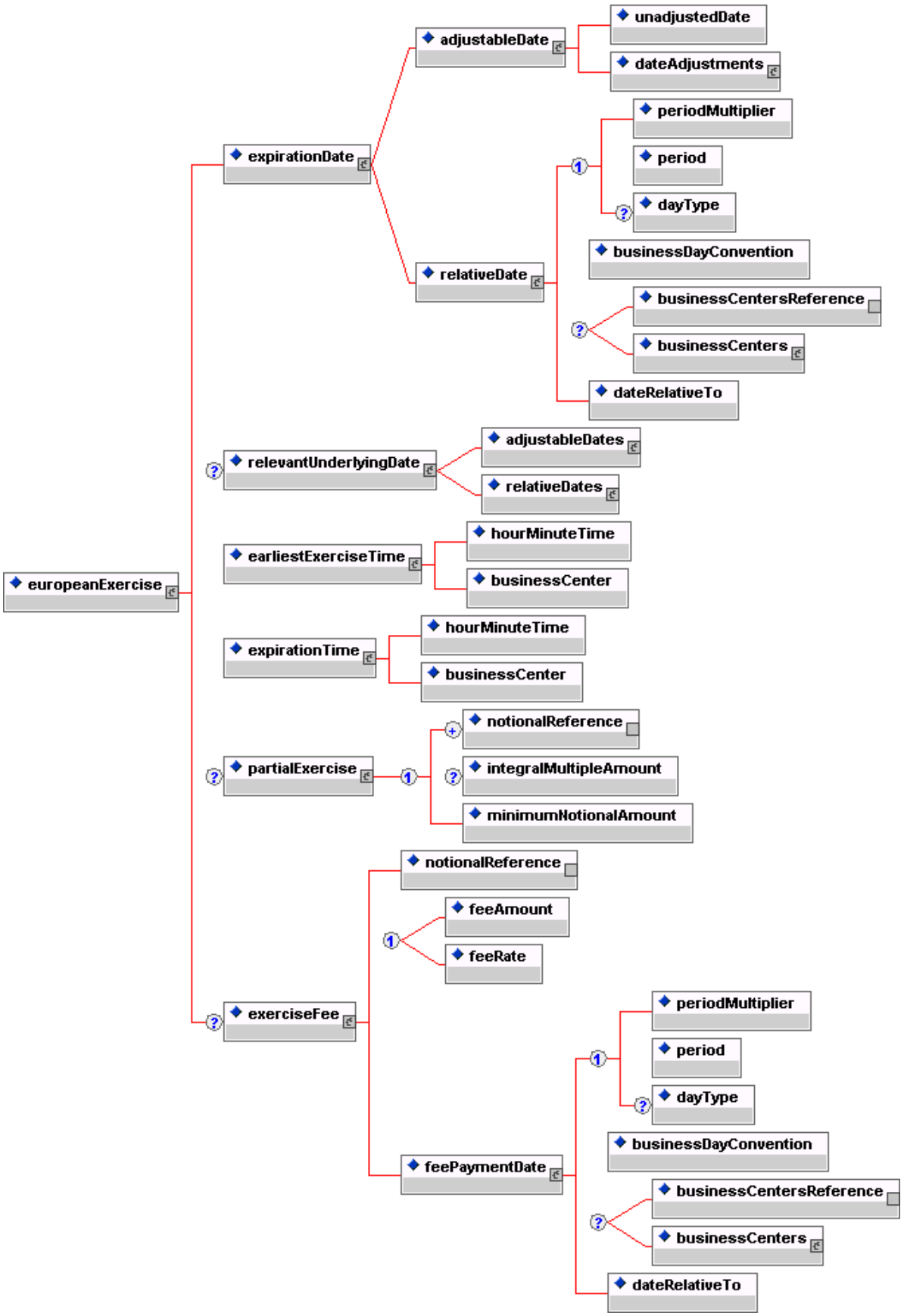
The ISDA 2000 Definitions have been followed closely in defining the various option dates and element names. Thus components for European, Bermudan and American exercise have been defined which are re-used in each of the first four components above. These components share an element called `relevantUnderlyingDate` whose meaning is dependent on the option component it is contained in:

Containing Component	**RelevantUnderlyingDate Meaning**
OptionalEarlyTermination	This represents the new terminationDate of the underlying swapStreams if the trade is terminated early.
CancelableProvision	This represents the new terminationDate of the underlying swapStreams if the trade is cancelled.
ExtendibleProvision	This represents the new terminationDate of the underlying swapStreams if the trade is extended.
Swaption	This represents the effectiveDate of the underlying swapStreams if the swaption is exercised.

4.3.1 European Exercise

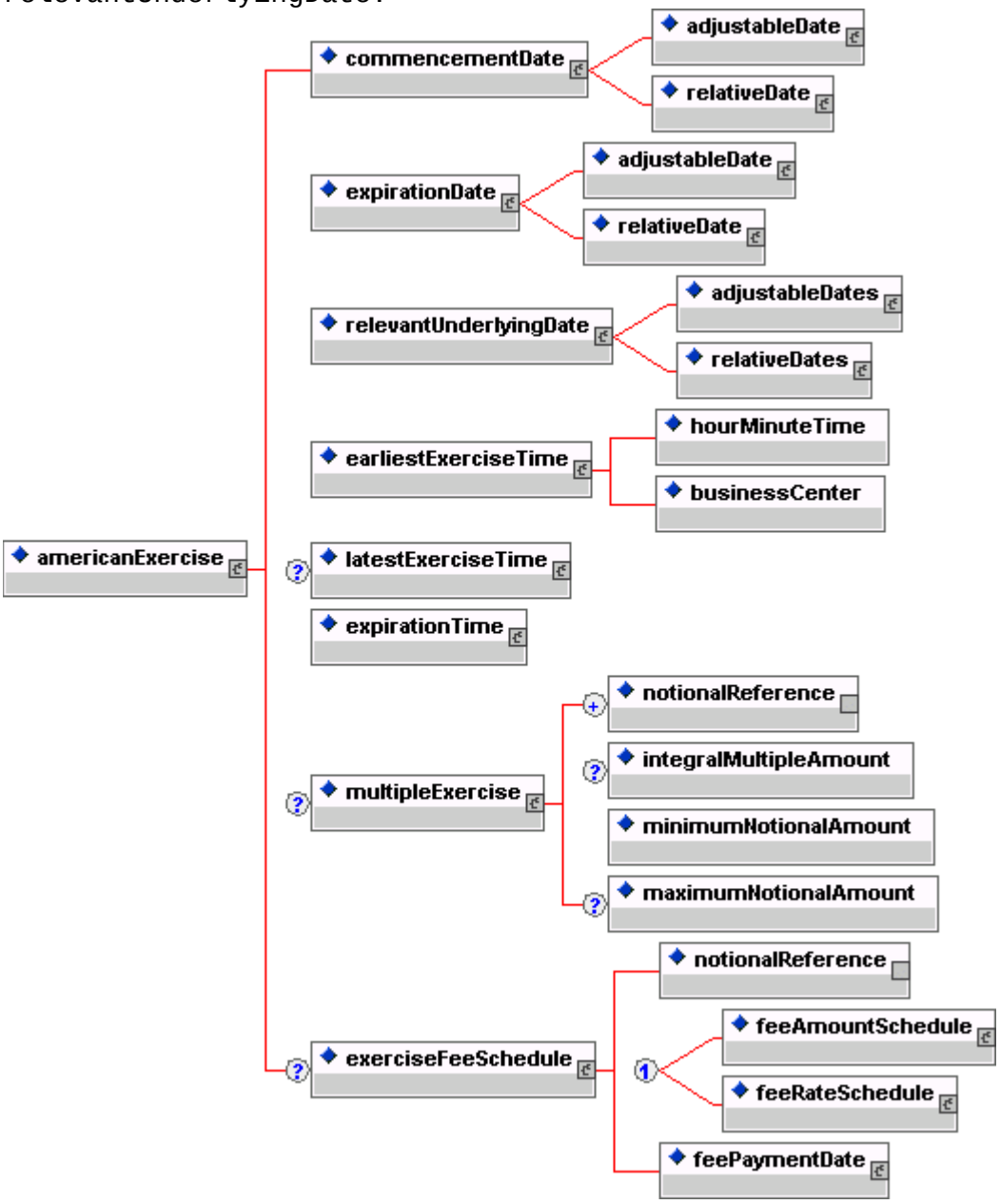
Option exercise on one date. This date can be specified either as an `adjustableDate` or as a `relativeDate` though the latter is only expected to be used in the case of cash settled cancellations where the expiration date may be defined as an offset to the cash settlement payment date.

The `relevantUnderlyingDate` is optional, in its absence the `effectiveDate` of the underlying is the `effectiveDate` defined in the `swapStream`s.


europeanExercise
expirationDate
adjustableDate
unadjustedDate
dateAdjustments
relativeDate
periodMultiplier
period
dayType
businessDayConvention
businessCentersReference
businessCenters
dateRelativeTo
relevantUnderlyingDate
adjustableDates
relativeDates
earliestExerciseTime
hourMinuteTime
businessCenter
expirationTime
hourMinuteTime
businessCenter
partialExercise
notionalReference
integralMultipleAmount
minimumNotionalAmount
exerciseFee
notionalReference
feeAmount
feeRate
feePaymentDate
periodMultiplier
period
dayType
businessDayConvention
businessCentersReference
businessCenters
dateRelativeTo

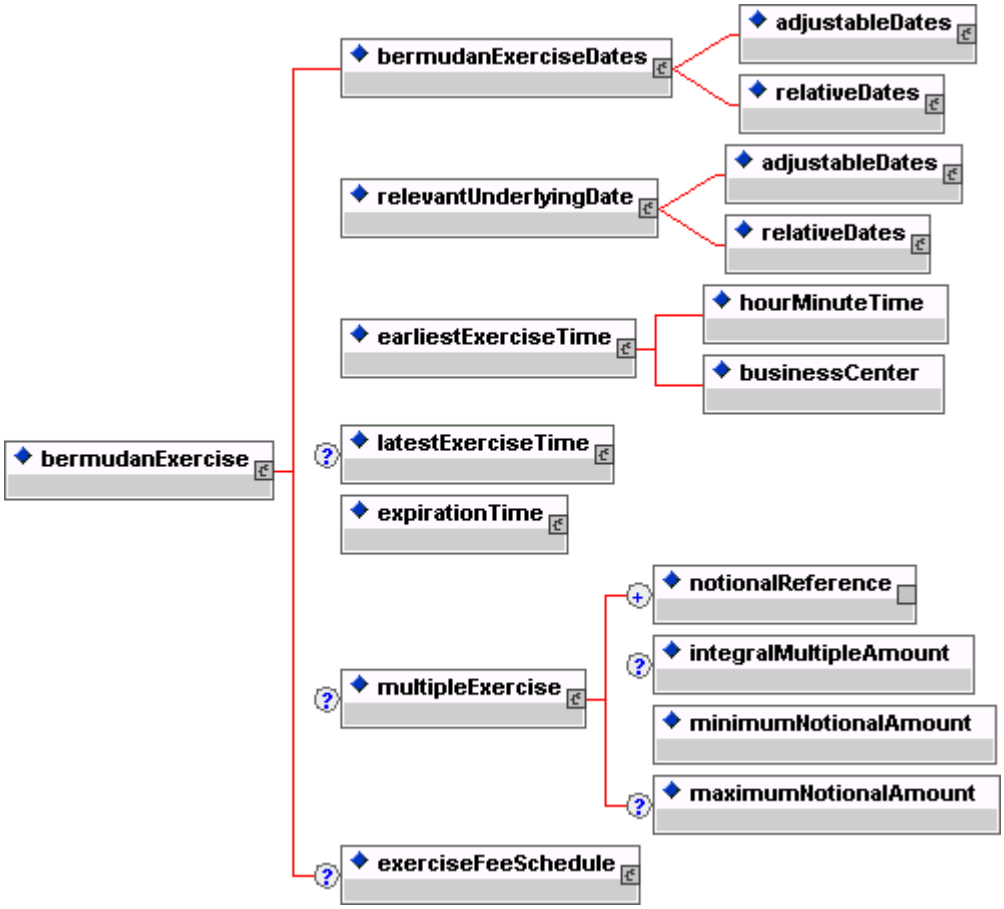
4.3.2 American Exercise

Option exercise within a period. The underlying should specify its effective date based on the earliest possible exercise. When exercise implies a stub period this will be taken to be a short stub at the start, i.e. the underlying swap defines a series of flows, exercise merely brings the flows into existence from the `relevantUnderlyingDate`.



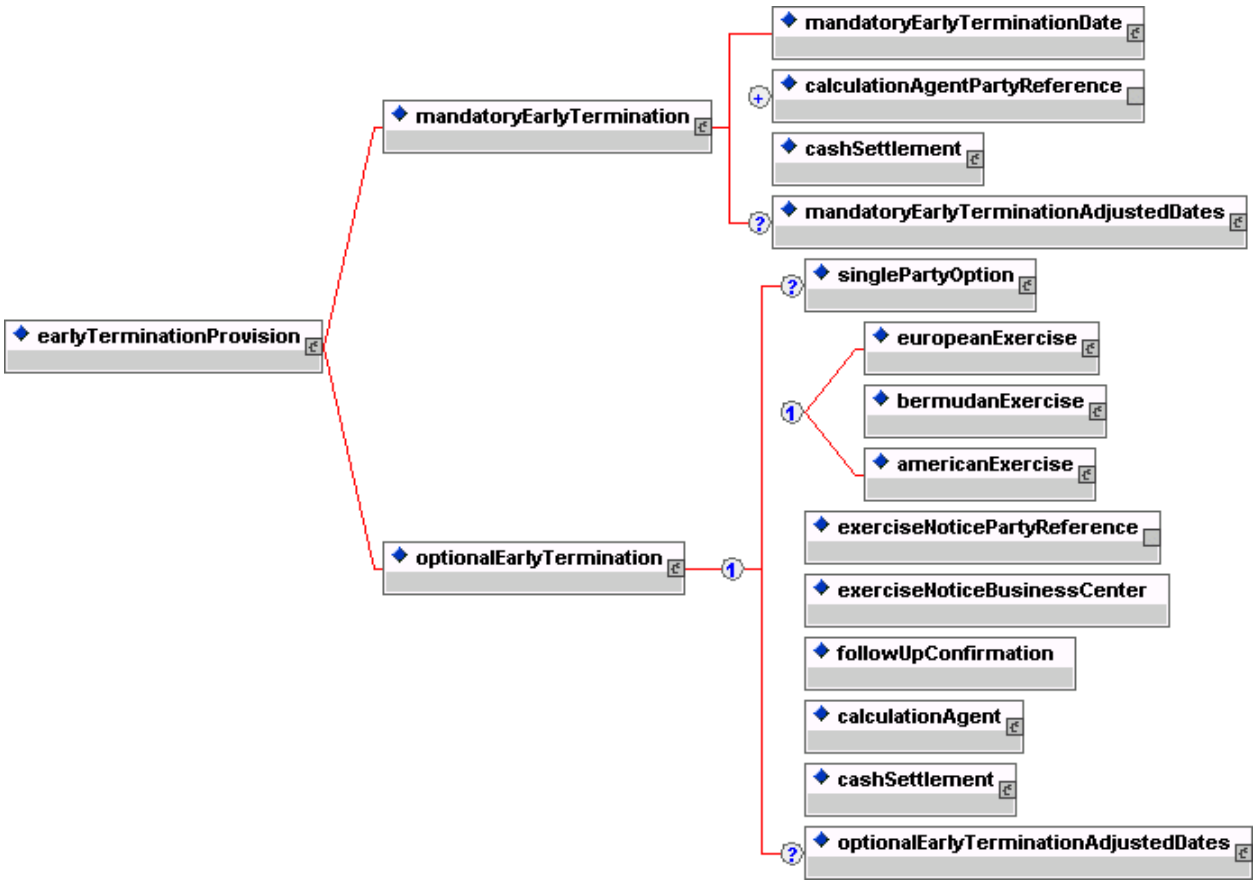
4.3.3 Bermudan Exercise

Option exercise on many discrete dates. These dates can be defined as a list together with adjustments or by reference to an existing schedule elsewhere in the trade (e.g. `resetDates`). In the latter case bounds can be placed on the referenced schedule to define a subset of the whole schedule.



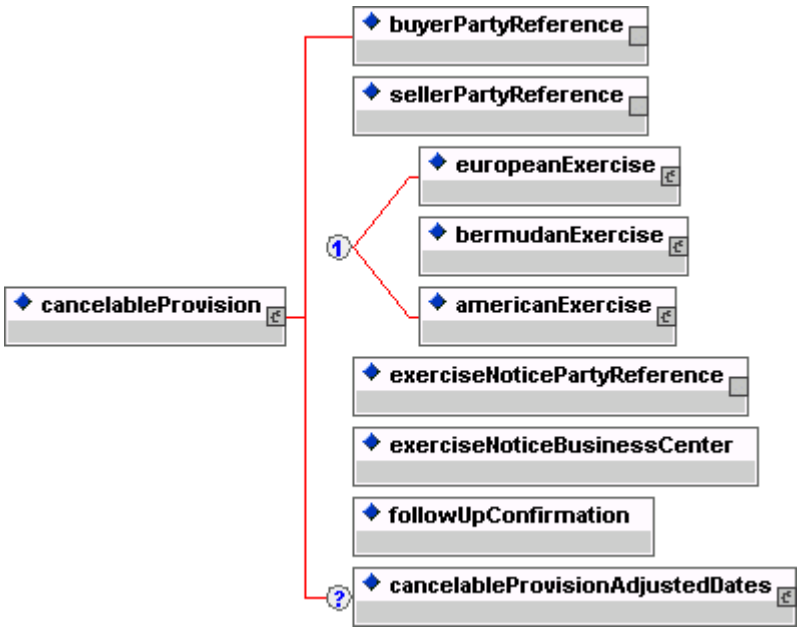
4.3.4 Early Termination Clause

The right for either or both parties to terminate the trade and settle the remaining fair value.



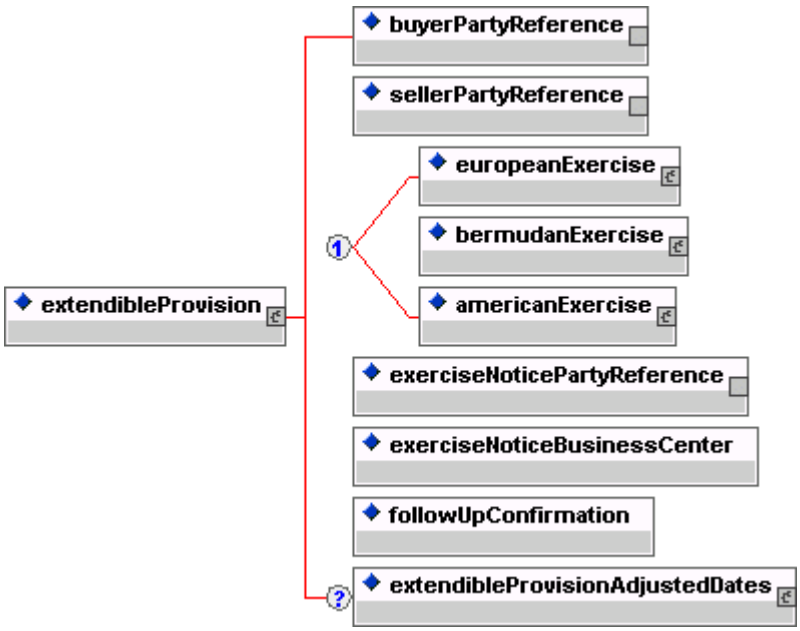
4.3.5 Cancelable Provision

Gives the buyer the right to terminate all `swapStreams`. This does not require the payment of fair value though a fee payment may be specified. This component is very similar to `optionalEarlyTermination`, the only different being that in early termination fair value is paid when the swap is terminated whereas in a cancelable provision the only payments made are those specified in the `exerciseFeeSchedule`.



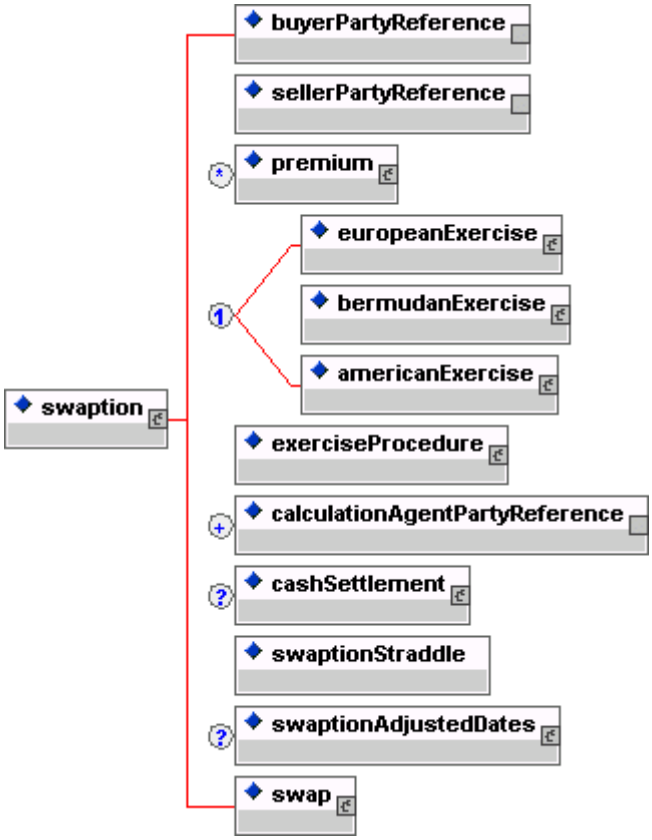
4.3.6 Extendible Provision

Gives the buyer the right to extend all `swapStreams`. This does not require the payment of fair value though a fee payment may be specified. This provision is very similar to a `cancelableProvision`, in fact in terms of market risk the equivalent extendible and cancelable provision have the same risk profile. The FpML standard has made a clear distinction between these two components since the operational risk is high if it is not clear which one is part of the trade. Though each of these components give the right to chose between two termination dates for the swap the required action to shorten or lengthen the swap is opposite in each case. For example, a 10 year swap with the right to cancel after 5 years has exactly the same risk profile as a 5 year swap with the right to extend for 5 years after 5 years, however, the action required on the exercise date is the opposite.



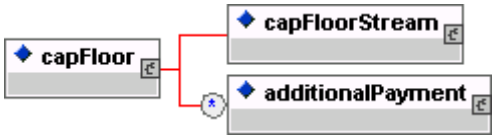
4.3.7 Swaption

The option to enter into a swap is defined as it's own product and contains the underlying swap as a `swap` element. A swaption straddle is defined by setting the `swaptionStraddle` element to `true`: this implies that the swaption buyer has the right, on exercise, to decide whether to pay or receive fixed on the underlying swap. If the underlying does not contain a single fixed stream and a single floating stream then the straddle is invalid and thus this flag should be set to `false`..


swaption
buyerPartyReference
sellerPartyReference
premium
europeanExercise
bermudanExercise
americanExercise
exerciseProcedure
calculationAgentPartyReference
cashSettlement
swaptionStraddle
swaptionAdjustedDates
swap

4.3.8 Cap / Floor

Caps and Floors are defined as one or more `capFloorStreams` and zero or more `additionalPayments`. The `capFloorStream` re-uses the `FpML_InterestRateStream` entity and thus its content is identical to a `swapStream`.



Though a `capFloorStream` allows the definition of fixed streams or known amount streams these are not the intended use of this component and there use would be considered an invalid FpML trade.

The `floatingRateCalculation` component has been amended to allow the specification of cap/floor structures within a single stream (e.g. straddles, corridors). The changes are:

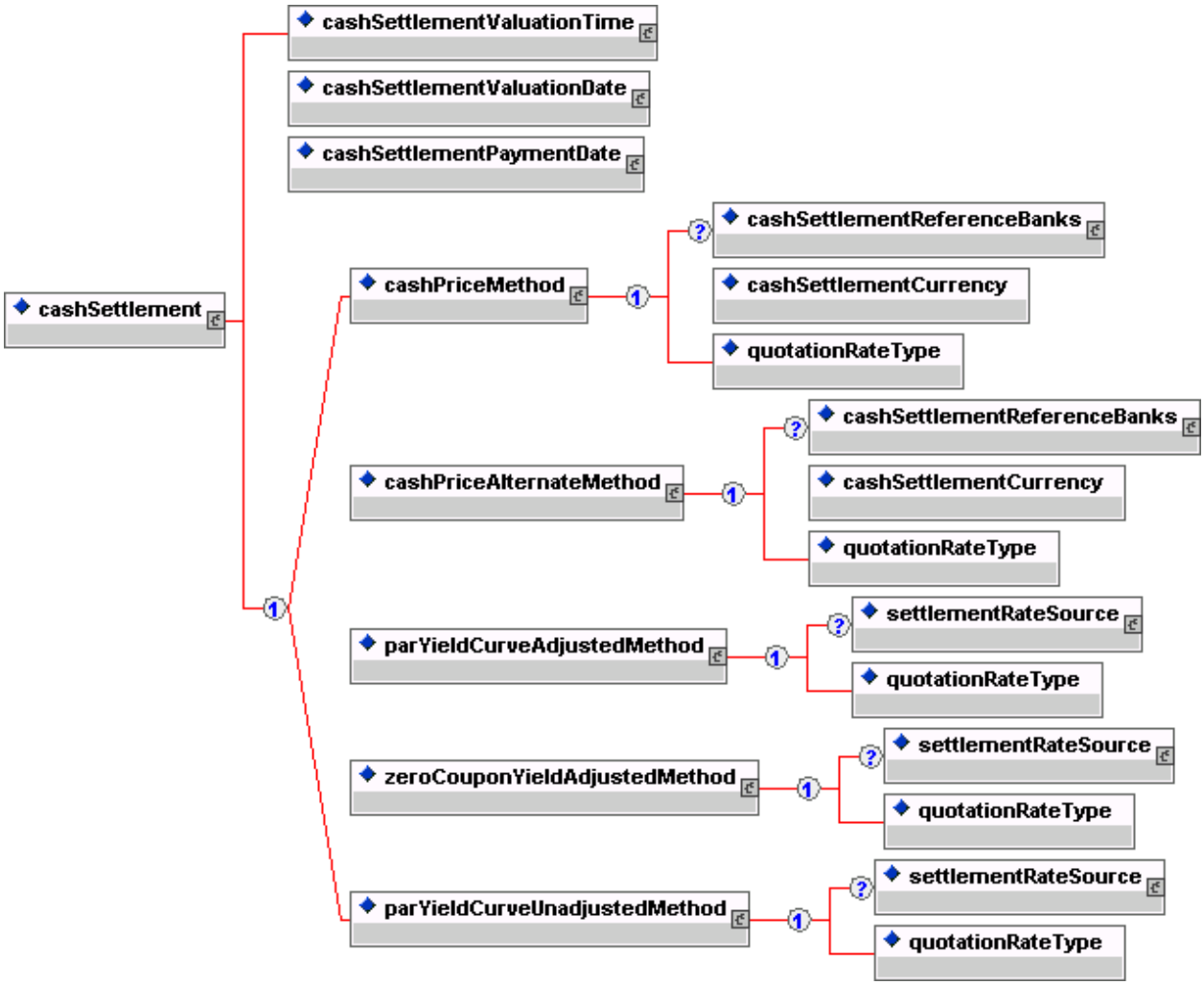
- The single optional `capRateSchedules` and `floorRateSchedules` have been replaced by zero or more `capRateSchedules` and `floorRateSchedules`.
- An optional `buyer` and `seller` reference have been added to these schedules

These additions allow for multiple cap and floor rates to be added to the stream and to define precisely which party bought and sold them. To maintain backward compatibility with FpML1.0 the `buyer` and `seller` are optional. When absent the following rules apply:

	Buyer	**seller**
CapRateSchedule	Stream payer	Stream receiver
floorRateSchedule	Stream receiver	Stream payer

4.4 Cash Settlement

The cash settlement component is used by `mandatoryEarlyTermination`, `optionEarlyTermination` and `swaption`. The language used within the component corresponds to the ISDA language for the various cash settlement methods. Of the five methods included, three share one underlying component and the other two share another component. Thus there is re-use whilst maintaining ease of identification of the type. Also, this approach allows for easy integration of other methods should they arise.


cashSettlement
cashSettlementValuationTime
cashSettlementValuationDate
cashSettlementPaymentDate
cashPriceMethod
cashSettlementReferenceBanks
cashSettlementCurrency
quotationRateType
cashPriceAlternateMethod
cashSettlementReferenceBanks
cashSettlementCurrency
quotationRateType
parYieldCurveAdjustedMethod
settlementRateSource
quotationRateType
zeroCouponYieldAdjustedMethod
settlementRateSource
quotationRateType
parYieldCurveUnadjustedMethod
settlementRateSource
quotationRateType

5 COMPONENT DEFINITIONS

5.1 Interpreting the Diagrams

The DTD source shown below is graphically represented in Figure 4.1. Important features of the diagram are highlighted, which include:

- Graphical representation of an XML entity definition
- Sequence indicators, i.e. comma (,) and vertical bar (|)
- Content specifications, i.e. text or sub-elements
- Occurrence indicators, i.e. can appear zero or once (?), can appear one or more times (+), can appear zero or more times (*).
-

```
<!ENTITY % FpML_Root "SubElementA?,SubElementB+">
<!ELEMENT SubElementA (LeafElementA*,(LeafElementB | LeafElementC))>
<!ELEMENT SubElementB (LeafElementA,LeafElementB,LeafElementC,LeafElementD)*>
<!ELEMENT LeafElementA (#PCDATA)>
<!ELEMENT LeafElementB (#PCDATA)>
<!ELEMENT LeafElementC (#PCDATA)>
<!ELEMENT LeafElementD (#PCDATA)>
```

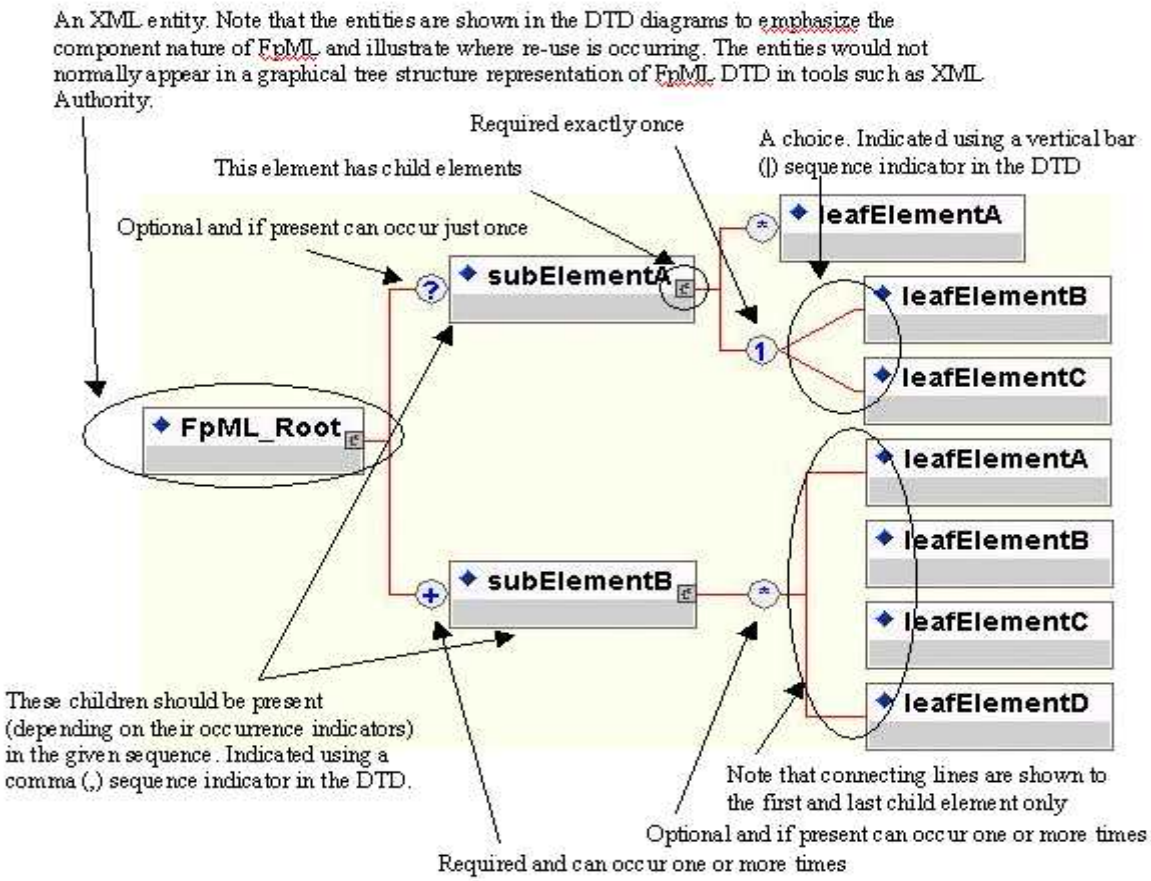


Figure 4.1: Graphical Representation of a DTD

5.2 Root Element Definition

FpML

Description:

The root element in an FpML trade document.

Figure:

Contents:

trade (exactly one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_Trade)

- The FpML trade definition.

DTD Fragment:

```
<!ELEMENT FpML (trade)>
```

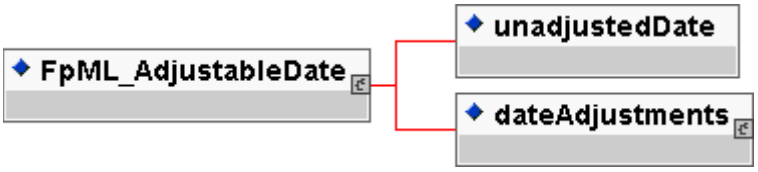
5.3 Entity Definitions

FpML_AdjustableDate

Description:

An entity for defining a date that shall be subject to adjustment if it would otherwise fall on a day that is not a business day in the specified business centers, together with the convention for adjusting the date.

Figure:



Contents:

unadjustedDate (exactly one occurrence; of type *date*)

- A date subject to adjustment.

dateAdjustments (exactly one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_BusinessDayAdjustments)

- The business day convention and financial business centers used for adjusting the date if it would otherwise fall on a day that is not a business day in the specified business centers.

Used by:

adjustableDate
effectiveDate
firstPeriodStartDate
mandatoryEarlyTerminationDate
paymentDate
terminationDate

DTD Fragment:

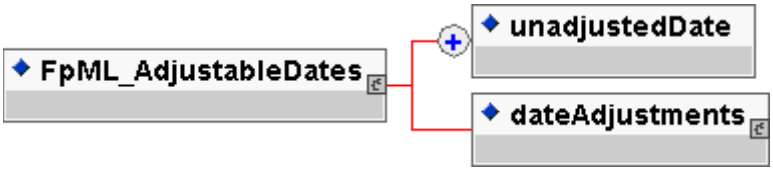
```
<!ENTITY % FpML_AdjustableDate "unadjustedDate , dateAdjustments">
```

FpML_AdjustableDates

Description:

An entity for defining a series of dates that shall be subject to adjustment if they would otherwise fall on a day that is not a business day in the specified business centers, together with the convention for adjusting the dates.

Figure:



Contents:

unadjustedDate (one or more occurrences; of type *date*)

- A date subject to adjustment.

dateAdjustments (exactly one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_BusinessDayAdjustments)

- The business day convention and financial business centers used for adjusting the date if it would otherwise fall on a day that is not a business day in the specified business centers.

Used by:

adjustableDates

DTD Fragment:

```
<!ENTITY % FpML_AdjustableDates "unadjustedDate+ , dateAdjustments">
```

FpML_AdjustableOrRelativeDate

Description:

An entity for the choice between defining a date as an explicit date together with applicable adjustments or as relative to some other (anchor) date.

Figure:

Contents:

Either

adjustableDate (exactly one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_AdjustableDate)

- A date that shall be subject to adjustment if it would otherwise fall on a day that is not a business day in the specified business centers, together with the convention for adjusting the date.

Or

relativeDate (exactly one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_RelativeDateOffset)

- A date specified as some offset to another date (the anchor date).

Used by:

commencementDate
expirationDate

DTD Fragment:

```
<!ENTITY % FpML_AdjustableOrRelativeDate "adjustableDate | relativeDate">
```

FpML_AdjustableOrRelativeDates

Description:

An entity for the choice between defining a series of dates as an explicit list of dates together with applicable adjustments or as relative to some other series of (anchor) dates.

Figure:

Contents:

Either

adjustableDates (exactly one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_AdjustableDates)

- A series of dates that shall be subject to adjustment if they would otherwise fall on a day that is not a business day in the specified business centers, together with the convention for adjusting the date.

Or

relativeDates (exactly one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_RelativeDates)

- A series of dates specified as some offset to another series of dates. (the anchor dates).

Used by:

bermudanExerciseDates
relevantUnderlyingDate

DTD Fragment:

```
<!ENTITY % FpML_AdjustableOrRelativeDates "adjustableDates | relativeDates">
```

FpML_AmericanExercise

Description:

An entity for defining the exercise period for an American style option together with any rules governing the notional amount of the underlying which can be exercised on any given exercise date and any associated exercise fees.

Figure:

Contents:

commencementDate (exactly one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_AdjustableOrRelativeDate)

- The first day of the exercise period for an American style option.

expirationDate (exactly one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_AdjustableOrRelativeDate)

- The last day within an exercise period for a Bermudan or American style option. For a European style option it is the only day within the exercise period.

relevantUnderlyingDate (exactly one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_AdjustableOrRelativeDates)

- The date on the underlying set by the exercise of an option. What this date is depends on the option (eg in a swaption it is the effective date, in a extendible / cancelable provision is is the termination date).

earliestExerciseTime (exactly one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_BusinessCenterTime)

- The earliest time at which notice of exercise can be given by the buyer to the seller (or seller's agent) i) on the expiration date, in the case of a European style option, (ii) on each bermuda option exercise date and the expiration date, in the case of a Bermudan style option and (iii) all days that are exercise business days from and including the commencement date to, and including, the expiration date, in the case of an American style option.

latestExerciseTime (zero or one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_BusinessCenterTime)

- For a Bermudan or American style options, the latest time on an exercise business day (excluding the expiration date) within the exercise period that notice of exercise can be given by buyer to the seller or seller's agent. Notice of exercise given after this time will be deemed to have been given on the next exercise business day.

expirationTime (exactly one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_BusinessCenterTime)

- The latest time for expiration on expirationDate.

multipleExercise (zero or one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_MultipleExercise)

- As defined in the 2000 ISDA Definitions, Section 12.4. Multiple Exercise, the buyer of the option has the right to exercise all or less than all the unexercised notional amount of the underlying swap on one or more days in the exercise period, but on any such day may not exercise less than the minimum notional amount or more than the maximum notional amount, and if an integral multiple amount is specified, the notional amount exercised must be equal to, or be an integral multiple of, the integral multiple amount.

exerciseFeeSchedule (zero or one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_ExerciseFeeSchedule)

- The fees associated with an exercise date. The fees are conditional on the exercise occurring. The fees can be specified as actual currency amounts or as percentages of the notional amount being exercised.

Used by:

americanExercise

DTD Fragment:

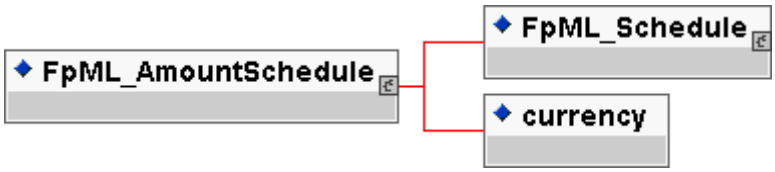
```
<!ENTITY % FpML_AmericanExercise "commencementDate , expirationDate ,
relevantUnderlyingDate , earliestExerciseTime , latestExerciseTime? ,
expirationTime , multipleExercise? , exerciseFeeSchedule?">
```

FpML_AmountSchedule

Description:

An entity for defining a currency amount or a currency amount schedule. This entity inherits from a base entity, FpML_Schedule.

Figure:



Contents:

inherited element(s) (this entity inherits the element(s) defined by exactly one occurrence of the entity FpML_Schedule)

- An entity for defining a schedule of rate or amounts in terms of an initial value and then a series of step date and value pairs. On each step date the rate or amount changes to the new step value. The series of step date and value pairs are optional. If not specified, this implies that the initial value remains unchanged over time.

currency (exactly one occurrence; of type *string*, an enumerated domain value defined by *currencyScheme*)

- The currency in which an amount is denominated.

Used by:

knownAmountSchedule
notionalStepSchedule

DTD Fragment:

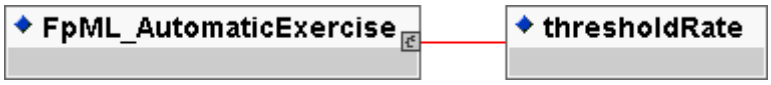
```
<!ENTITY % FpML_AmountSchedule "(%FpML_Schedule; , currency)">
```

FpML_AutomaticExercise

Description:

An entity to define automatic exercise of a swaption. With automatic exercise the option is deemed to have exercised if it is in hte money by more than the threshold amount on the exercise date.

Figure:



Contents:

thresholdRate (exactly one occurrence; of type *decimal*)

- A threshold rate. A threshold of 0.10% would be represented as 0.001.

Used by:

automaticExercise

DTD Fragment:

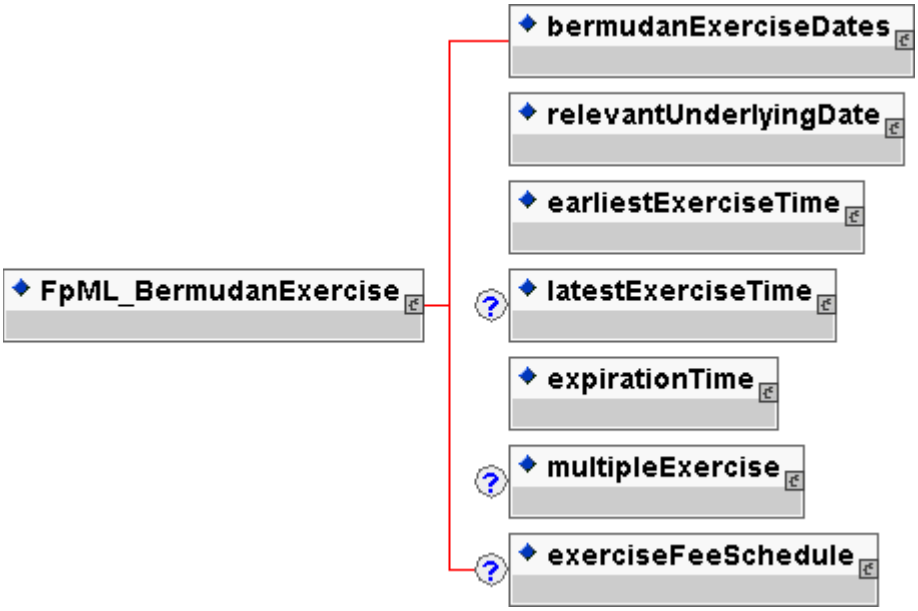
```
<!ENTITY % FpML_AutomaticExercise "thresholdRate">
```

FpML_BermudanExercise

Description:

An entity to define the bermudan option exercise dates and the expiration date together with any rules governing the notional amount of the underlying which can be exercised on any given exercise date and any associated exercise fees.

Figure:



Contents:

bermudanExerciseDates (exactly one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_AdjustableOrRelativeDates)

- The dates that define the bermudan option exercise dates and the expiration date. The last specified exercise date is assumed to be the expiration date. The dates can either be specified as a series of explicit dates and associated adjustments or as a series of dates defined relative to another schedule of dates, for example, the calculation period start dates. Where a relative series of dates are defined the first and last possible exercise dates can be separately specified.

relevantUnderlyingDate (exactly one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_AdjustableOrRelativeDates)

- The date on the underlying set by the exercise of an option. What this date is depends on the option (eg in a swaption it is the effective date, in a extendible / cancelable provision is is the termination date).

earliestExerciseTime (exactly one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_BusinessCenterTime)

- The earliest time at which notice of exercise can be given by the buyer to the seller (or seller's agent) i) on the expiration date, in the case of a European style option, (ii) on each bermuda option exercise date and the expiration date, in the case of a Bermudan style option and (iii) all days that are exercise business days from and including the commencement date to, and including, the expiration date, in the case of an American style option.

latestExerciseTime (zero or one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_BusinessCenterTime)

- For a Bermudan or American style options, the latest time on an exercise business day (excluding the expiration date) within the exercise period that notice of exercise can be given by buyer to the seller or seller's agent. Notice of exercise given after this time will be deemed to have been given on the next exercise business day.

expirationTime (exactly one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_BusinessCenterTime)

- The latest time for expiration on expirationDate.

multipleExercise (zero or one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_MultipleExercise)

- As defined in the 2000 ISDA Definitions, Section 12.4. Multiple Exercise, the buyer of the option has the right to exercise all or less than all the unexercised notional amount of the underlying swap on one or more days in the exercise period, but on any such day may not exercise less than the minimum notional amount or more than the maximum notional amount, and if an integral multiple amount is specified, the notional amount exercised must be equal to, or be an integral multiple of, the integral multiple amount.

exerciseFeeSchedule (zero or one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_ExerciseFeeSchedule)

- The fees associated with an exercise date. The fees are conditional on the exercise occurring. The fees can be specified as actual currency amounts or as percentages of the notional amount being exercised.

Used by:

bermudanExercise

DTD Fragment:

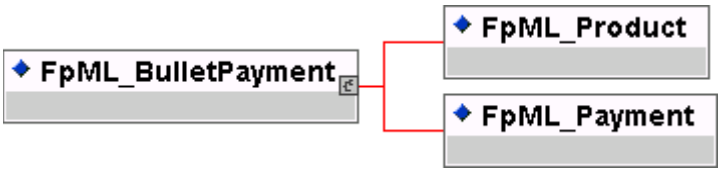
```
<!ENTITY % FpML_BermudanExercise "bermudanExerciseDates ,
relevantUnderlyingDate , earliestExerciseTime , latestExerciseTime? ,
expirationTime , multipleExercise? , exerciseFeeSchedule?">
```

FpML_BulletPayment

Description:

An product to represent a single cashflow.

Figure:



Contents:

inherited element(s) (this entity inherits the element(s) defined by exactly one occurrence of the entity FpML_Product)

- The base entity which all FpML products extend.

inherited element(s) (this entity inherits the element(s) defined by exactly one occurrence of the entity FpML_Payment)

- An entity for defining payments.

Used by:

bulletPayment

DTD Fragment:

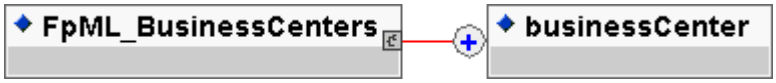
```
<!ENTITY %FpML_BulletPayment "%FpML_Product;,%FpML_Payment;">
```

FpML_BusinessCenters

Description:

An entity for defining financial business centers used in determining whether a day is a business day or not.

Figure:



Contents:

businessCenter (one or more occurrences; of type *string*, an enumerated domain value defined by *businessCenterScheme*)

- A code identifying a financial business center location. A list of business centers may be ordered in the document alphabetically based on business center code. An FpML document containing an unordered business center list is still regarded as a conformant document.

Used by:

businessCenters

DTD Fragment:

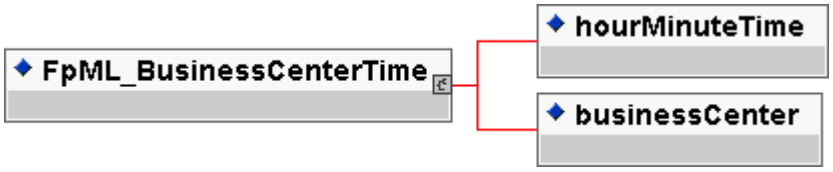
```
<!ENTITY % FpML_BusinessCenters "businessCenter+">
```

FpML_BusinessCenterTime

Description:

An entity for defining a time with respect to a business center location. For example, 11:00 am London time.

Figure:



Contents:

hourMinuteTime (exactly one occurrence; of type *time*)

- A time specified in hh:mm:ss format where the second component must be '00', e.g. 11am would be represented as 11:00:00.

businessCenter (exactly one occurrence; of type *string*, an enumerated domain value defined by *businessCenterScheme*)

- A code identifying a financial business center location. A list of business centers may be ordered in the document alphabetically based on business center code. An FpML document containing an unordered business center list is still regarded as a conformant document.

Used by:

cashSettlementValuationTime
earliestExerciseTime
expirationTime
fixingTime
latestExerciseTime

DTD Fragment:

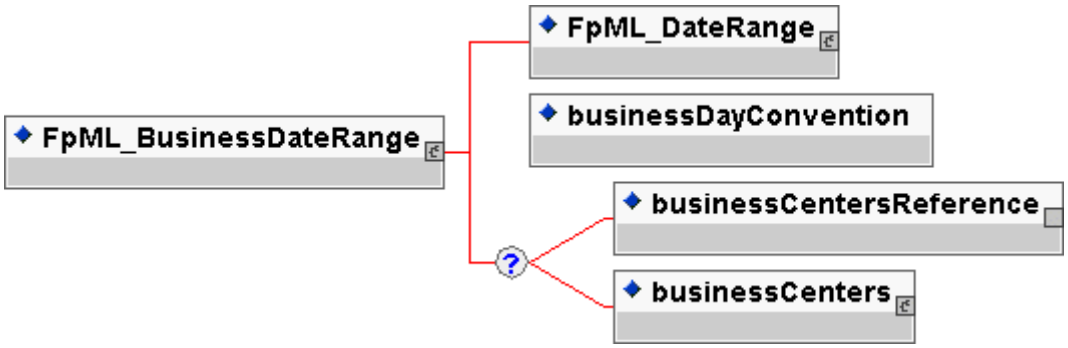
```
<!ENTITY % FpML_BusinessCenterTime "hourMinuteTime , businessCenter">
```

FpML_BusinessDateRange

Description:

An entity for defining a range of contiguous business days by defining an unadjusted first date, an unadjusted last date and a business day convention and business centers for adjusting the first and last dates if they would otherwise fall on a non business day in the specified business centers. The days between the first and last date must also be good business days in the specified business centers to be counted in the range. This entity inherits from the base entity, FpML_DateRange.

Figure:



Contents:

inherited element(s) (this entity inherits the element(s) defined by exactly one occurrence of the entity FpML_DateRange)

- A range of dates

businessDayConvention (exactly one occurrence; of type *string*, an enumerated domain value defined by *businessDayConventionScheme*)

- The convention for adjusting a date if it would otherwise fall on a day that is not a business day.

Zero or one occurrence of either

businessCentersReference (exactly one occurrence; an *empty* element containing an *href* attribute)

- A pointer style reference to a set of financial business centers defined elsewhere in the document. This set of business centers is used to determine whether a particular day is a business day or not.

Or

businessCenters (zero or one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_BusinessCenters)

- A container for a set of financial business centers. This set of business centers is used to determine whether a day is a business day or not.

Used by:

businessDateRange

DTD Fragment:

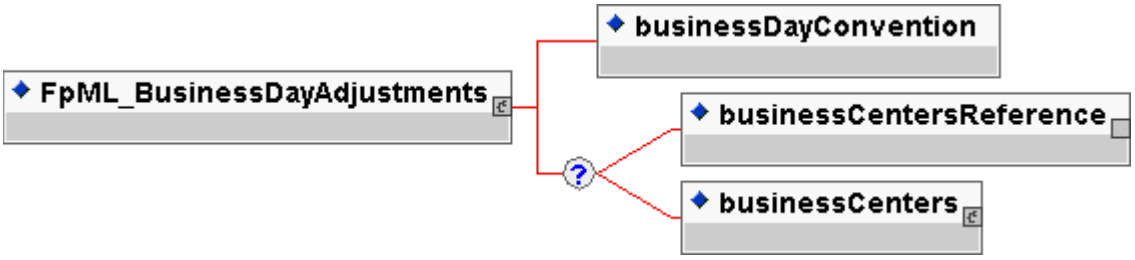
```
<!ENTITY % FpML_BusinessDateRange "(%FpML_DateRange; , businessDayConvention
, (businessCentersReference | businessCenters)?)">
```

FpML_BusinessDayAdjustments

Description:

An entity for defining the business day convention and financial business centers used for adjusting any relevant date if it would otherwise fall on a day that is not a business day in the specified business centers.

Figure:



Contents:

businessDayConvention (exactly one occurrence; of type *string*, an enumerated domain value defined by *businessDayConventionScheme*)

- The convention for adjusting a date if it would otherwise fall on a day that is not a business day. If the business day convention value is NONE then neither the businessCentersReference or businessCenters element should be included

Zero or one occurrence of either

businessCentersReference (exactly one occurrence; an *empty* element containing an *href* attribute)

- A pointer style reference to a set of financial business centers defined elsewhere in the document. This set of business centers is used to determine whether a particular day is a business day or not.

Or

businessCenters (zero or one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_BusinessCenters)

- A container for a set of financial business centers. This set of business centers is used to determine whether a day is a business day or not.

Used by:

calculationPeriodDatesAdjustments
dateAdjustments
paymentDatesAdjustments
resetDatesAdjustments

DTD Fragment:

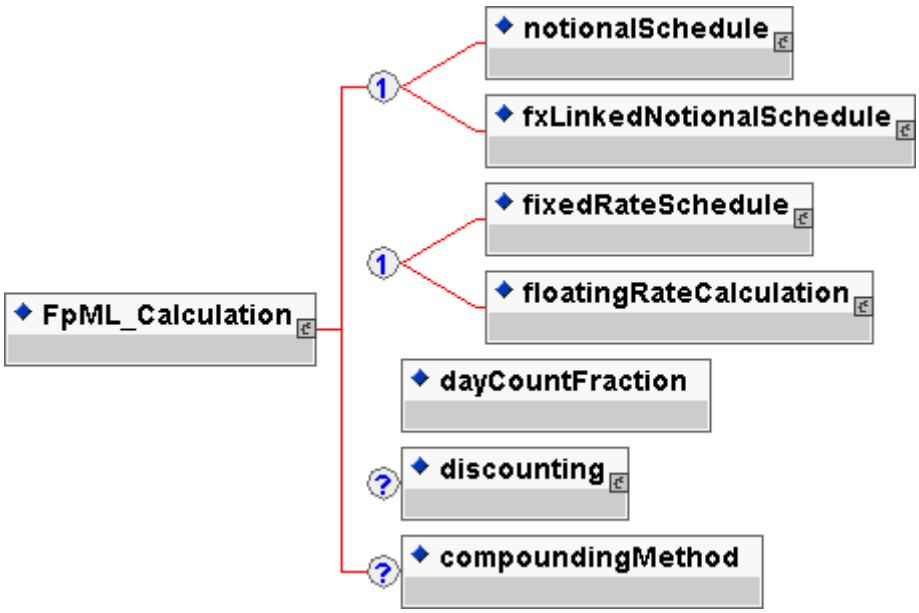
```
<!ENTITY % FpML_BusinessDayAdjustments "businessDayConvention ,
(businessCentersReference | businessCenters)?">
```

FpML_Calculation

Description:

An entity for defining the parameters used in the calculation of fixed or floating calculation period amounts.

Figure:



Contents:

Either

notionalSchedule (exactly one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_Notional)

- The notional amount or notional amount schedule.

Or

fxLinkedNotionalSchedule (exactly one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_FxLinkedNotionalSchedule)

- A notional amount schedule where each notional that applies to a calculation period is calculated with reference to a notional amount or notional amount schedule in a different currency by means of a spot currency exchange rate which is normally observed at the beginning of each period.

Either

fixedRateSchedule (exactly one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_Schedule)

- The fixed rate or fixed rate schedule expressed as explicit fixed rates and dates. In the case of a schedule, the step dates may be

subject to adjustment in accordance with any adjustments specified in calculationPeriodDatesAdjustments.

Or

floatingRateCalculation (exactly one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_FloatingRateCalculation)

- The floating rate calculation definitions.

dayCountFraction (exactly one occurrence; of type *string*, an enumerated domain value defined by *dayCountFractionScheme*)

- The day count fraction.

discounting (zero or one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_Discounting)

- The parameters specifying any discounting conventions that may apply. This element must only be included if discounting applies.

compoundingMethod (zero or one occurrence; of type *string*, an enumerated domain value defined by *compoundingMethodScheme*)

- If more than one calculation period contributes to a single payment amount this element specifies whether compounding is applicable, and if so, what compounding method is to be used. This element must only be included when more than one calculation period contributes to a single payment amount.

Used by:

calculation

DTD Fragment:

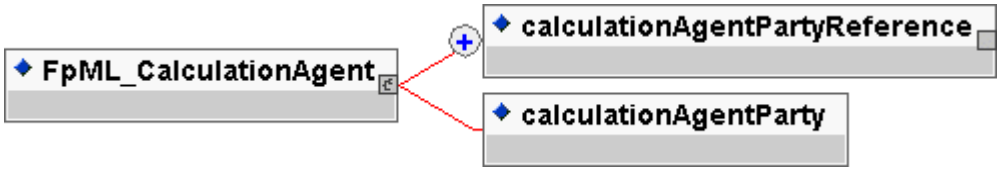
```
<!ENTITY % FpML_Calculation "((notionalSchedule | fxLinkedNotionalSchedule) ,
(fixedRateSchedule | floatingRateCalculation) , dayCountFraction ,
discounting? , compoundingMethod?)">
```

FpML_CalculationAgent

Description:

An entity for defining the ISDA Calculation Agent responsible for performing duties associated with the optional early termination on a swap transaction.

Figure:



Contents:

Either

calculationAgentPartyReference (one or more occurrences; an *empty* element containing an *href* attribute)

- A pointer style reference to a party identifier defined elsewhere in the document. The party referenced is the ISDA Calculation Agent for the trade. If more than one party is referenced then the parties are assumed to be co-calculation agents, i.e. they have joint responsibility.

Or

calculationAgentParty (exactly one occurrence; of type *string*, an enumerated domain value defined by *calculationAgentPartyScheme*)

- The ISDA Calculation Agent where the actual party responsible for performing the duties associated with an optional early termination provision will be determined at exercise. For example, the Calculation Agent may be defined as being the Non-exercising Party.

Used by:

calculationAgent

DTD Fragment:

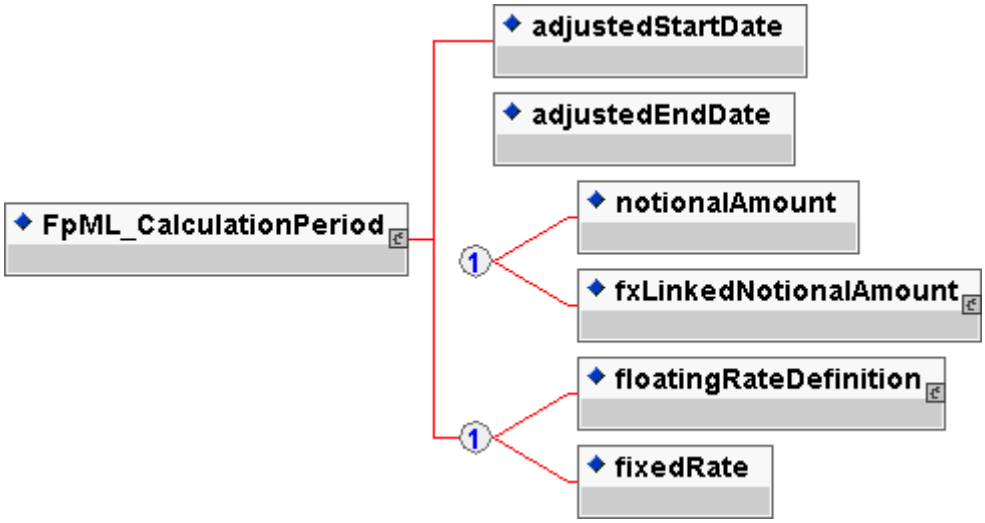
```
<!ENTITY % FpML_CalculationAgent "calculationAgentPartyReference+ |
calculationAgentParty">
```

FpML_CalculationPeriod

Description:

An entity for defining the parameters used in the calculation of a fixed or floating rate calculation period amount. This entity forms part of the cashflows representation of a swap stream.

Figure:



Contents:

adjustedStartDate (exactly one occurrence; of type *date*)

- The calculation period start date, adjusted according to any relevant business day convention.

adjustedEndDate (exactly one occurrence; of type *date*)

- The calculation period end date, adjusted according to any relevant business day convention.

Either

notionalAmount (exactly one occurrence; of type *decimal*)

- The calculation period notional amount.

Or

fxLinkedNotionalAmount (exactly one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_FxLinkedNotionalAmount)

- The amount that a cashflow will accrue interest on. This is the calculated amount of the fx linked notional - ie the other currency notional amount multiplied by the appropriate fx spot rate.

Either

floatingRateDefinition (exactly one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_FloatingRateDefinition)

- The floating rate reset information for the calculation period.

Or

fixedRate (exactly one occurrence; of type *decimal*)

- The calculation period fixed rate. A per annum rate, expressed as a decimal. A fixed rate of 5% would be represented as 0.05.

Used by:

calculationPeriod

DTD Fragment:

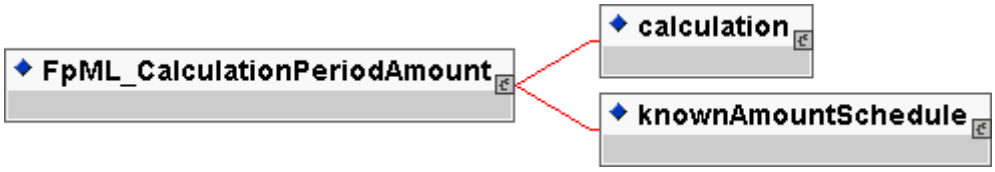
```
<!ENTITY % FpML_CalculationPeriod "adjustedStartDate , adjustedEndDate ,
(notionalAmount | fxLinkedNotionalAmount) , (floatingRateDefinition |
fixedRate)">
```

FpML_CalculationPeriodAmount

Description:

An entity for defining the parameters used in the calculation of fixed or floating rate calculation period amounts or for specifying a known calculation period amount or known amount schedule.

Figure:



Contents:

Either

calculation (exactly one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_Calculation)

- The parameters used in the calculation of fixed or floating rate calculation period amounts.

Or

knownAmountSchedule (exactly one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_AmountSchedule)

- The known calculation period amount or a known amount schedule expressed as explicit known amounts and dates. In the case of a schedule, the step dates may be subject to adjustment in accordance with any adjustments specified in calculationPeriodDatesAdjustments.

Used by:

calculationPeriodAmount

DTD Fragment:

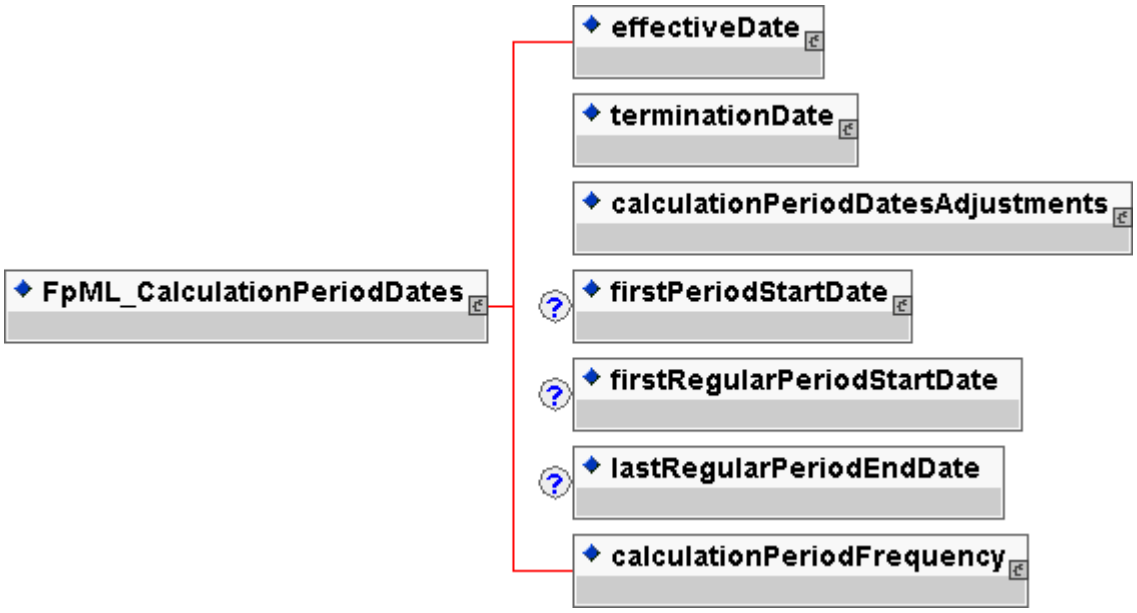
```
<!ENTITY % FpML_CalculationPeriodAmount "calculation | knownAmountSchedule">
```

FpML_CalculationPeriodDates

Description:

An entity for defining the parameters used to generate the calculation periods dates schedule, including the specification of any initial or final stub calculation periods. A calculation period schedule consists of an optional initial stub calculation period, one or more regular calculation periods and an optional final stub calculation period. In the absence of any initial or final stub calculation periods, the regular part of the calculation period schedule is assumed to be between the effective date and the termination date. No implicit stubs are allowed, i.e. stubs must be explicitly specified using an appropriate combination of firstPeriodStartDate, firstRegularPeriodStartDate and lastRegularPeriodEndDate.

Figure:



Contents:

effectiveDate (exactly one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_AdjustableDate)

- The first day of the term of the trade. This day may be subject to adjustment in accordance with a business day convention.

terminationDate (exactly one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_AdjustableDate)

- The last day of the term of the trade. This day may be subject to adjustment in accordance with a business day convention.

calculationPeriodDatesAdjustments (exactly one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_BusinessDayAdjustments)

- The business day convention to apply to each calculation period end date if it would otherwise fall on a day that is not a business day in the specified financial business centers.

firstPeriodStartDate (zero or one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_AdjustableDate)

- The start date of the first calculation period if the date falls before the effective date. It must only be specified if it is not equal to the effective date. This day may be subject to adjustment in accordance with a business day convention.

firstRegularPeriodStartDate (zero or one occurrence; of type *date*)

- The start date of the regular part of the calculation period schedule. It must only be specified if there is an initial stub calculation period. This day may be subject to adjustment in accordance with any adjustments specified in calculationPeriodDatesAdjustments.

lastRegularPeriodEndDate (zero or one occurrence; of type *date*)

- The end date of the regular part of the calculation period schedule. It must only be specified if there is a final stub calculation period. This day may be subject to adjustment in accordance with any adjustments specified in calculationPeriodDatesAdjustments.

calculationPeriodFrequency (exactly one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_CalculationPeriodFrequency)

- The frequency at which calculation period end dates occur within the regular part of the calculation period schedule and their roll date convention.

Used by:

calculationPeriodDates

DTD Fragment:

```
<!ENTITY % FpML_CalculationPeriodDates "effectiveDate , terminationDate ,
calculationPeriodDatesAdjustments , firstPeriodStartDate? ,
firstRegularPeriodStartDate? , lastRegularPeriodEndDate? ,
calculationPeriodFrequency">
```

FpML_CalculationPeriodFrequency

Description:

An entity for defining the frequency at which calculation period end dates occur within the regular part of the calculation period schedule and their roll date convention. This entity inherits from a base entity, FpML_Interval.

Figure:

Contents:

inherited element(s) (this entity inherits the element(s) defined by exactly one occurrence of the entity FpML_Interval)

- An entity for defining a time interval or offset, e.g. one day, three months. Used for specifying frequencies at which events occur, the tenor of a floating rate or an offset relative to another date.

rollConvention (exactly one occurrence; of type *string*, an enumerated domain value defined by *rollConventionScheme*)

- Used in conjunction with a frequency and the regular period start date of a calculation period, determines each calculation period end date within the regular part of a calculation period schedule.

Used by:

calculationPeriodFrequency

DTD Fragment:

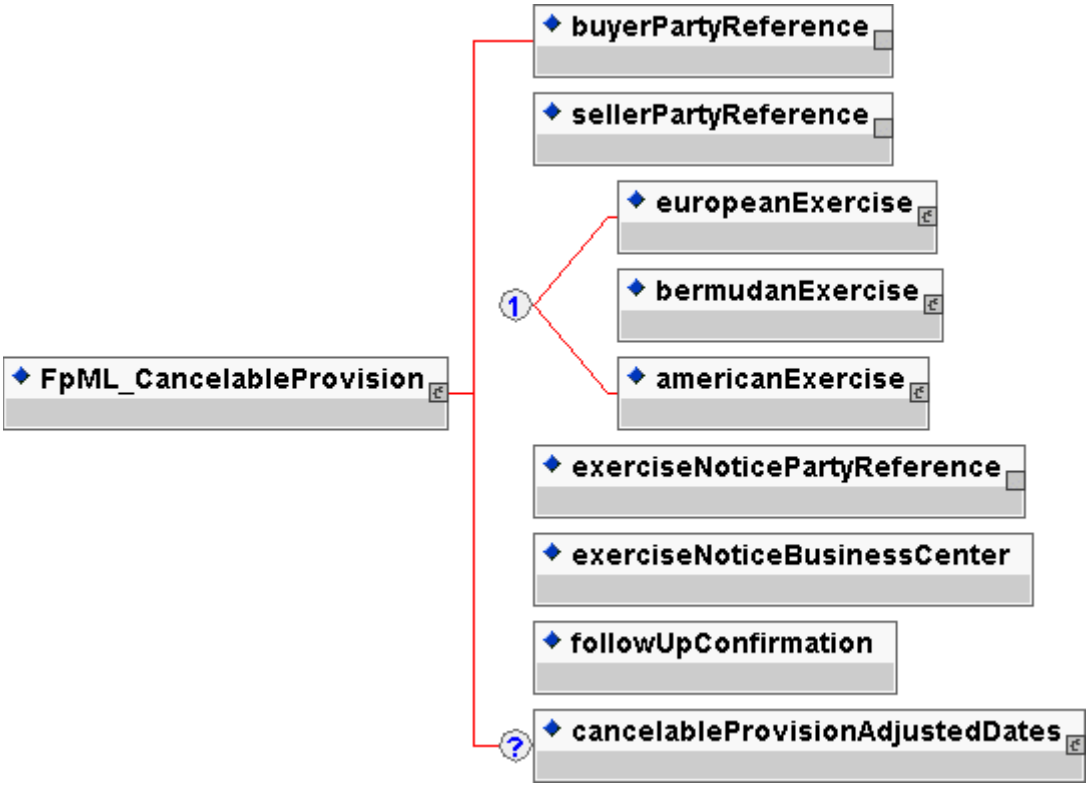
```
<!ENTITY % FpML_CalculationPeriodFrequency "(%FpML_Interval; ,
rollConvention)">
```

FpML_CancelableProvision

Description:

An entity to define the the right for a party to cancel a swap transaction on the specified exercise dates. This provision is for 'walkaway' cancellation (ie the fair value of the swap is not paid). A fee on to be paid on exercise can be specified.

Figure:



Contents:

buyerPartyReference (exactly one occurrence; an *empty* element containing an *href* attribute)

- A pointer style reference to a party identifier defined elsewhere in the document. The party referenced is the buyer of the instrument, also known as the fixed rate payer.

sellerPartyReference (exactly one occurrence; an *empty* element containing an *href* attribute)

- A pointer style reference to a party identifier defined elsewhere in the document. The party referenced is the seller of the instrument, also known as the floating rate payer.

Either

europeanExercise (exactly one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_EuropeanExercise)

- The parameters for defining the exercise period for a European style option together with any rules governing the notional amount of the underlying which can be exercised on any given exercise date and any associated exercise fees.

Or

bermudanExercise (exactly one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_BermudanExercise)

- The parameters for defining the exercise period for a Bermudan style option together with any rules governing the notional amount of the underlying which can be exercised on any given exercise date and any associated exercise fees.

Or

americanExercise (exactly one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_AmericanExercise)

- The parameters for defining the exercise period for an American style option together with any rules governing the notional amount of the underlying which can be exercised on any given exercise date and any associated exercise fees.

exerciseNoticePartyReference (exactly one occurrence; an *empty* element containing an *href* attribute)

- A pointer style reference to a party identifier defined elsewhere in the document. The party referenced is the party to which notice of exercise should be given by the buyer.

exerciseNoticeBusinessCenter (exactly one occurrence; of type *string*, an enumerated domain value defined by *businessCenterScheme*)

- The business center location where notice of exercise should be given to the seller or seller's agent.

followUpConfirmation (exactly one occurrence; of type *boolean*)

- A flag to indicate whether follow-up confirmation of exercise (written or electronic) is required following telephonic notice by the buyer to the seller or seller's agent.

cancelableProvisionAdjustedDates (zero or one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_CancelableProvisionAdjustedDates)

- The adjusted dates associated with a cancelable provision. These dates have been adjusted for any applicable business day convention.

Used by:

cancelableProvision

DTD Fragment:

```
<!ENTITY % FpML_CancelableProvision "buyerPartyReference ,
sellerPartyReference , (europeanExercise | bermudanExercise |
americanExercise) , exerciseNoticePartyReference ,
```

```
exerciseNoticeBusinessCenter , followUpConfirmation ,
cancelableProvisionAdjustedDates?">
```

FpML_CancelableProvisionAdjustedDates

Description:

An entity to define the adjusted dates for a cancelable provision.

Figure:

Contents:

cancellationEvent (one or more occurrences; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_CancellationEvent)

- The adjusted dates for an individual cancellation date.

Used by:

cancelableProvisionAdjustedDates

DTD Fragment:

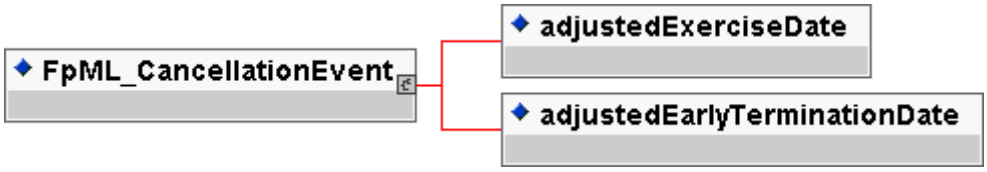
```
<!ENTITY % FpML_CancelableProvisionAdjustedDates "cancellationEvent+">
```

FpML_CancellationEvent

Description:

The adjusted dates for a specific cancellation date - this includes the adjusted exercise date and adjusted termination date

Figure:



Contents:

adjustedExerciseDate (exactly one occurrence; of type *date*)

- The date on which option exercise takes place. This date should already be adjusted for any applicable business day convention.

adjustedEarlyTerminationDate (exactly one occurrence; of type *date*)

- The early termination date that is applicable if an early termination provision is exercised. This date should already be adjusted for any applicable business day convention.

Used by:

cancellationEvent

DTD Fragment:

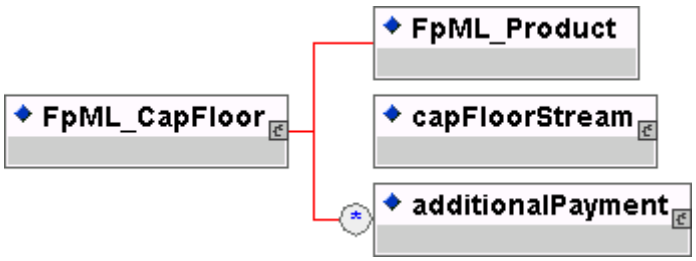
```
<!ENTITY % FpML_CancellationEvent "adjustedExerciseDate ,
adjustedEarlyTerminationDate">
```

FpML_CapFloor

Description:

An entity to describe and cap floor product. This includes a capFloorStream and zero or more additional payments.

Figure:



Contents:

inherited element(s) (this entity inherits the element(s) defined by exactly one occurrence of the entity FpML_Product)

- The base entity which all FpML products extend.

capFloorStream (exactly one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_InterestRateStream)

- A cap, floor or cap floor structure stream.

additionalPayment (zero or more occurrences; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_Fee)

- Additional payments between the principal parties involved in the swap.

Used by:

capFloor

DTD Fragment:

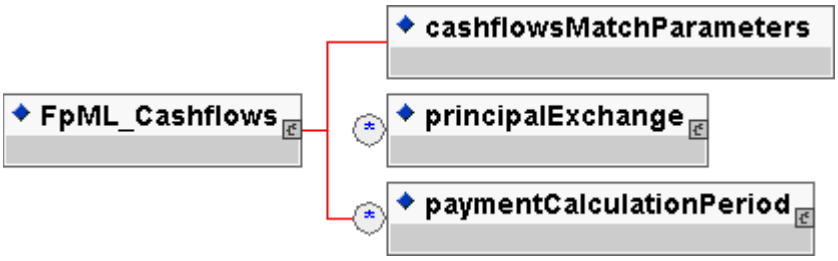
```
<!ENTITY % FpML_CapFloor "%FpML_Product;,capFloorStream ,
additionalPayment*">
```

FpML_Cashflows

Description:

An entity for defining the cashflow representation of a swap trade.

Figure:



Contents:

cashflowsMatchParameters (exactly one occurrence; of type *boolean*)

- A true/false flag to indicate whether the cashflows match the parametric definition of the stream, i.e. whether the cashflows could be regenerated from the parameters without loss of information.

principalExchange (zero or more occurrences; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_PrincipalExchange)

- The initial, intermediate and final principal exchange amounts. Typically required on cross currency interest rate swaps where actual exchanges of principal occur. A list of principal exchange elements may be ordered in the document by ascending adjusted principal exchange date. An FpML document containing an unordered principal exchange list is still regarded as a conformant document.

paymentCalculationPeriod (zero or more occurrences; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_PaymentCalculationPeriod)

- The adjusted payment date and associated calculation period parameters required to calculate the actual or projected payment amount. A list of payment calculation period elements may be ordered in the document by ascending adjusted payment date. An FpML document containing an unordered list of payment calculation periods is still regarded as a conformant document.

Used by:

cashflows

DTD Fragment:

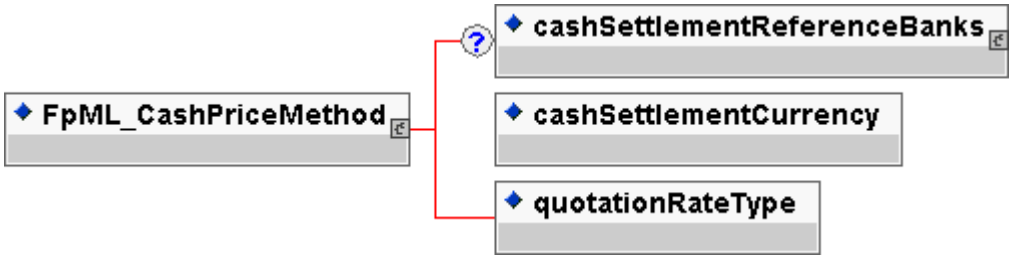
```
<!ENTITY % FpML_Cashflows "cashflowsMatchParameters , principalExchange* ,
paymentCalculationPeriod*">
```

FpML_CashPriceMethod

Description:

An entity to define the parameters necessary for each of the ISDA defined cash price methods for cash settlement.

Figure:



Contents:

cashSettlementReferenceBanks (zero or one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_CashSettlementReferenceBanks)

- A container for a set of reference institutions. These reference institutions may be called upon to provide rate quotations as part of the method to determine the applicable cash settlement amount.

cashSettlementCurrency (exactly one occurrence; of type *string*, an enumerated domain value defined by *currencyScheme*)

- The currency in which the cash settlement amount will be specified.

quotationRateType (exactly one occurrence; of type *string*, an enumerated domain value defined by *quotationRateTypeScheme*)

- Which rate quote is to be observed, either Bid, Mid, Offer or Exercising Party Pays. The meaning of Exercising Party Pays is defined in the 2000 ISDA Definitions, Section 17.2. Certain Definitions Relating to Cash Settlement, paragraph (j)

Used by:

cashPriceAlternateMethod
cashPriceMethod

DTD Fragment:

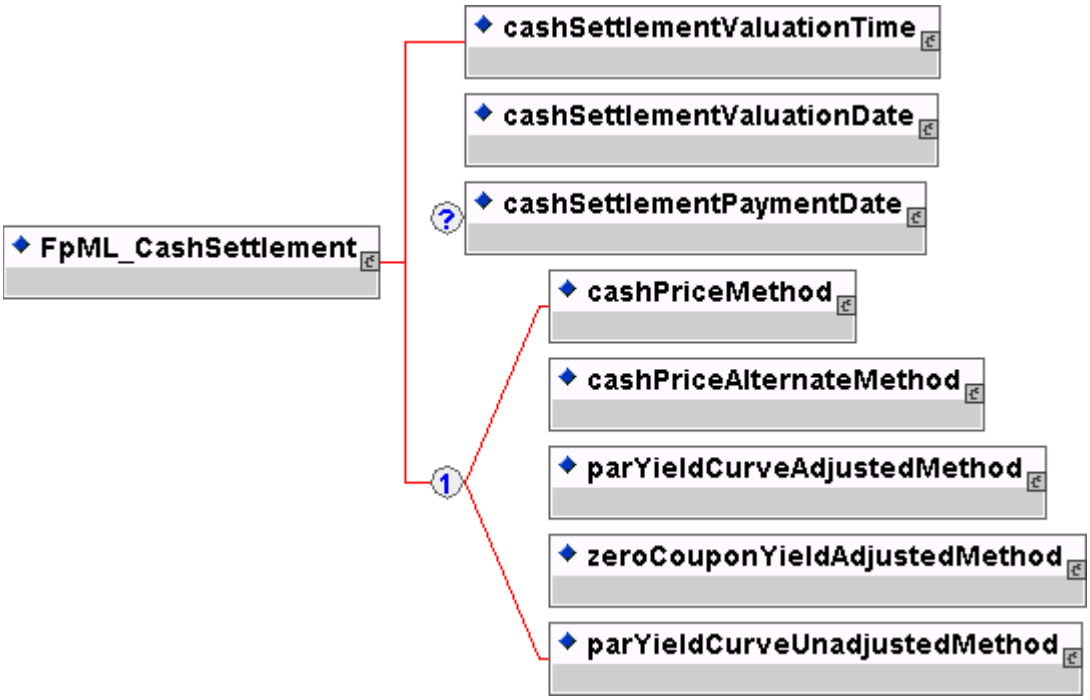
```
<!ENTITY % FpML_CashPriceMethod "cashSettlementReferenceBanks? ,
cashSettlementCurrency , quotationRateType">
```

FpML_CashSettlement

Description:

An entity to define the cash settlement of a swaption.

Figure:



Contents:

cashSettlementValuationTime (exactly one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_BusinessCenterTime)

- The time on the cash settlement valuation date when the cash settlement amount will be determined according to the cash settlement method if the parties have not otherwise been able to agree the cash settlement amount.

cashSettlementValuationDate (exactly one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_RelativeDateOffset)

- The date on which the cash settlement amount will be determined according to the cash settlement method if the parties have not otherwise been able to agree the cash settlement amount.

cashSettlementPaymentDate (exactly one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_CashSettlementPaymentDate)

- The date on which the cash settlement amount will be paid, subject to adjustment in accordance with any applicable business day convention. This element would not be present for a mandatory early termination

provision where the cash settlement payment date is the mandatory early termination date.

Either

cashPriceMethod (exactly one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_CashPriceMethod)

- An ISDA defined cash settlement method used for the determination of the applicable cash settlement amount. The method is defined in the 2000 ISDA Definitions, Section 17.3. Cash Settlement Methods, paragraph (a).

Or

cashPriceAlternateMethod (exactly one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_CashPriceMethod)

- An ISDA defined cash settlement method used for the determination of the applicable cash settlement amount. The method is defined in the 2000 ISDA Definitions, Section 17.3. Cash Settlement Methods, paragraph (b).

Or

parYieldCurveAdjustedMethod (exactly one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_YieldCurveMethod)

- An ISDA defined cash settlement method used for the determination of the applicable cash settlement amount. The method is defined in the 2000 ISDA Definitions, Section 17.3. Cash Settlement Methods, paragraph (c).

Or

zeroCouponYieldAdjustedMethod (exactly one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_YieldCurveMethod)

- An ISDA defined cash settlement method used for the determination of the applicable cash settlement amount. The method is defined in the 2000 ISDA Definitions, Section 17.3. Cash Settlement Methods, paragraph (d).

Or

parYieldCurveUnadjustedMethod (exactly one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_YieldCurveMethod)

- An ISDA defined cash settlement method used for the determination of the applicable cash settlement amount. The method is defined in the 2000 ISDA Definitions, Section 17.3. Cash Settlement Methods, paragraph (e).

Used by:

cashSettlement

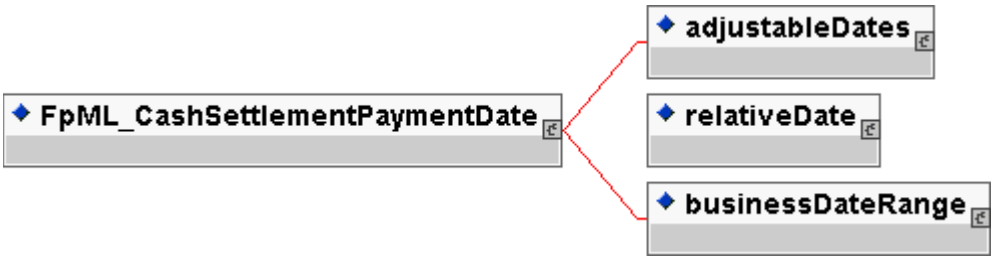
DTD Fragment:

```
<!ENTITY % FpML_CashSettlement "cashSettlementValuationTime ,
cashSettlementValuationDate , cashSettlementPaymentDate , (cashPriceMethod |
cashPriceAlternateMethod | parYieldCurveAdjustedMethod |
zeroCouponYieldAdjustedMethod | parYieldCurveUnadjustedMethod)">
```

FpML_CashSettlementPaymentDate

Description:

Figure:



Contents:

Either

adjustableDates (exactly one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_AdjustableDates)

- A series of dates that shall be subject to adjustment if they would otherwise fall on a day that is not a business day in the specified business centers, together with the convention for adjusting the date.

Or

relativeDate (exactly one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_RelativeDateOffset)

- A date specified as some offset to another date (the anchor date).

Or

businessDateRange (exactly one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_BusinessDateRange)

- A range of contiguous business days.

Used by:

cashSettlementPaymentDate

DTD Fragment:

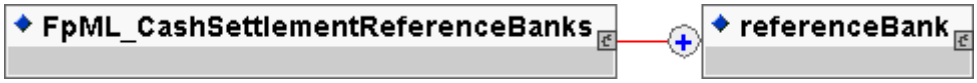
```
<!ENTITY % FpML_CashSettlementPaymentDate "adjustableDates | relativeDate |
businessDateRange">
```

FpML_CashSettlementReferenceBanks

Description:

An entity to define the list of reference banks to be used in agreeing the cash settlement value for an exercised swaption.

Figure:



Contents:

referenceBank (one or more occurrences; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_ReferenceBank)

- An institution (party) identified by means of a coding scheme and an optional name.

Used by:

cashSettlementReferenceBanks

DTD Fragment:

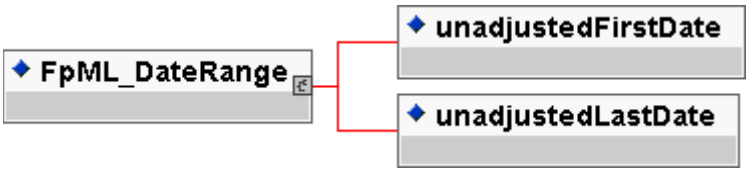
```
<!ENTITY % FpML_CashSettlementReferenceBanks "referenceBank+">
```

FpML_DateRange

Description:

A range of dates

Figure:



Contents:

unadjustedFirstDate (exactly one occurrence; of type *date*)

- The first date of a date range.

unadjustedLastDate (exactly one occurrence; of type *date*)

- The last date of a date range.

Used by:

FpML_BusinessDateRange
scheduleBounds

DTD Fragment:

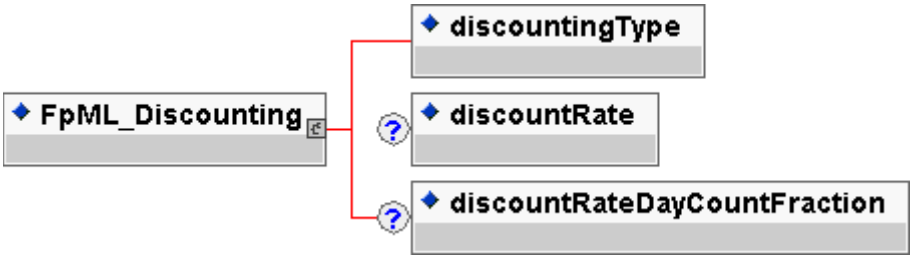
```
<!ENTITY % FpML_DateRange "unadjustedFirstDate , unadjustedLastDate">
```

FpML_Discounting

Description:

An entity for defining discounting information. The 2000 ISDA Definitions, Section 8.4. Discounting (related to the calculation of a discounted fixed amount or floating amount) apply. This entity must only be included if discounting applies.

Figure:



Contents:

discountingType (exactly one occurrence; of type *string*, an enumerated domain value defined by *discountingTypeScheme*)

- The discounting method that is applicable.

discountRate (zero or one occurrence; of type *decimal*)

- A discount rate, expressed as a decimal, to be used in the calculation of a discounted amount. A discount rate of 5% would be represented as 0.05.

discountRateDayCountFraction (zero or one occurrence; of type *string*, an enumerated domain value defined by *dayCountFractionScheme*)

- A discount day count fraction to be used in the calculation of a discounted amount.

Used by:

discounting

DTD Fragment:

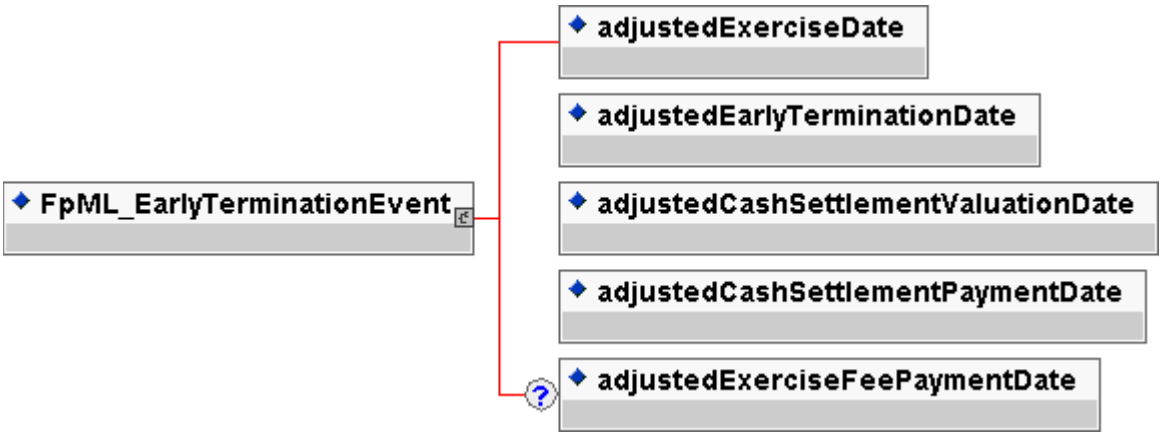
```
<!ENTITY % FpML_Discounting "discountingType , discountRate? ,
discountRateDayCountFraction?">
```

FpML_EarlyTerminationEvent

Description:

An entity to define the adjusted dates associated with an early termination provision.

Figure:



Contents:

adjustedExerciseDate (exactly one occurrence; of type *date*)

- The date on which option exercise takes place. This date should already be adjusted for any applicable business day convention.

adjustedEarlyTerminationDate (exactly one occurrence; of type *date*)

- The early termination date that is applicable if an early termination provision is exercised. This date should already be adjusted for any applicable business day convention.

adjustedCashSettlementValuationDate (exactly one occurrence; of type *date*)

- The date by which the cash settlement amount must be agreed. This date should already be adjusted for any applicable business day convention.

adjustedCashSettlementPaymentDate (exactly one occurrence; of type *date*)

- The date on which the cash settlement amount is paid. This date should already be adjusted for any applicable business day convention.

adjustedExerciseFeePaymentDate (zero or one occurrence; of type *date*)

- The date on which the exercise fee amount is paid. This date should already be adjusted for any applicable business day convention.

Used by:

earlyTerminationEvent

DTD Fragment:

```
<!ENTITY % FpML_EarlyTerminationEvent "adjustedExerciseDate ,
```

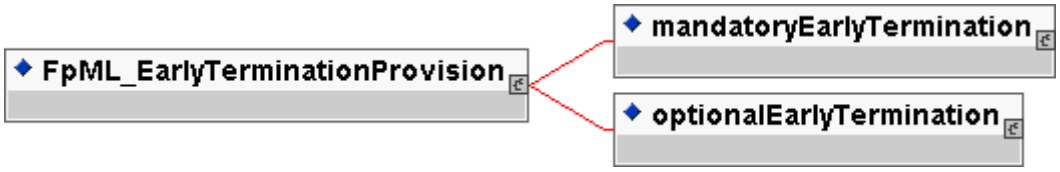
```
adjustedEarlyTerminationDate , adjustedCashSettlementValuationDate ,
adjustedCashSettlementPaymentDate , adjustedExerciseFeePaymentDate?">
```

FpML_EarlyTerminationProvision

Description:

An entity to define an early termination provision for a product. This early termination is at fair value, ie on termination the fair value of the product must be settled between the parties.

Figure:



Contents:

Either

mandatoryEarlyTermination (exactly one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_MandatoryEarlyTermination)

- A mandatory early termination provision to terminate the trade at fair value.

Or

optionalEarlyTermination (exactly one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_OptionalEarlyTermination)

- The option for either or both parties to terminate the swap at fair value.

Used by:

earlyTerminationProvision

DTD Fragment:

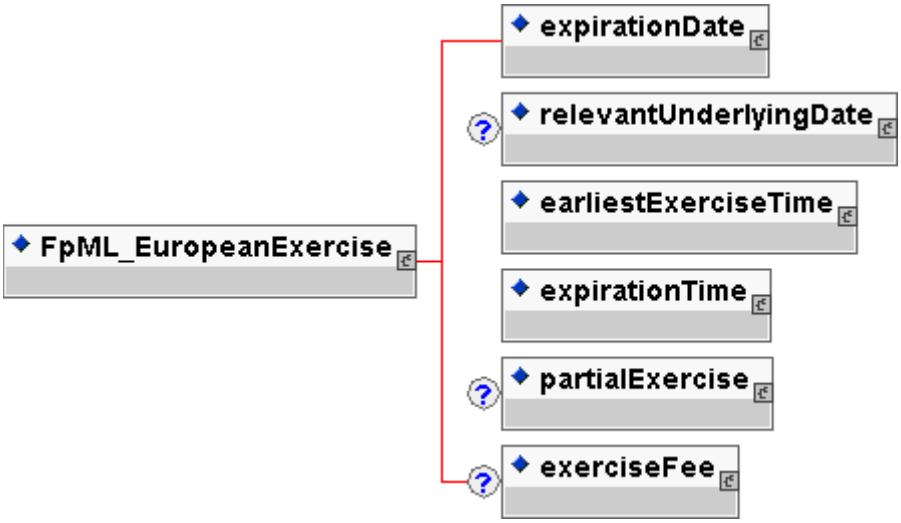
```
<!ENTITY % FpML_EarlyTerminationProvision "mandatoryEarlyTermination |
optionalEarlyTermination">
```

FpML_EuropeanExercise

Description:

An entity to define the exercise period for a European style option together with any rules governing the notional amount of the underlying which can be exercised on any given exercise date and any associated exercise fees.

Figure:



Contents:

expirationDate (exactly one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_AdjustableOrRelativeDate)

- The last day within an exercise period for a Bermudan or American style option. For a European style option it is the only day within the exercise period.

relevantUnderlyingDate (zero or one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_AdjustableOrRelativeDates)

- The date on the underlying set by the exercise of an option. What this date is depends on the option (eg in a swaption it is the effective date, in a extendible / cancelable provision is is the termination date).

earliestExerciseTime (exactly one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_BusinessCenterTime)

- The earliest time at which notice of exercise can be given by the buyer to the seller (or seller's agent) i) on the expiration date, in the case of a European style option, (ii) on each bermuda option exercise date and the expiration date, in the case of a Bermudan style option and (iii) all days that are exercise business days from and including the commencement date to, and including, the expiration date, in the case of an American style option.

expirationTime (exactly one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_BusinessCenterTime)

- The latest time for expiration on expirationDate.

partialExercise (zero or one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_PartialExercise)

- As defined in the 2000 ISDA Definitions, Section 12.3. Partial Exercise, the buyer of the option has the right to exercise all or less than all the notional amount of the underlying swap on the expiration date, but may not exercise less than the minimum notional amount, and if an integral multiple amount is specified, the notional amount exercised must be equal to, or be an integral multiple of, the integral multiple amount.

exerciseFee (zero or one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_ExerciseFee)

- A fee to be paid on exercise. This could be represented as an amount or a rate and notional reference on which to apply the rate.

Used by:

europeanExercise

DTD Fragment:

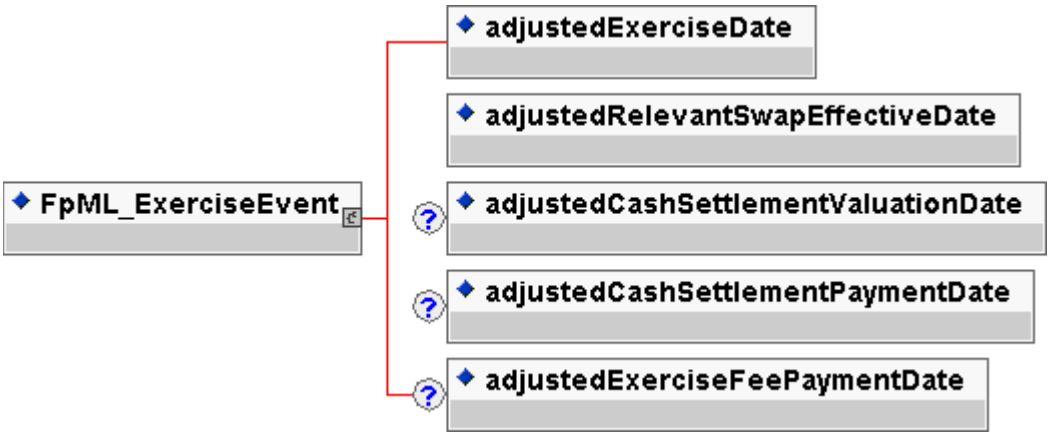
```
<!ENTITY % FpML_EuropeanExercise "expirationDate , relevantUnderlyingDate? ,
earliestExerciseTime , expirationTime , partialExercise? , exerciseFee?">
```

FpML_ExerciseEvent

Description:

An entity to define the adjusted dates associated with a particular excercise event.

Figure:



Contents:

adjustedExerciseDate (exactly one occurrence; of type *date*)

- The date on which option exercise takes place. This date should already be adjusted for any applicable business day convention.

adjustedRelevantSwapEffectiveDate (exactly one occurrence; of type *date*)

- The effective date of the underlying swap associated with a given exercise date. This date should already be adjusted for any applicable business day convention.

adjustedCashSettlementValuationDate (zero or one occurrence; of type *date*)

- The date by which the cash settlement amount must be agreed. This date should already be adjusted for any applicable business day convention.

adjustedCashSettlementPaymentDate (zero or one occurrence; of type *date*)

- The date on which the cash settlement amount is paid. This date should already be adjusted for any applicable business day convention.

adjustedExerciseFeePaymentDate (zero or one occurrence; of type *date*)

- The date on which the exercise fee amount is paid. This date should already be adjusted for any applicable business day convention.

Used by:

exerciseEvent

DTD Fragment:

```
<!ENTITY % FpML_ExerciseEvent "adjustedExerciseDate ,
```

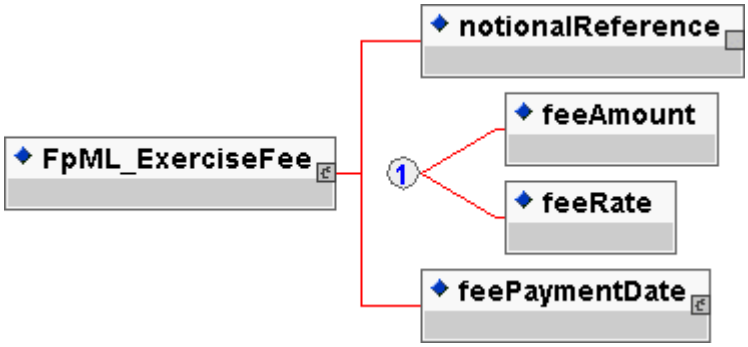
```
adjustedRelevantSwapEffectiveDate , adjustedCashSettlementValuationDate? ,
adjustedCashSettlementPaymentDate? , adjustedExerciseFeePaymentDate?">
```

FpML_ExerciseFee

Description:

An entity to define a fee to be payable on exercise of an option. This fee may be defined as an amount or a percentage of the notional exercised.

Figure:



Contents:

notionalReference (exactly one occurrence; an *empty* element containing an *href* attribute)

- A pointer style reference to the associated notional schedule defined elsewhere in the document.

Either

feeAmount (exactly one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity *decimal*)

- The amount of fee to be paid on exercise. The currency of this fee is the currency of the referenced notional

Or

feeRate (exactly one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity *decimal*)

- A fee represented as a percentage of some referenced notional

feePaymentDate (exactly one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_RelativeDateOffset)

- The date on which exercise fees will be paid. It can be specified as a reference date or a relative date.

Used by:

exerciseFee

DTD Fragment:

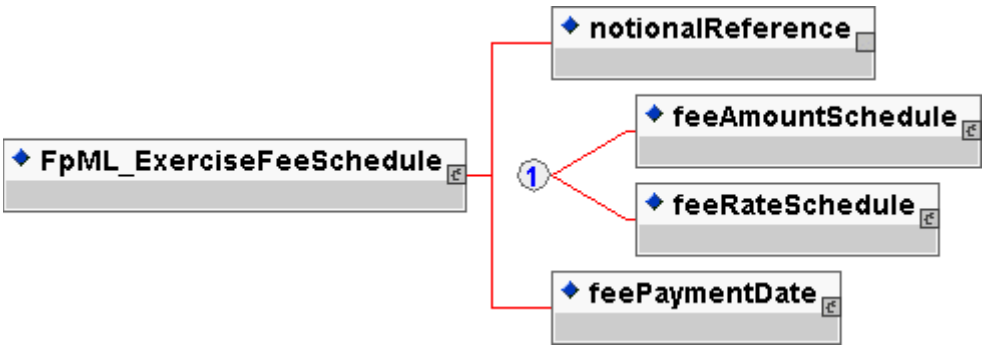
```
<!ENTITY % FpML_ExerciseFee "notionalReference , (feeAmount | feeRate) ,
feePaymentDate">
```

FpML_ExerciseFeeSchedule

Description:

An entity to define a fee or schedule of fees to be payable on exercise of an option. This fee may be defined as an amount or a percentage of the notional exercised.

Figure:



Contents:

notionalReference (exactly one occurrence; an *empty* element containing an *href* attribute)

- A pointer style reference to the associated notional schedule defined elsewhere in the document.

Either

feeAmountSchedule (exactly one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_AmountSchedule)

- The exercise fee amount schedule. The fees are expressed as currency amounts. The currency of the fee is assumed to be that of the notional schedule referenced.

Or

feeRateSchedule (exactly one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_Schedule)

- The exercise fee rate schedule. The fees are expressed as percentage rates of the notional being exercised. The currency of the fee is assumed to be that of the notional schedule referenced.

feePaymentDate (exactly one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_RelativeDateOffset)

- The date on which exercise fees will be paid. It can be specified as a reference date or a relative date.

Used by:

exerciseFeeSchedule

DTD Fragment:

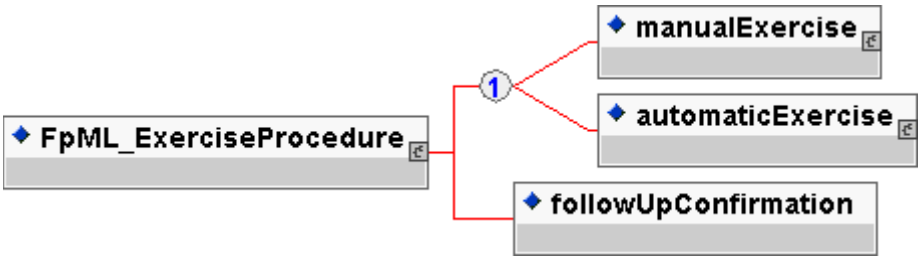
```
<!ENTITY % FpML_ExerciseFeeSchedule "notionalReference , (feeAmountSchedule |
feeRateSchedule) , feePaymentDate">
```

FpML_ExerciseProcedure

Description:

An entity to describe how notice of exercise should be given. This can either be manual or automatic.

Figure:



Contents:

Either

manualExercise (exactly one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_ManualExercise)

- Specifies that the notice of exercise must be given by the buyer to the seller or seller's agent.

Or

automaticExercise (exactly one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_AutomaticExercise)

- If automatic exercise is specified then the notional amount of the underlying swap, not previously exercised under the swaption, will be automatically exercised at the expiration time on the expiration date if at such time the buyer is in-the-money, provided that the difference between the settlement rate and the fixed rate under the relevant underlying swap is not less than the specified thresholdRate. The term In-the-money is assumed to have the meaning defined in the 2000 ISDA Definitions, Section 17.4. In-the-money.

followUpConfirmation (exactly one occurrence; of type *boolean*)

- A flag to indicate whether follow-up confirmation of exercise (written or electronic) is required following telephonic notice by the buyer to the seller or seller's agent.

Used by:

exerciseProcedure

DTD Fragment:

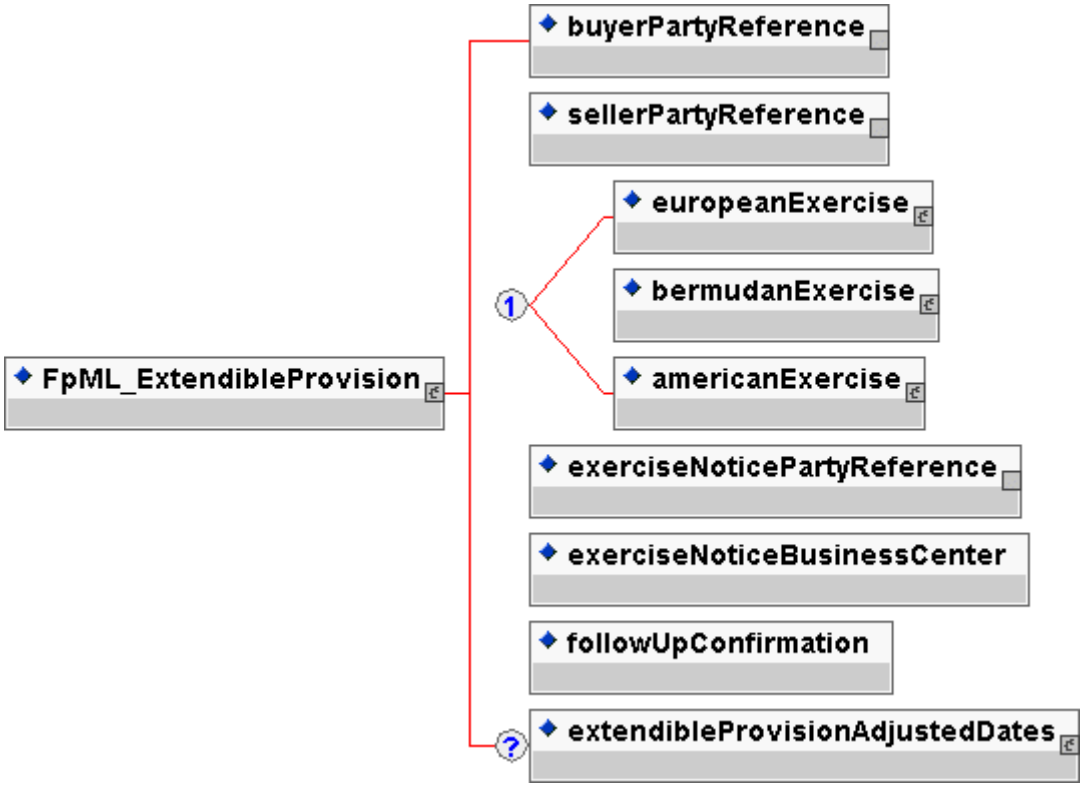
```
<!ENTITY % FpML_ExerciseProcedure "(manualExercise | automaticExercise) ,
followUpConfirmation">
```

FpML_ExtendibleProvision

Description:

An entity to define the the right for a party to extend a swap transaction on the specified exercise dates to a specified new termination date.

Figure:



Contents:

buyerPartyReference (exactly one occurrence; an *empty* element containing an *href* attribute)

- A pointer style reference to a party identifier defined elsewhere in the document. The party referenced is the buyer of the instrument, also known as the fixed rate payer.

sellerPartyReference (exactly one occurrence; an *empty* element containing an *href* attribute)

- A pointer style reference to a party identifier defined elsewhere in the document. The party referenced is the seller of the instrument, also known as the floating rate payer.

Either

europeanExercise (exactly one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_EuropeanExercise)

- The parameters for defining the exercise period for a European style option together with any rules governing the notional amount of the underlying which can be exercised on any given exercise date and any associated exercise fees.

Or

bermudanExercise (exactly one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_BermudanExercise)

- The parameters for defining the exercise period for a Bermudan style option together with any rules governing the notional amount of the underlying which can be exercised on any given exercise date and any associated exercise fees.

Or

americanExercise (exactly one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_AmericanExercise)

- The parameters for defining the exercise period for an American style option together with any rules governing the notional amount of the underlying which can be exercised on any given exercise date and any associated exercise fees.

exerciseNoticePartyReference (exactly one occurrence; an *empty* element containing an *href* attribute)

- A pointer style reference to a party identifier defined elsewhere in the document. The party referenced is the party to which notice of exercise should be given by the buyer.

exerciseNoticeBusinessCenter (exactly one occurrence; of type *string*, an enumerated domain value defined by *businessCenterScheme*)

- The business center location where notice of exercise should be given to the seller or seller's agent.

followUpConfirmation (exactly one occurrence; of type *boolean*)

- A flag to indicate whether follow-up confirmation of exercise (written or electronic) is required following telephonic notice by the buyer to the seller or seller's agent.

extendibleProvisionAdjustedDates (zero or one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_ExtendibleProvisionAdjustedDates)

- The adjusted dates associated with a extendible provision. These dates have been adjusted for any applicable business day convention.

Used by:

extendibleProvision

DTD Fragment:

```
<!ENTITY % FpML_ExtendibleProvision "buyerPartyReference ,
sellerPartyReference , (europeanExercise | bermudanExercise |
americanExercise) , exerciseNoticePartyReference ,
```

```
exerciseNoticeBusinessCenter , followUpConfirmation ,
extendibleProvisionAdjustedDates?">
```

FpML_ExtendibleProvisionAdjustedDates

Description:

AN entity to define the adjusted dates associated with a provision to extend a swap.

Figure:

Contents:

extensionEvent (one or more occurrences; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_ExtensionEvent)

- The adjusted dates associated with a single extendible exercise date.

Used by:

extendibleProvisionAdjustedDates

DTD Fragment:

```
<!ENTITY % FpML_ExtendibleProvisionAdjustedDates "extensionEvent+">
```

FpML_ExtensionEvent

Description:

An entity to define the adjusted dates associated with an individual extension event.

Figure:

Contents:

adjustedExerciseDate (exactly one occurrence; of type *date*)

- The date on which option exercise takes place. This date should already be adjusted for any applicable business day convention.

adjustedExtendedTerminationDate (exactly one occurrence; of type *date*)

- The termination date if an extendible provision is exercised. This date should already be adjusted for any applicable business day convention.

Used by:

extensionEvent

DTD Fragment:

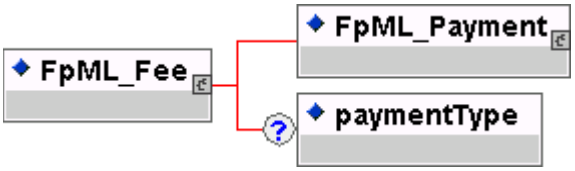
```
<!ENTITY % FpML_ExtensionEvent "adjustedExerciseDate ,
adjustedExtendedTerminationDate">
```

FpML_Fee

Description:

An entity for defining additional payments associated with a trade which are not defined as part of the stream payments. It may be used to define additional payments between the principal parties involved in the trade or other third parties such as a broker.

Figure:



Contents:

inherited element(s) (this entity inherits the element(s) defined by exactly one occurrence of the entity FpML_Payment)

- An entity for defining payments.

paymentType (zero or one occurrence; of type *string*, an enumerated domain value defined by *paymentTypeScheme*)

- A classification of the type of fee or additional payment, e.g. brokerage, upfront fee etc. FpML does not define domain values for this element.

Used by:

additionalPayment
otherPartyPayment

DTD Fragment:

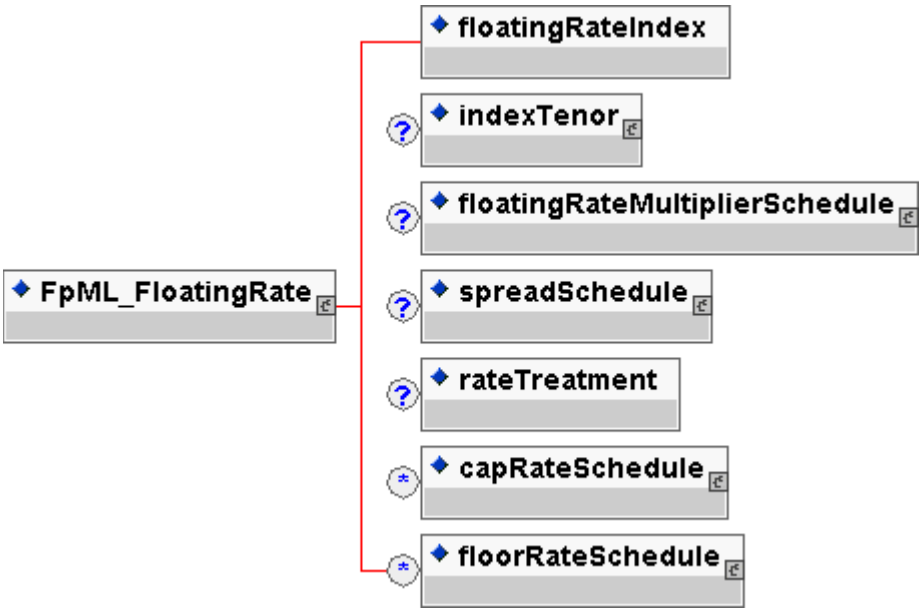
```
<!ENTITY % FpML_Fee "%FpML_Payment; , paymentType?">
```

FpML_FloatingRate

Description:

An entity for defining the floating rate definitions.

Figure:



Contents:

floatingRateIndex (exactly one occurrence; of type *string*, an enumerated domain value defined by *floatingRateIndexScheme*)

- The ISDA Floating Rate Option, i.e. the floating rate index.

indexTenor (zero or one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_Interval)

- The ISDA Designated Maturity, i.e. the tenor of the floating rate.

floatingRateMultiplierSchedule (zero or one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_Schedule)

- A schedule of values by which the floating rate will be multiplied by.

spreadSchedule (zero or one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_Schedule)

- The ISDA Spread or a Spread schedule expressed as explicit spreads and dates. In the case of a schedule, the step dates may be subject to adjustment in accordance with any adjustments specified in calculationPeriodDatesAdjustments. The spread is a per annum rate, expressed as a decimal. For purposes of determining a calculation period amount, if positive the spread will be added to the floating rate and if negative the spread will be subtracted from the floating rate. A positive 10 basis point (0.1%) spread would be represented as 0.001.

rateTreatment (zero or one occurrence; of type *string*, an enumerated domain value defined by *rateTreatmentScheme*)

- The specification of any rate conversion which needs to be applied to the observed rate before being used in any calculations. The two common conversions are for securities quoted on a bank discount basis which will need to be converted to either a Money Market Yield or Bond Equivalent Yield. See the Annex to the 2000 ISDA Definitions, Section 7.3. Certain General Definitions Relating to Floating Rate Options, paragraphs (g) and (h) for definitions of these terms.

capRateSchedule (zero or more occurrences; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_StrikeRateSchedule)

- The cap rate or cap rate schedule, if any, which applies to the floating rate. The cap rate (strike) is only required where the floating rate on a swap stream is capped at a certain strike level. A cap rate schedule is expressed as explicit cap rates and dates and the step dates may be subject to adjustment in accordance with any adjustments specified in calculationPeriodDatesAdjustments. The cap rate is assumed to be exclusive of any spread and is a per annum rate, expressed as a decimal. A cap rate of 5% would be represented as 0.05.

floorRateSchedule (zero or more occurrences; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_StrikeRateSchedule)

- The floor rate or floor rate schedule, if any, which applies to the floating rate. The floor rate (strike) is only required where the floating rate on a swap stream is floored at a certain strike level. A floor rate schedule is expressed as explicit floor rates and dates and the step dates may be subject to adjustment in accordance with any adjustments specified in calculationPeriodDatesAdjustments. The floor rate is assumed to be exclusive of any spread and is a per annum rate, expressed as a decimal. A floor rate of 5% would be represented as 0.05.

Used by:

FpML_FloatingRateCalculation
floatingRate

DTD Fragment:

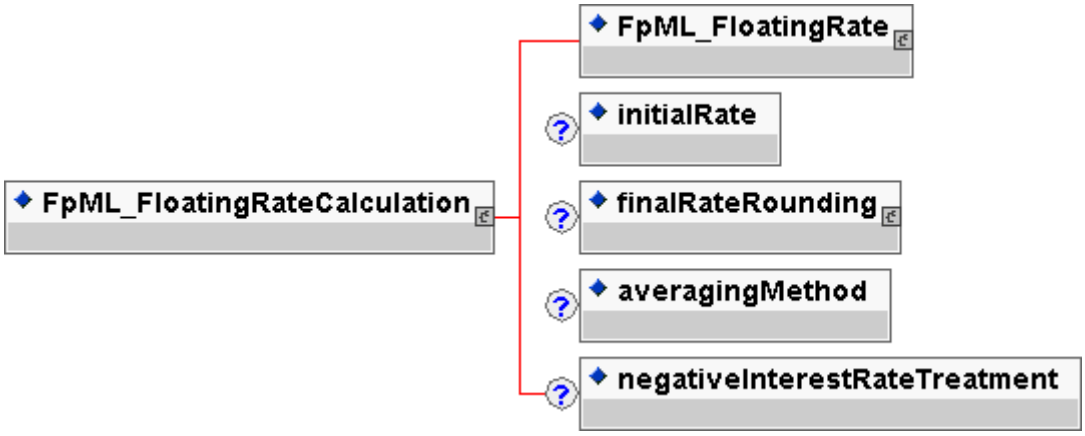
```
<!ENTITY % FpML_FloatingRate "floatingRateIndex , indexTenor? ,
floatingRateMultiplierSchedule? , spreadSchedule? , rateTreatment? ,
capRateSchedule* , floorRateSchedule*">
```

FpML_FloatingRateCalculation

Description:

An entity for defining the floating rate definitions and definitions relating to the calculation of floating rate amounts. This entity inherits from a base entity, FpML_FloatingRate.

Figure:



Contents:

inherited element(s) (this entity inherits the element(s) defined by exactly one occurrence of the entity FpML_FloatingRate)

- An entity for defining the floating rate definitions.

initialRate (zero or one occurrence; of type *decimal*)

- The initial floating rate reset agreed between the principal parties involved in the trade. This is assumed to be the first required reset rate for the first regular calculation period. It should only be included when the rate is not equal to the rate published on the source implied by the floating rate index. An initial rate of 5% would be represented as 0.05.

finalRateRounding (zero or one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_Rounding)

- The rounding convention to apply to the final rate used in determination of a calculation period amount.

averagingMethod (zero or one occurrence; of type *string*, an enumerated domain value defined by *averagingMethodScheme*)

- If averaging is applicable, this element specifies whether a weighted or unweighted average method of calculation is to be used. The element must only be included when averaging applies.

negativeInterestRateTreatment (zero or one occurrence; of type *string*, an enumerated domain value defined by *negativeInterestRateTreatmentScheme*)

- The specification of any provisions for calculating payment obligations when a floating rate is negative (either due to a quoted negative

floating rate or by operation of a spread that is subtracted from the floating rate).

Used by:

floatingRateCalculation

DTD Fragment:

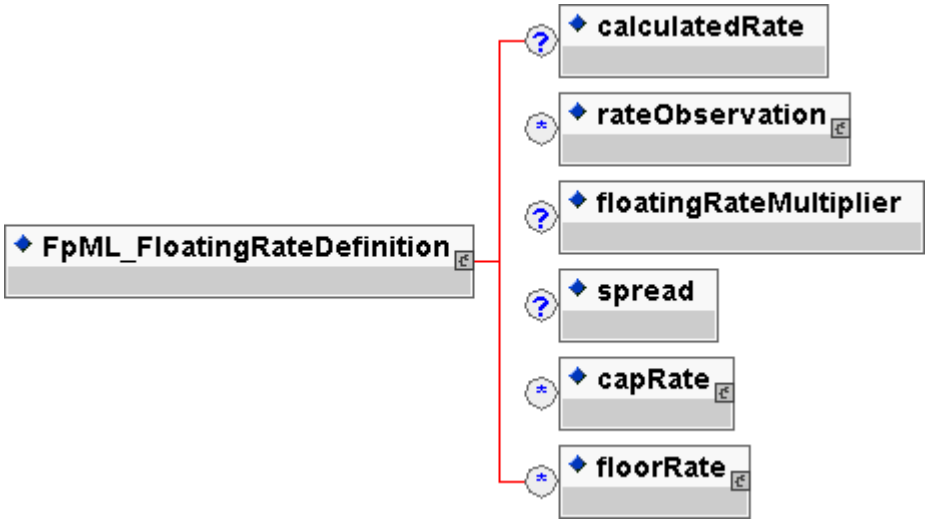
```
<!ENTITY % FpML_FloatingRateCalculation "(%FpML_FloatingRate; , initialRate?
, finalRateRounding? , averagingMethod? , negativeInterestRateTreatment?)">
```

FpML_FloatingRateDefinition

Description:

An entity defining parameters associated with a floating rate reset. This entity forms part of the cashflows representation of a stream.

Figure:



Contents:

calculatedRate (zero or one occurrence; of type *decimal*)

- The final calculated rate for a calculation period after any required averaging of rates. A calculated rate of 5% would be represented as 0.05.

rateObservation (zero or more occurrences; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_RateObservation)

- The details of a particular rate observation, including the fixing date and observed rate. A list of rate observation elements may be ordered in the document by ascending adjusted fixing date. An FpML document containing an unordered list of rate observations is still regarded as a conformant document.

floatingRateMultiplier (zero or one occurrence; of type *decimal*)

- An amount by which the floating rate is multiplied by.

spread (zero or one occurrence; of type *decimal*)

- The ISDA Spread, if any, which applies for the calculation period. The spread is a per annum rate, expressed as a decimal. For purposes of determining a calculation period amount, if positive the spread will be added to the floating rate and if negative the spread will be subtracted from the floating rate. A positive 10 basis point (0.1%) spread would be represented as 0.001.

capRate (zero or more occurrences; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_StrikeRate)

- The cap rate, if any, which applies to the floating rate for the calculation period. The cap rate (strike) is only required where the floating rate on a swap stream is capped at a certain strike level. The cap rate is assumed to be exclusive of any spread and is a per annum rate, expressed as a decimal. A cap rate of 5% would be represented as 0.05.

floorRate (zero or more occurrences; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_StrikeRate)

- The floor rate, if any, which applies to the floating rate for the calculation period. The floor rate (strike) is only required where the floating rate on a swap stream is floored at a certain strike level. The floor rate is assumed to be exclusive of any spread and is a per annum rate, expressed as a decimal. A floor rate of 5% would be represented as 0.05.

Used by:

floatingRateDefinition

DTD Fragment:

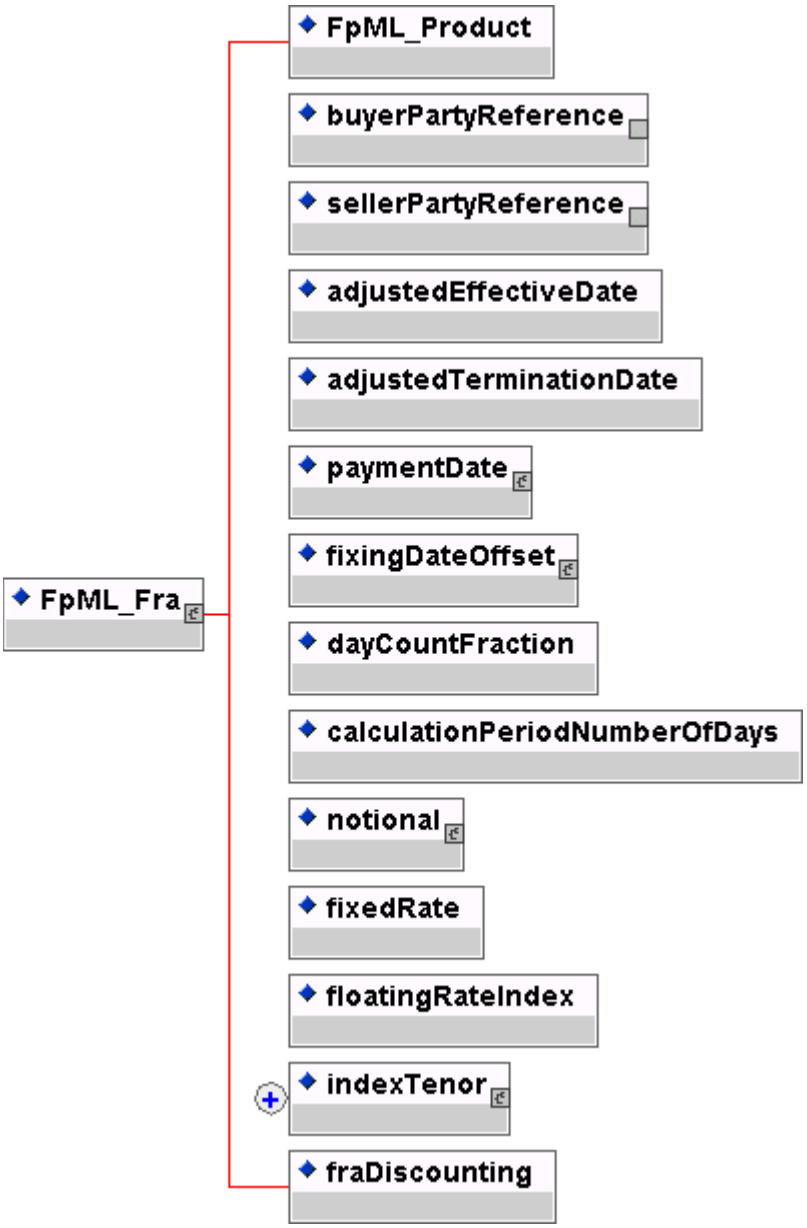
```
<!ENTITY % FpML_FloatingRateDefinition "calculatedRate? , rateObservation* ,
floatingRateMultiplier? , spread? , capRate* , floorRate*">
```

FpML_Fra

Description:

An entity for defining the forward rate agreement (FRA) product.

Figure:



Contents:

inherited element(s) (this entity inherits the element(s) defined by exactly one occurrence of the entity FpML_Product)

- The base entity which all FpML products extend.

buyerPartyReference (exactly one occurrence; an *empty* element containing an *href* attribute)

- A pointer style reference to a party identifier defined elsewhere in the document. The party referenced is the buyer of the instrument, also known as the fixed rate payer.

sellerPartyReference (exactly one occurrence; an *empty* element containing an *href* attribute)

- A pointer style reference to a party identifier defined elsewhere in the document. The party referenced is the seller of the instrument, also known as the floating rate payer.

adjustedEffectiveDate (exactly one occurrence; of type *date*)

- The start date of the calculation period. This date should already be adjusted for any applicable business day convention. This is also the date when the observed rate is applied, the reset date.

adjustedTerminationDate (exactly one occurrence; of type *date*)

- The end date of the calculation period. This date should already be adjusted for any applicable business day convention.

paymentDate (exactly one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_AdjustableDate)

- The payment date. This date is subject to adjustment in accordance with any applicable business day convention.

fixingDateOffset (exactly one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_RelativeDateOffset)

- Specifies the fixing date relative to the reset date in terms of a business days offset and an associated set of financial business centers. Normally these offset calculation rules will be those specified in the ISDA definition for the relevant floating rate index (ISDA's Floating Rate Option). However, non-standard offset calculation rules may apply for a trade if mutually agreed by the principal parties to the transaction. The href attribute on the dateRelativeTo element should reference the id attribute on the adjustedEffectiveDate element.

dayCountFraction (exactly one occurrence; of type *string*, an enumerated domain value defined by *dayCountFractionScheme*)

- The day count fraction.

calculationPeriodNumberOfDays (exactly one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_positiveInteger)

- The number of days from the adjusted effective date to the adjusted termination date calculated in accordance with the applicable day count fraction.

notional (exactly one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_Money)

- The notional amount.

fixedRate (exactly one occurrence; of type *decimal*)

- The calculation period fixed rate. A per annum rate, expressed as a decimal. A fixed rate of 5% would be represented as 0.05.

floatingRateIndex (exactly one occurrence; of type *string*, an enumerated domain value defined by *floatingRateIndexScheme*)

- The ISDA Floating Rate Option, i.e. the floating rate index.

indexTenor (one or more occurrences; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_Interval)

- The ISDA Designated Maturity, i.e. the tenor of the floating rate.

fraDiscounting (exactly one occurrence; of type *boolean*)

- A true/false flag to indicate whether ISDA FRA Discounting applies. If false, then the calculation will be based on a par value and no discounting will apply.

Used by:

fra

DTD Fragment:

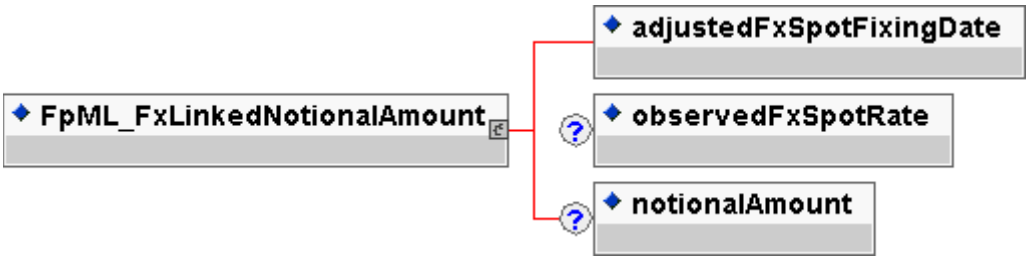
```
<!ENTITY % FpML_Fra "%FpML_Product;,buyerPartyReference ,
sellerPartyReference , adjustedEffectiveDate , adjustedTerminationDate ,
paymentDate , fixingDateOffset , dayCountFraction ,
calculationPeriodNumberOfDays , notional , fixedRate , floatingRateIndex ,
indexTenor+ , fraDiscounting">
```

FpML_FxLinkedNotionalAmount

Description:

An entity to describe the cashflow representation for fx linked notionals.

Figure:



Contents:

adjustedFxSpotFixingDate (exactly one occurrence; of type *date*)

- The date on which the fx spot rate is observed. This date should already be adjusted for any applicable business day convention.

observedFxSpotRate (zero or one occurrence; of type *decimal*)

- The actual observed fx spot rate.

notionalAmount (zero or one occurrence; of type *decimal*)

- The calculation period notional amount. The notional in the currency of the stream. This notional can be calculated once the FX Spot rate is known. It is optional since it should not be present prior to the fx spot reset date.

Used by:

fxLinkedNotionalAmount

DTD Fragment:

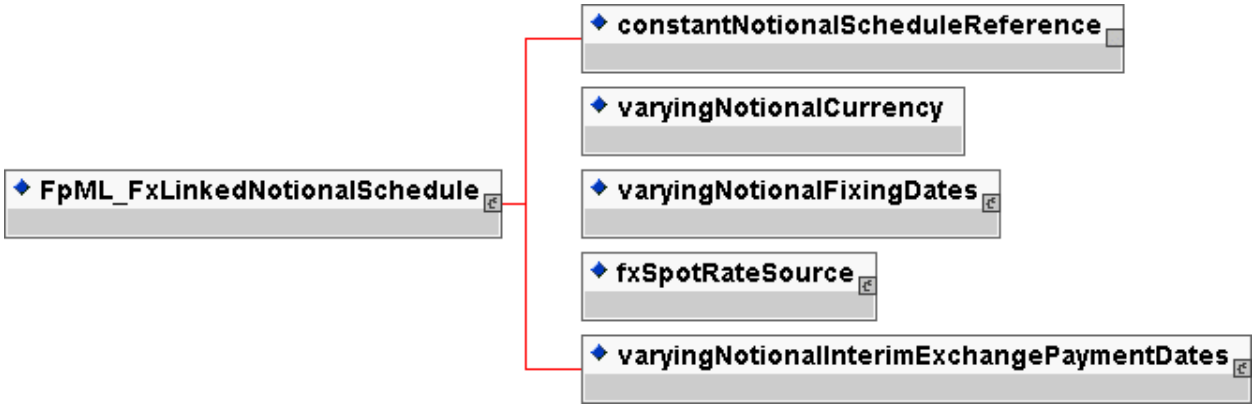
```
<!ENTITY % FpML_FxLinkedNotionalAmount "adjustedFxSpotFixingDate ,
observedFxSpotRate? , notionalAmount?">
```

FpML_FxLinkedNotionalSchedule

Description:

An entity to describe a notional amount schedule where each notional that applies to a calculation period is calculated with reference to a notional amount or notional amount schedule in a different currency by means of a spot currency exchange rate which is normally observed at the beginning of each period.

Figure:



Contents:

constantNotionalScheduleReference (exactly one occurrence; an *empty* element containing an *href* attribute)

- A pointer style reference to the associated constant notional schedule defined elsewhere in the document which contains the currency amounts which will be converted into the varying notional currency amounts using the spot currency exchange rate.

varyingNotionalCurrency (exactly one occurrence; of type *string*, an enumerated domain value defined by *currencyScheme*)

- The currency of the varying notional amount, i.e. the notional amount being determined periodically based on observation of a spot currency exchange rate.

varyingNotionalFixingDates (exactly one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_RelativeDateOffset)

- The dates on which spot currency exchange rates are observed for purposes of determining the varying notional currency amount that will apply to a calculation period.

fxSpotRateSource (exactly one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_FxSpotRateSource)

- The information source and time at which the spot currency exchange rate will be observed.

varyingNotionalInterimExchangePaymentDates (exactly one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_RelativeDateOffset)

- The dates on which interim exchanges of notional are paid. Interim exchanges will arise as a result of changes in the spot currency exchange amount or changes in the constant notional schedule (e.g. amortization).

Used by:

fxLinkedNotionalSchedule

DTD Fragment:

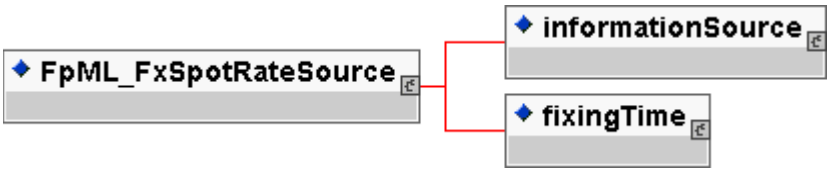
```
<!ENTITY % FpML_FxLinkedNotionalSchedule "constantNotionalScheduleReference ,
varyingNotionalCurrency , varyingNotionalFixingDates , fxSpotRateSource ,
varyingNotionalInterimExchangePaymentDates">
```

FpML_FxSpotRateSource

Description:

An entity to define the source and time for an fx rate.

Figure:



Contents:

informationSource (exactly one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_InformationSource)

- The information source where a published or displayed market rate will be obtained, e.g. Telerate Page 3750.

fixingTime (exactly one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_BusinessCenterTime)

- The time at which the spot currency exchange rate will be observed. It is specified as a time in a specific business center, e.g. 11:00 am London time.

Used by:

fxSpotRateSource

DTD Fragment:

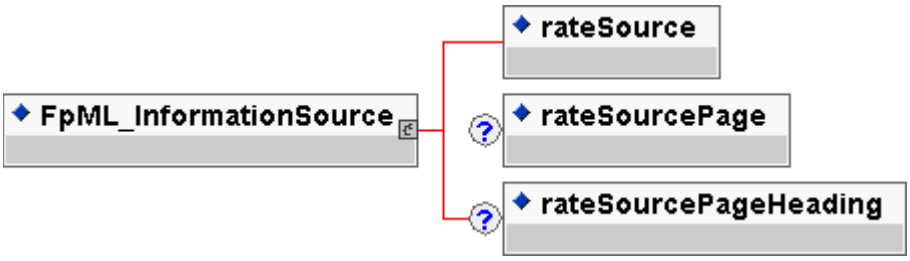
```
<!ENTITY % FpML_FxSpotRateSource "informationSource , fixingTime">
```

FpML_InformationSource

Description:

An entity to define the source for a piece of information (eg a rate refix or a fx fixing).

Figure:



Contents:

rateSource (exactly one occurrence; of type *string*, an enumerated domain value defined by *informationProviderScheme*)

- An information source for obtaining a market rate. For example Bloomberg, Reuters, Telerate etc.

rateSourcePage (zero or one occurrence; of type *string*, an enumerated domain value defined by *rateSourcePageScheme*)

- A specific page for the rate source for obtaining a market rate.

rateSourcePageHeading (zero or one occurrence; of type *string*)

- The specific information source page for obtaining a market rate. For example, 3750 (Telerate), LIBO (Reuters) etc.

Used by:

informationSource

DTD Fragment:

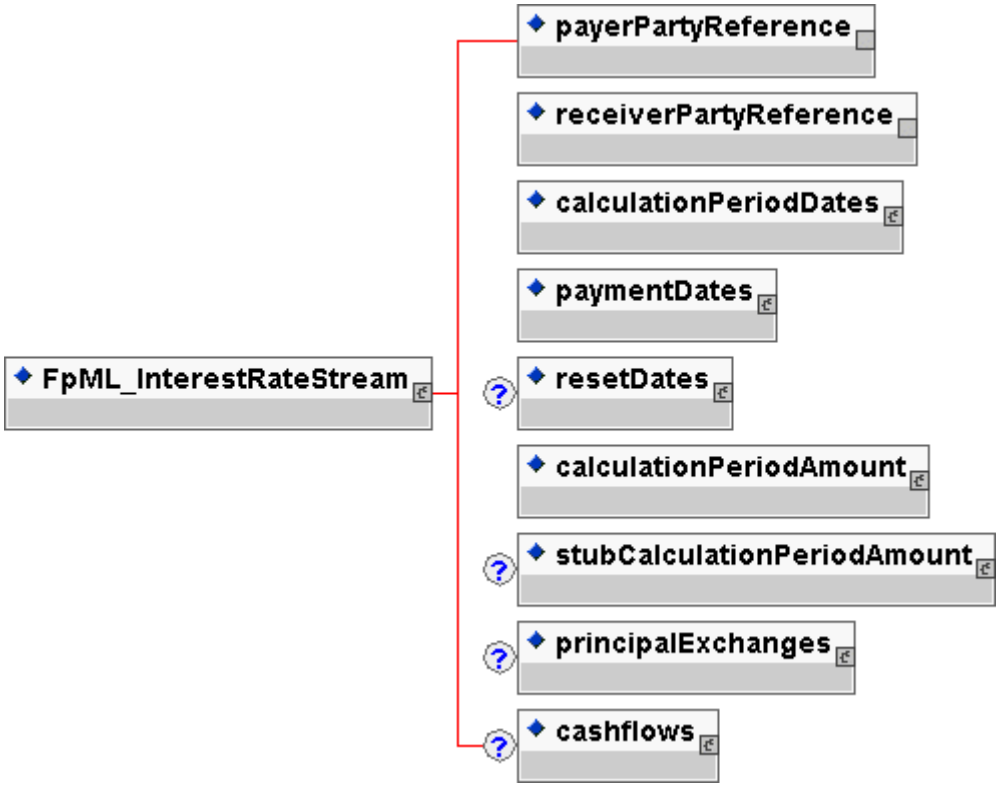
```
<!ENTITY % FpML_InformationSource "rateSource , rateSourcePage? ,
rateSourcePageHeading?">
```

FpML_InterestRateStream

Description:

An entity for defining the components specifying an interest rate payments stream, including both a parametric and cashflows representation for the stream of payments.

Figure:



Contents:

payerPartyReference (exactly one occurrence; an *empty* element containing an *href* attribute)

- A pointer style reference to a party identifier defined elsewhere in the document.

receiverPartyReference (exactly one occurrence; an *empty* element containing an *href* attribute)

- A pointer style reference to a party identifier defined elsewhere in the document.

calculationPeriodDates (exactly one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_CalculationPeriodDates)

- The calculation periods dates schedule.

paymentDates (exactly one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_PaymentDates)

- The payment dates schedule.

resetDates (zero or one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_ResetDates)

- The reset dates schedule. The reset dates schedule only applies for a floating rate stream.

calculationPeriodAmount (exactly one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_CalculationPeriodAmount)

- The calculation period amount parameters.

stubCalculationPeriodAmount (zero or one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_StubCalculationPeriodAmount)

- The stub calculation period amount parameters. This element must only be included if there is an initial or final stub calculation period. Even then, it must only be included if either the stub references a different floating rate tenor to the regular calculation periods, or if the stub is calculated as a linear interpolation of two different floating rate tenors, or if a specific stub rate or stub amount has been negotiated.

principalExchanges (zero or one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_PrincipalExchanges)

- The true/false flags indicating whether initial, intermediate or final exchanges of principal should occur.

cashflows (zero or one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_Cashflows)

- The cashflows representation of the swap stream.

Used by:

capFloorStream
swapStream

DTD Fragment:

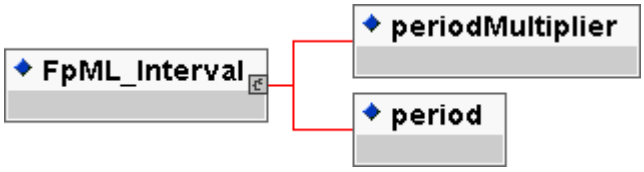
```
<!ENTITY % FpML_InterestRateStream "payerPartyReference ,
receiverPartyReference , calculationPeriodDates , paymentDates , resetDates?
, calculationPeriodAmount , stubCalculationPeriodAmount? ,
principalExchanges? , cashflows?">
```

FpML_Interval

Description:

An entity for defining a time interval or offset, e.g. one day, three months. Used for specifying frequencies at which events occur, the tenor of a floating rate or an offset relative to another date.

Figure:



Contents:

periodMultiplier (exactly one occurrence; of type *integer*)

- A time period multiplier, e.g. 1, 2 or 3 etc. A negative value can be used when specifying an offset relative to another date, e.g. -2 days. If the period value is T (Term) then periodMultiplier must contain the value 1.

period (exactly one occurrence; of type *string*, an enumerated domain value defined by *periodScheme*)

- A time period, e.g. a day, week, month, year or term of the stream. If the periodMultiplier value is 0 (zero) then period must contain the value D (day).

Used by:

FpML_CalculationPeriodFrequency
FpML_Offset
FpML_ResetFrequency
indexTenor
paymentFrequency
stepFrequency

DTD Fragment:

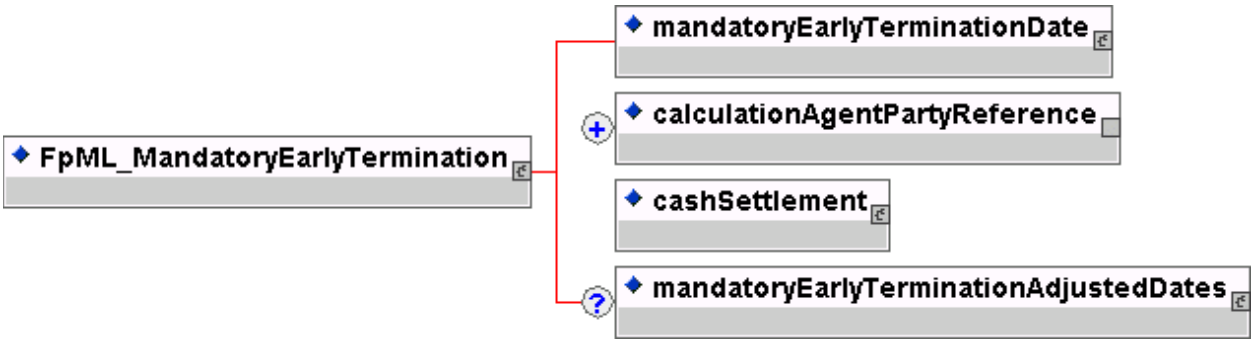
```
<!ENTITY % FpML_Interval "periodMultiplier , period">
```

FpML_MandatoryEarlyTermination

Description:

An entity to define the an early termination provision for which exercise is mandatory.

Figure:



Contents:

mandatoryEarlyTerminationDate (exactly one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_AdjustableDate)

- The early termination date associated with a mandatory early termination of a swap.

calculationAgentPartyReference (one or more occurrences; an *empty* element containing an *href* attribute)

- A pointer style reference to a party identifier defined elsewhere in the document. The party referenced is the ISDA Calculation Agent for the trade. If more than one party is referenced then the parties are assumed to be co-calculation agents, i.e. they have joint responsibility.

cashSettlement (exactly one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_CashSettlement)

- If specified, this means that cash settlement is applicable to the transaction and defines the parameters associated with the cash settlement procedure. If not specified, then physical settlement is applicable.

mandatoryEarlyTerminationAdjustedDates (zero or one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_MandatoryEarlyTerminationAdjustedDates)

- The adjusted dates associated with a mandatory early termination provision. These dates have been adjusted for any applicable business day convention.

Used by:

mandatoryEarlyTermination

DTD Fragment:

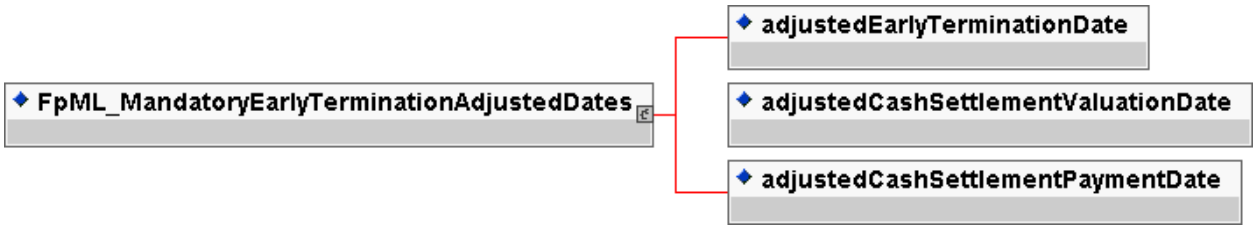
```
<!ENTITY % FpML_MandatoryEarlyTermination "mandatoryEarlyTerminationDate ,
calculationAgentPartyReference+ , cashSettlement ,
mandatoryEarlyTerminationAdjustedDates?">
```

FpML_MandatoryEarlyTerminationAdjustedDates

Description:

An entity to define the adjusted dates associated with a mandatory early termination provision.

Figure:



Contents:

adjustedEarlyTerminationDate (exactly one occurrence; of type *date*)

- The early termination date that is applicable if an early termination provision is exercised. This date should already be adjusted for any applicable business day convention.

adjustedCashSettlementValuationDate (exactly one occurrence; of type *date*)

- The date by which the cash settlement amount must be agreed. This date should already be adjusted for any applicable business day convention.

adjustedCashSettlementPaymentDate (exactly one occurrence; of type *date*)

- The date on which the cash settlement amount is paid. This date should already be adjusted for any applicable business day convention.

Used by:

mandatoryEarlyTerminationAdjustedDates

DTD Fragment:

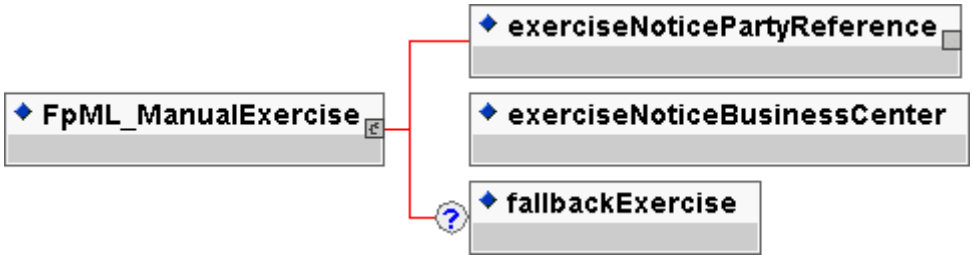
```
<!ENTITY % FpML_MandatoryEarlyTerminationAdjustedDates
"adjustedEarlyTerminationDate , adjustedCashSettlementValuationDate ,
adjustedCashSettlementPaymentDate">
```

FpML_ManualExercise

Description:

An entity to define manual exercise. ie that option buyer counterparty must give notice to the option seller of exercise.

Figure:



Contents:

exerciseNoticePartyReference (exactly one occurrence; an *empty* element containing an *href* attribute)

- A pointer style reference to a party identifier defined elsewhere in the document. The party referenced is the party to which notice of exercise should be given by the buyer.

exerciseNoticeBusinessCenter (exactly one occurrence; of type *string*, an enumerated domain value defined by *businessCenterScheme*)

- The business center location where notice of exercise should be given to the seller or seller's agent.

fallbackExercise (zero or one occurrence; of type *boolean*)

- If fallback exercise is specified then the notional amount of the underlying swap, not previously exercised under the swaption, will be automatically exercised at the expiration time on the expiration date if at such time the buyer is in-the-money, provided that the difference between the settlement rate and the fixed rate under the relevant underlying swap is not less than one tenth of a percentage point (0.10% or 0.001). The term In-the-money is assumed to have the meaning defined in the 2000 ISDA Definitions, Section 17.4. In-the-money.

Used by:

manualExercise

DTD Fragment:

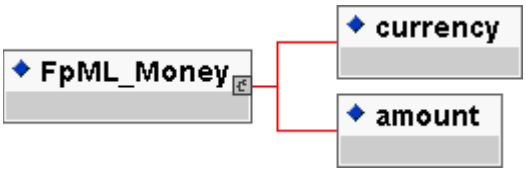
```
<!ENTITY % FpML_ManualExercise "exerciseNoticePartyReference ,
exerciseNoticeBusinessCenter , fallbackExercise?">
```

FpML_Money

Description:

An entity for defining a currency amount.

Figure:



Contents:

currency (exactly one occurrence; of type *string*, an enumerated domain value defined by *currencyScheme*)

- The currency in which an amount is denominated.

amount (exactly one occurrence; of type *decimal*)

- The monetary quantity in currency units.

Used by:

notional
paymentAmount
stubAmount

DTD Fragment:

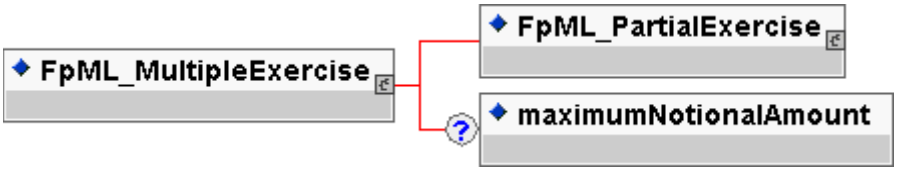
```
<!ENTITY % FpML_Money "currency , amount">
```

FpML_MultipleExercise

Description:

An entity to define multiple exercise. As defined in the 2000 ISDA Definitions, Section 12.4. Multiple Exercise, the buyer of the option has the right to exercise all or less than all the unexercised notional amount of the underlying swap on one or more days in the exercise period, but on any such day may not exercise less than the minimum notional amount or more than the maximum notional amount, and if an integral multiple amount is specified, the notional amount exercised must be equal to, or be an integral multiple of, the integral multiple amount.

Figure:



Contents:

inherited element(s) (this entity inherits the element(s) defined by exactly one occurrence of the entity FpML_PartialExercise)

- An entity to define partial exercise. As defined in the 2000 ISDA Definitions, Section 12.3 Partial Exercise, the buyer of the option may exercise all or less than all the notional amount of the underlying swap but may not be less than the minimum notional amount (if specified) and must be an integral multiple of the integral multiple amount if specified.

maximumNotionalAmount (zero or one occurrence; of type *decimal*)

- The maximum notional amount that can be exercised on a given exercise date.

Used by:

multipleExercise

DTD Fragment:

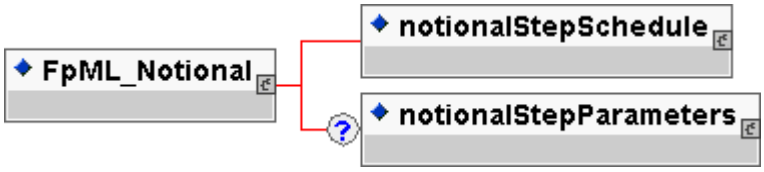
```
<!ENTITY % FpML_MultipleExercise "%FpML_PartialExercise; ,
maximumNotionalAmount?">
```

FpML_Notional

Description:

An entity for defining the notional amount or notional amount schedule associated with a swap stream. The notional schedule will be captured by explicitly specifying the dates that the notional changes and the outstanding notional amount that applies from that date. A parametric representation of the rules defining the notional step schedule can optionally be included.

Figure:



Contents:

notionalStepSchedule (exactly one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_AmountSchedule)

- The notional amount or notional amount schedule expressed as explicit outstanding notional amounts and dates. In the case of a schedule, the step dates may be subject to adjustment in accordance with any adjustments specified in calculationPeriodDatesAdjustments.

notionalStepParameters (zero or one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_NotionalStepRule)

- A parametric representation of the notional step schedule, i.e. parameters used to generate the notional schedule.

Used by:

notionalSchedule

DTD Fragment:

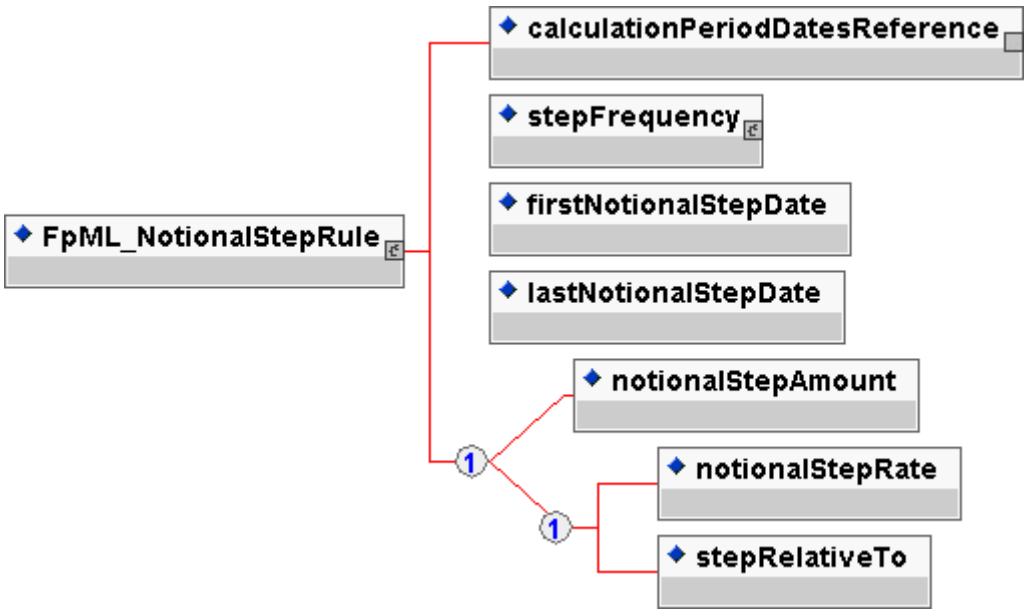
```
<!ENTITY % FpML_Notional "notionalStepSchedule , notionalStepParameters?">
```

FpML_NotionalStepRule

Description:

An entity for defining a parametric representation of the notional step schedule, i.e. parameters used to generate the notional balance on each step date. The step change in notional can be expressed in terms of either a fixed amount or as a percentage of either the initial notional or previous notional amount. This parametric representation is intended to cover the more common amortizing/accreting.

Figure:



Contents:

calculationPeriodDatesReference (exactly one occurrence; an *empty* element containing an *href* attribute)

- A pointer style reference to the associated calculation period dates component defined elsewhere in the document.

stepFrequency (exactly one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_Interval)

- The frequency at which the step changes occur. This frequency must be a multiple of the stream calculation period frequency.

firstNotionalStepDate (exactly one occurrence; of type *date*)

- The unadjusted calculation period start date of the first change in notional. This day may be subject to adjustment in accordance with any adjustments specified in calculationPeriodDatesAdjustments.

lastNotionalStepDate (exactly one occurrence; of type *date*)

- The unadjusted calculation period end date of the last change in notional. This day may be subject to adjustment in accordance with any adjustments specified in calculationPeriodDatesAdjustments.

Either

notionalStepAmount (exactly one occurrence; of type *decimal*)

- The explicit amount that the notional changes on each step date. This can be a positive or negative amount.

Or

notionalStepRate (exactly one occurrence; of type *decimal*)

- The percentage amount by which the notional changes on each step date. The percentage is either a percentage applied to the initial notional amount or the previous outstanding notional, depending on the value of the element stepRelativeTo. The percentage can be either positive or negative. A percentage of 5% would be represented as 0.05.

Or

stepRelativeTo (exactly one occurrence; of type *string*, an enumerated domain value defined by *stepRelativeToScheme*)

- Specifies whether the notionalStepRate should be applied to the initial notional or the previous notional in order to calculate the notional step change amount.

Used by:

notionalStepParameters

DTD Fragment:

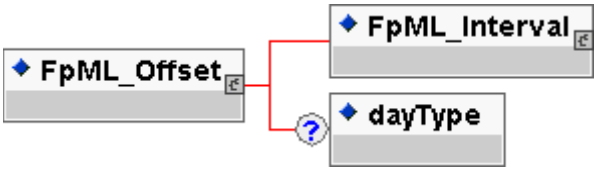
```
<!ENTITY % FpML_NotionalStepRule "calculationPeriodDatesReference ,
stepFrequency , firstNotionalStepDate , lastNotionalStepDate ,
(notionalStepAmount | (notionalStepRate , stepRelativeTo))">
```

FpML_Offset

Description:

An entity for defining an offset used in calculating a new date relative to a reference date. Currently, the only offsets defined are expected to be expressed as either calendar or business day offsets. This entity inherits from a base entity, FpML_Interval.

Figure:



Contents:

inherited element(s) (this entity inherits the element(s) defined by exactly one occurrence of the entity FpML_Interval)

- An entity for defining a time interval or offset, e.g. one day, three months. Used for specifying frequencies at which events occur, the tenor of a floating rate or an offset relative to another date.

dayType (zero or one occurrence; of type *string*, an enumerated domain value defined by *dayTypeScheme*)

- In the case of an offset specified as a number of days, this element defines whether consideration is given as to whether a day is a good business day or not. If a day type of business days is specified then non-business days are ignored when calculating the offset. The financial business centers to use for determination of business days are implied by the context in which this element is used. This element must only be included when the offset is specified as a number of days. If the offset is zero days then the dayType element should not be included.

Used by:

FpML_RelativeDateOffset
paymentDaysOffset
rateCutOffDaysOffset

DTD Fragment:

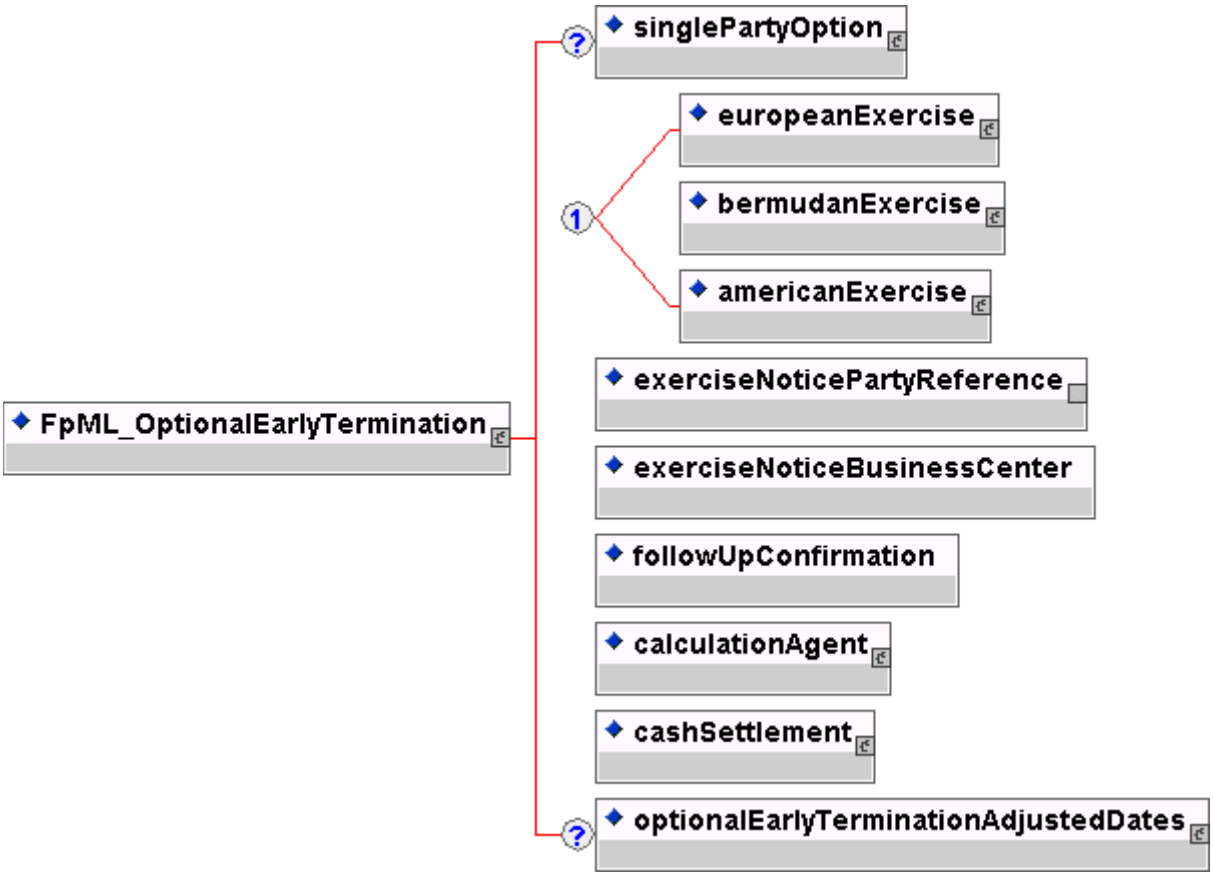
```
<!ENTITY % FpML_Offset "(%FpML_Interval; , dayType?)">
```

FpML_OptionalEarlyTermination

Description:

An entity to define an early termination provision where either or both parties have the right to exercise.

Figure:



Contents:

singlePartyOption (zero or one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_SinglePartyOption)

- If optional early termination is not available to both parties then this component specifies the buyer and seller of the option.

Either

europeanExercise (exactly one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_EuropeanExercise)

- The parameters for defining the exercise period for a European style option together with any rules governing the notional amount of the underlying which can be exercised on any given exercise date and any associated exercise fees.

Or

bermudanExercise (exactly one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_BermudanExercise)

- The parameters for defining the exercise period for a Bermudan style option together with any rules governing the notional amount of the underlying which can be exercised on any given exercise date and any associated exercise fees.

Or

americanExercise (exactly one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_AmericanExercise)

- The parameters for defining the exercise period for an American style option together with any rules governing the notional amount of the underlying which can be exercised on any given exercise date and any associated exercise fees.

exerciseNoticePartyReference (exactly one occurrence; an *empty* element containing an *href* attribute)

- A pointer style reference to a party identifier defined elsewhere in the document. The party referenced is the party to which notice of exercise should be given by the buyer.

exerciseNoticeBusinessCenter (exactly one occurrence; of type *string*, an enumerated domain value defined by *businessCenterScheme*)

- The business center location where notice of exercise should be given to the seller or seller's agent.

followUpConfirmation (exactly one occurrence; of type *boolean*)

- A flag to indicate whether follow-up confirmation of exercise (written or electronic) is required following telephonic notice by the buyer to the seller or seller's agent.

calculationAgent (exactly one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_CalculationAgent)

- The ISDA Calculation Agent responsible for performing duties associated with an optional early termination.

cashSettlement (exactly one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_CashSettlement)

- If specified, this means that cash settlement is applicable to the transaction and defines the parameters associated with the cash settlement procedure. If not specified, then physical settlement is applicable.

optionalEarlyTerminationAdjustedDates (zero or one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_OptionalEarlyTerminationAdjustedDates)

- An early termination provision to terminate the trade at fair value where one or both parties have the right to decide on termination.

Used by:

optionalEarlyTermination

DTD Fragment:

```
<!ENTITY % FpML_OptionalEarlyTermination "singlePartyOption? ,
(europeanExercise | bermudanExercise | americanExercise) ,
exerciseNoticePartyReference , exerciseNoticeBusinessCenter ,
followUpConfirmation , calculationAgent , cashSettlement ,
optionalEarlyTerminationAdjustedDates?">
```

FpML_OptionalEarlyTerminationAdjustedDates

Description:

An entity to define the adjusted dates associated with an optional early termination provision.

Figure:

Contents:

earlyTerminationEvent (one or more occurrences; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_EarlyTerminationEvent)

- The adjusted dates associated with an individual early termination date.

Used by:

optionalEarlyTerminationAdjustedDates

DTD Fragment:

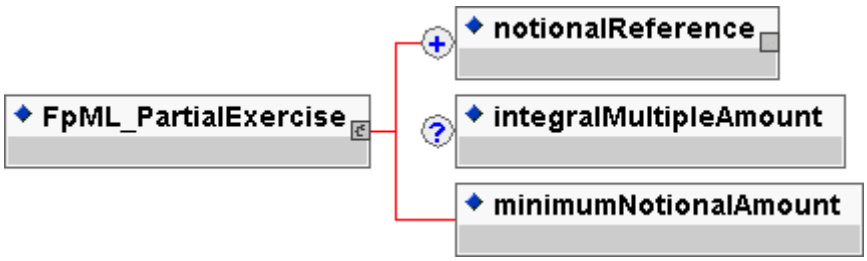
```
<!ENTITY % FpML_OptionalEarlyTerminationAdjustedDates
"earlyTerminationEvent+">
```

FpML_PartialExercise

Description:

An entity to define partial exercise. As defined in the 2000 ISDA Definitions, Section 12.3 Partial Exercise, the buyer of the option may exercise all or less than all the notional amount of the underlying swap but may not be less than the minimum notional amount (if specified) and must be an integral multiple of the integral multiple amount if specified.

Figure:



Contents:

notionalReference (one or more occurrences; an *empty* element containing an *href* attribute)

- A pointer style reference to the associated notional schedule defined elsewhere in the document.

integralMultipleAmount (zero or one occurrence; of type *decimal*)

- A notional amount which restricts the amount of notional that can be exercised when partial exercise or multiple exercise is applicable. The integral multiple amount defines a lower limit of notional that can be exercised and also defines a unit multiple of notional that can be exercised, i.e. only integer multiples of this amount can be exercised.

minimumNotionalAmount (exactly one occurrence; of type *decimal*)

- The minimum notional amount that can be exercised on a given exercise date. See multipleExercise.

Used by:

FpML_MultipleExercise
partialExercise

DTD Fragment:

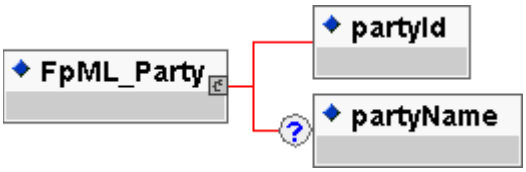
```
<!ENTITY % FpML_PartialExercise "notionalReference+ , integralMultipleAmount?
, minimumNotionalAmount">
```

FpML_Party

Description:

An entity for defining party identifier information.

Figure:



Contents:

partyId (exactly one occurrence; of type *string*, an enumerated domain value defined by *partyIdScheme*)

- A party identifier, e.g. a S.W.I.F.T. bank identifier code (BIC).

partyName (zero or one occurrence; of type *string*)

- The name of the party. A free format string. FpML does not define usage rules for this element

Used by:

party

DTD Fragment:

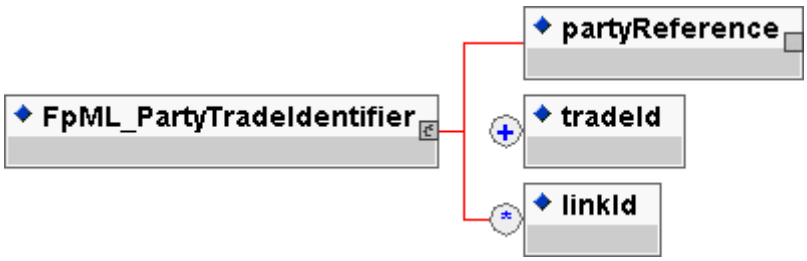
```
<!ENTITY % FpML_Party "partyId , partyName?">
```

FpML_PartyTradeIdentifier

Description:

An entity for defining one or more trade reference identifiers allocated to the trade by a party. A link identifier allows the trade to be associated with other related trades, e.g. trades forming part of a larger structured transaction. It is expected that for external communication of a trade there will be only one tradeId sent in the document per party.

Figure:



Contents:

partyReference (exactly one occurrence; an *empty* element containing an *href* attribute)

- A pointer style reference to a party identifier defined elsewhere in the document. The party referenced has allocated the trade identifier.

tradeId (one or more occurrences; of type *string*, an enumerated domain value defined by *tradeIdScheme*)

- A trade reference identifier allocated by a party. FpML does not define the domain values associated with this element. Note that the domain values for this element are not strictly an enumerated list.

linkId (zero or more occurrences; of type *string*, an enumerated domain value defined by *linkIdScheme*)

- A link identifier allowing the trade to be associated with other related trades, e.g. the linkId may contain a tradeId for an associated trade or several related trades may be given the same linkId. FpML does not define the domain values associated with this element. Note that the domain values for this element are not strictly an enumerated list.

Used by:

partyTradeIdentifier

DTD Fragment:

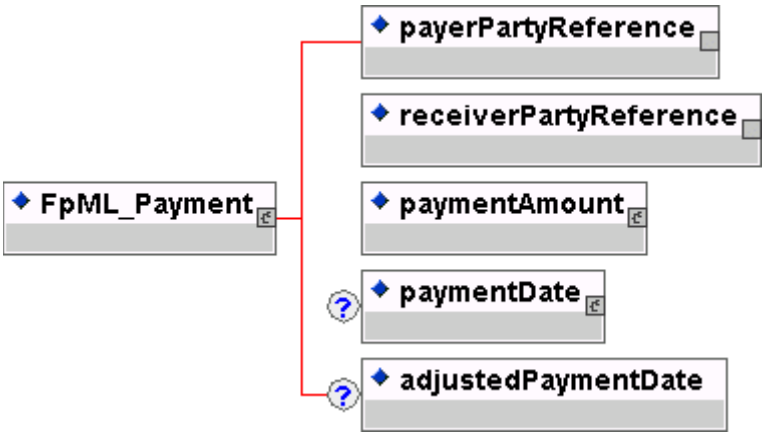
```
<!ENTITY % FpML_PartyTradeIdentifier "partyReference , tradeId+ , linkId*">
```

FpML_Payment

Description:

An entity for defining payments.

Figure:



Contents:

payerPartyReference (exactly one occurrence; an *empty* element containing an *href* attribute)

- A pointer style reference to a party identifier defined elsewhere in the document.

receiverPartyReference (exactly one occurrence; an *empty* element containing an *href* attribute)

- A pointer style reference to a party identifier defined elsewhere in the document.

paymentAmount (exactly one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_Money)

- The currency amount of the payment.

paymentDate (zero or one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_AdjustableDate)

- The payment date. This date is subject to adjustment in accordance with any applicable business day convention. This element is optional to allow the fee component to be used to capture commission amounts that might not have a known payment date associated with them, e.g. commissions may be invoiced and billed periodically.

adjustedPaymentDate (zero or one occurrence; of type *date*)

- The adjusted payment date. This date should already be adjusted for any applicable business day convention. This element is not intended for use in trade confirmation but may be specified to allow the fee structure to also serve as a cashflow type component (all dates in the FpML_Cashflows entity are adjusted payment dates).

Used by:

FpML_BulletPayment
premium

DTD Fragment:

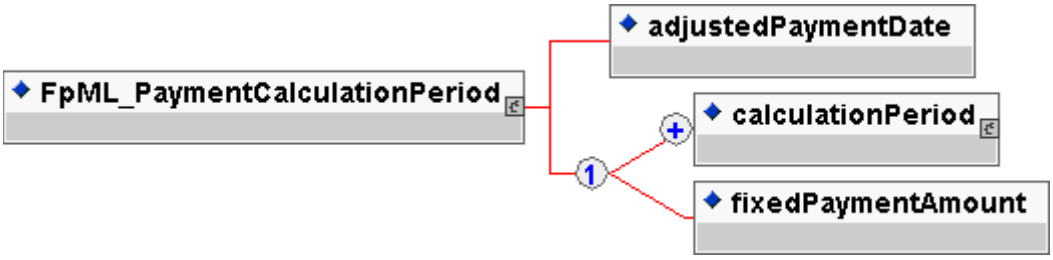
```
<!ENTITY % FpML_Payment "payerPartyReference , receiverPartyReference ,
paymentAmount , paymentDate? , adjustedPaymentDate?">
```

FpML_PaymentCalculationPeriod

Description:

An entity defining the adjusted payment date and associated calculation period parameters required to calculate the actual or projected payment amount. This entity forms part of the cashflows representation of a swap stream.

Figure:



Contents:

adjustedPaymentDate (exactly one occurrence; of type *date*)

- The adjusted payment date. This date should already be adjusted for any applicable business day convention.

Either

calculationPeriod (one or more occurrences; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_CalculationPeriod)

- The parameters used in the calculation of a fixed or floating rate calculation period amount. A list of calculation period elements may be ordered in the document by ascending adjusted start date. An FpML document which contains an unordered list of calculation periods is still regarded as a conformant document.

Or

fixedPaymentAmount (exactly one occurrence; of type *decimal*)

- A known fixed payment amount.

Used by:

paymentCalculationPeriod

DTD Fragment:

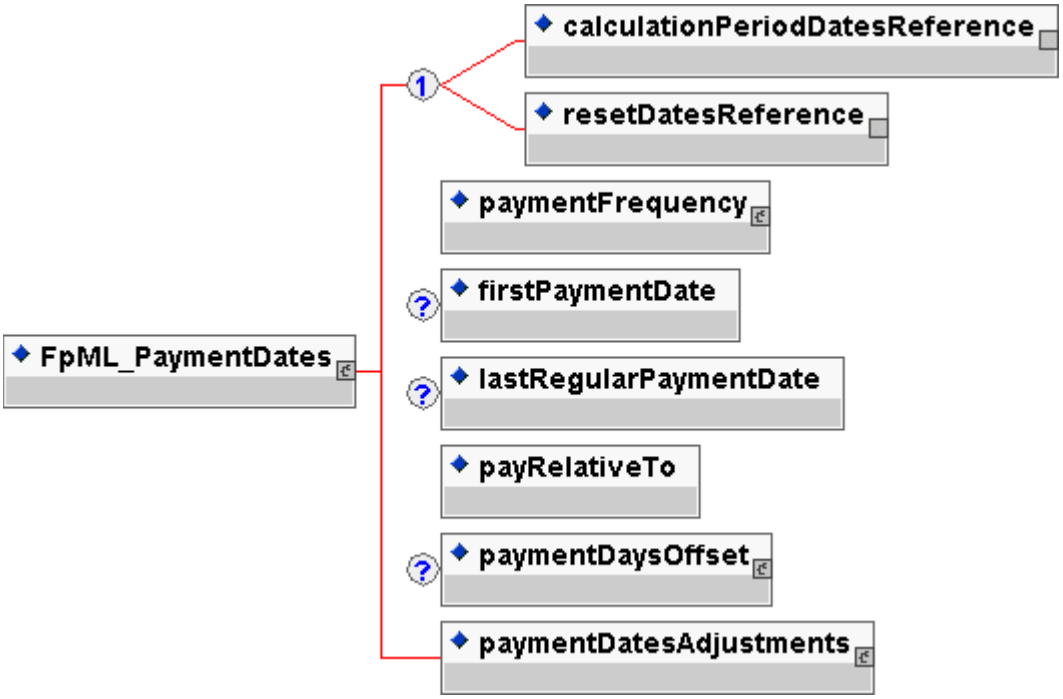
```
<!ENTITY % FpML_PaymentCalculationPeriod "adjustedPaymentDate ,
(calculationPeriod+ | fixedPaymentAmount)">
```

FpML_PaymentDates

Description:

An entity for defining the parameters used to generate the payment dates schedule, including the specification of early or delayed payments. Payment dates are determined relative to the calculation periods dates or the reset dates.

Figure:



Contents:

Either

calculationPeriodDatesReference (exactly one occurrence; an *empty* element containing an *href* attribute)

- A pointer style reference to the associated calculation period dates component defined elsewhere in the document.

Or

resetDatesReference (exactly one occurrence; an *empty* element containing an *href* attribute)

- A pointer style reference to the associated reset dates component defined elsewhere in the document.

paymentFrequency (exactly one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_Interval)

- The frequency at which regular payment dates occur. If the payment frequency is equal to the frequency defined in the calculation period dates component then one calculation period contributes to each payment

amount. If the payment frequency is less frequent than the frequency defined in the calculation period dates component then more than one calculation period will contribute to each payment amount. A payment frequency more frequent than the calculation period frequency or one that is not a multiple of the calculation period frequency is invalid.

firstPaymentDate (zero or one occurrence; of type *date*)

- The first unadjusted payment date. This day may be subject to adjustment in accordance with any business day convention specified in paymentDatesAdjustments. This element must only be included if there is an initial stub. This date will normally correspond to an unadjusted calculation period start or end date. This is true even if early or delayed payment is specified to be applicable since the actual first payment date will be the specified number of days before or after the applicable adjusted calculation period start or end date with the resulting payment date then being adjusted in accordance with any business day convention specified in paymentDatesAdjustments.

lastRegularPaymentDate (zero or one occurrence; of type *date*)

- The last regular unadjusted payment date. This day may be subject to adjustment in accordance with any business day convention specified in paymentDatesAdjustments. This element must only be included if there is a final stub. All calculation periods after this date contribute to the final payment. The final payment is made relative to the final set of calculation periods or the final reset date as the case may be. This date will normally correspond to an unadjusted calculation period start or end date. This is true even if early or delayed payment is specified to be applicable since the actual last regular payment date will be the specified number of days before or after the applicable adjusted calculation period start or end date with the resulting payment date then being adjusted in accordance with any business day convention specified in paymentDatesAdjustments.

payRelativeTo (exactly one occurrence; of type *string*, an enumerated domain value defined by *payRelativeToScheme*)

- Specifies whether the payments occur relative to each adjusted calculation period start date, adjusted calculation period end date or each reset date. The reset date is applicable in the case of certain euro (former French Franc) floating rate indices. Calculation period start date means relative to the start of the first calculation period contributing to a given payment. Similarly, calculation period end date means the end of the last calculation period contributing to a given payment.

paymentDaysOffset (zero or one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_Offset)

- If early payment or delayed payment is required, specifies the number of days offset that the payment occurs relative to what would otherwise be the unadjusted payment date. The offset can be specified in terms of either calendar or business days. Even in the case of a calendar days offset, the resulting payment date, adjusted for the specified calendar days offset, will still be adjusted in accordance with the specified payment dates adjustments. This element should only be included if early or delayed payment is applicable, i.e. if the periodMultiplier element value is not equal to zero. An early payment would be indicated by a negative periodMultiplier element value and a

delayed payment (or payment lag) would be indicated by a positive periodMultiplier element value.

paymentDatesAdjustments (exactly one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_BusinessDayAdjustments)

- The business day convention to apply to each payment date if it would otherwise fall on a day that is not a business day in the specified financial business centers.

Used by:

paymentDates

DTD Fragment:

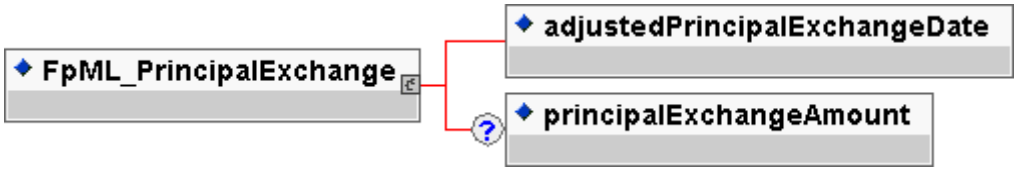
```
<!ENTITY % FpML_PaymentDates "((calculationPeriodDatesReference |
resetDatesReference) , paymentFrequency , firstPaymentDate? ,
lastRegularPaymentDate? , payRelativeTo , paymentDaysOffset? ,
paymentDatesAdjustments)">
```

FpML_PrincipalExchange

Description:

An entity for defining a principal exchange amount and adjusted exchange date. This entity forms part of the cashflows representation of a swap stream.

Figure:



Contents:

adjustedPrincipalExchangeDate (exactly one occurrence; of type *date*)

- The principal exchange date. This date should already be adjusted for any applicable business day convention.

principalExchangeAmount (zero or one occurrence; of type *decimal*)

- The principal exchange amount. This amount should be positive if the stream payer is paying the exchange amount and signed negative if they are receiving it.

Used by:

principalExchange

DTD Fragment:

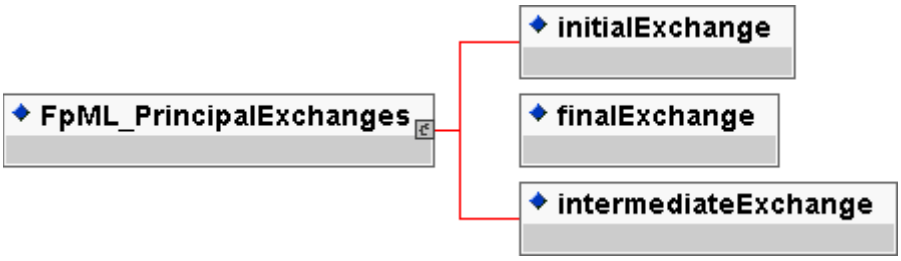
```
<!ENTITY % FpML_PrincipalExchange "adjustedPrincipalExchangeDate ,
principalExchangeAmount?">
```

FpML_PrincipalExchanges

Description:

An entity for defining which principal exchanges occur for the stream.

Figure:



Contents:

initialExchange (exactly one occurrence; of type *boolean*)

- A true/false flag to indicate whether there is an initial exchange of principal on the effective date.

finalExchange (exactly one occurrence; of type *boolean*)

- A true/false flag to indicate whether there is a final exchange of principal on the termination date.

intermediateExchange (exactly one occurrence; of type *boolean*)

- A true/false flag to indicate whether there are intermediate or interim exchanges of principal during the term of the swap.

Used by:

principalExchanges

DTD Fragment:

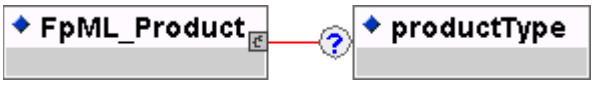
```
<!ENTITY % FpML_PrincipalExchanges "initialExchange , finalExchange ,
intermediateExchange">
```

FpML_Product

Description:

The base entity which all FpML products extend.

Figure:



Contents:

productType (zero or one occurrence; of type *string*, an enumerated domain value defined by *productTypeScheme*)

- A classification of the type of product. FpML does not define domain values for this element.

Used by:

FpML_BulletPayment
FpML_CapFloor
FpML_Fra
FpML_Strategy
FpML_Swap
FpML_Swaption

DTD Fragment:

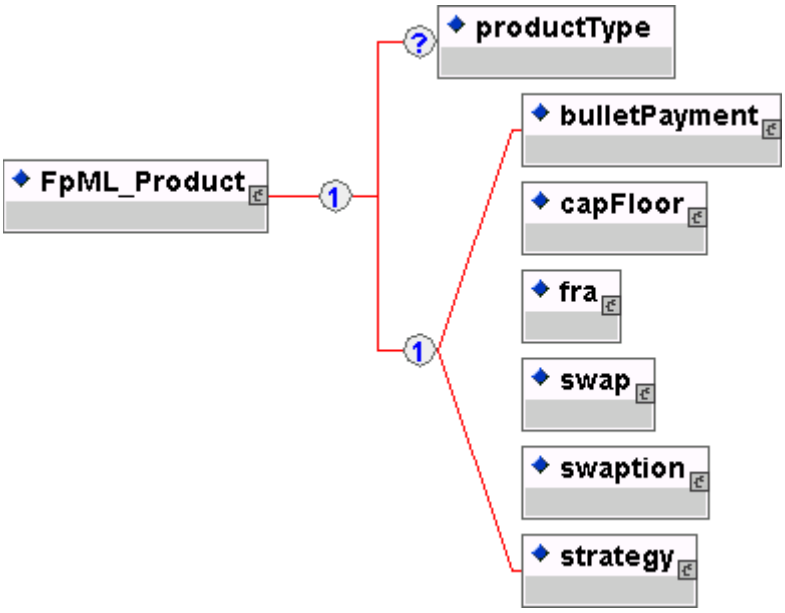
```
<!ENTITY % FpML_Product "productType?">
```

FpML_ProductList

Description:

An entity defining the available product definitions.

Figure:



Contents:

productType (zero or one occurrence; of type *string*, an enumerated domain value defined by *productTypeScheme*)

- A classification of the type of product. FpML does not define domain values for this element.

Either

bulletPayment (exactly one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_Fee)

- A product to represent one or more known payments.

Or

capFloor (exactly one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_CapFloor)

- A cap, floor or cap floor structures product definition.

Or

fra (exactly one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_Fra)

- A forward rate agreement product definition.

Or

swap (exactly one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_Swap)

- A swap product definition.

Or

swaption (exactly one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_Swaption)

- A swaption product definition.

Or

strategy (exactly one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_Strategy)

- A trade containing multiple products. It is envisaged that this will be used to represent structured products.

Used by:

product

DTD Fragment:

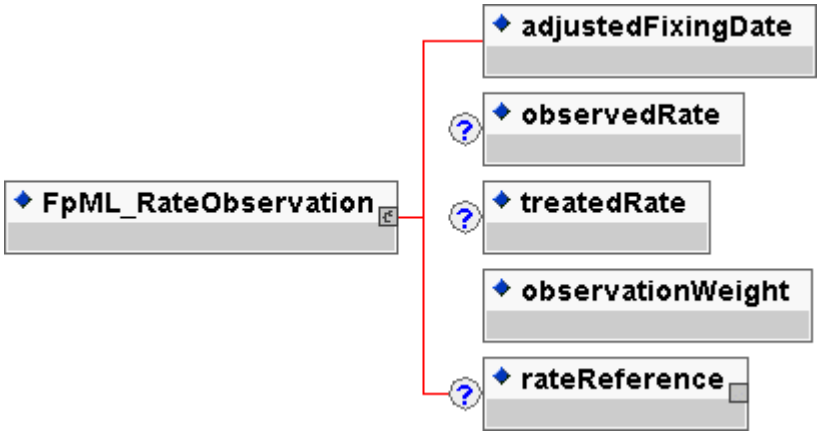
```
<!ENTITY % FpML_Product "productType? , (bulletPayment | capFloor | fra |
swap | swaption | strategy)">
```

FpML_RateObservation

Description:

An entity for defining parameters associated with an individual rate observation or fixing. This entity forms part of the cashflows representation of a stream.

Figure:



Contents:

adjustedFixingDate (exactly one occurrence; of type *date*)

- The adjusted fixing date, i.e. the actual date the rate is observed. This date should already be adjusted for any applicable business day convention.

observedRate (zero or one occurrence; of type *decimal*)

- The actual observed rate before any required rate treatment is applied, e.g. before converting a rate quoted on a discount basis to an equivalent yield. An observed rate of 5% would be represented as 0.05.

treatedRate (zero or one occurrence; of type *decimal*)

- The observed rate after any required rate treatment is applied. A treated rate of 5% would be represented as 0.05.

observationWeight (exactly one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_positiveInteger)

- The number of days weighting to be associated with the rate observation, i.e. the number of days such rate is in effect. This is applicable in the case of a weighted average method of calculation where more than one reset date is established for a single calculation period.

rateReference (zero or one occurrence; an *empty* element containing an *href* attribute)

- A pointer style reference to a floating rate component defined as part of a stub calculation period amount component. It is only required when it is necessary to distinguish two rate observations for the same

fixing date which could occur when linear interpolation of two different rates occurs for a stub calculation period.

Used by:

rateObservation

DTD Fragment:

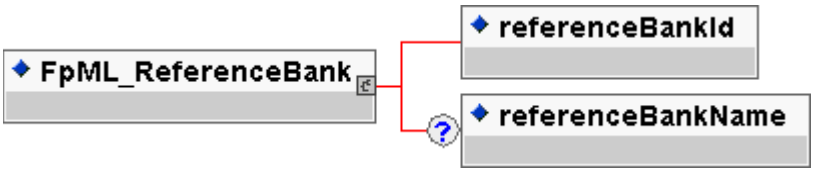
```
<!ENTITY % FpML_RateObservation "adjustedFixingDate , observedRate? ,
treatedRate? , observationWeight , rateReference?">
```

FpML_ReferenceBank

Description:

An entity to describe institution (party) identified by means of a coding scheme and an optional name.

Figure:



Contents:

referenceBankId (exactly one occurrence; of type *string*, an enumerated domain value defined by *referenceBankIdScheme*)

- An institution (party) identifier, e.g. a bank identifier code (BIC).

referenceBankName (zero or one occurrence; of type *string*)

- The name of the institution (party). A free format string. FpML does not define usage rules for the element.

Used by:

referenceBank

DTD Fragment:

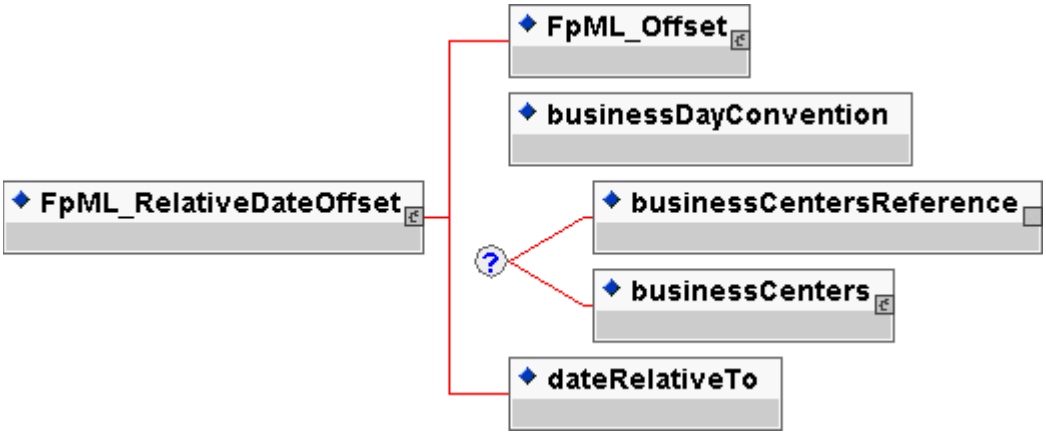
```
<!ENTITY % FpML_ReferenceBank "referenceBankId , referenceBankName?">
```

FpML_RelativeDateOffset

Description:

An entity for defining a date (referred to as the derived date) as a relative offset from another date (referred to as the anchor date). If the anchor date is itself an adjustable date then the offset is assumed to be calculated from the adjusted anchor date. A number of different scenarios can be supported, namely; 1) the derived date may simply be a number of calendar periods (days, weeks, months or years) preceding or following the anchor date; 2) the unadjusted derived date may be a number of calendar periods (days, weeks, months, years) preceding or following the anchor date with the resulting unadjusted derived date subject to adjustment in accordance with a specified business day convention, i.e. the derived date must fall on a good business day; 3) the derived date may be a number of business days preceding or following the anchor date. Note that the businessDayConvention element specifies any required adjustment to the unadjusted derived date. A negative or positive value in the periodMultiplier element indicates whether the unadjusted derived date precedes or follows the anchor date. The businessDayConvention element should contain a value of NONE if the dayType element contains a value of Business (since specifying a negative or positive business days offset would already guarantee that the derived date would fall on a good business day in the specified business centers).

Figure:



Contents:

inherited element(s) (this entity inherits the element(s) defined by exactly one occurrence of the entity FpML_Offset)

- An entity for defining an offset used in calculating a new date relative to a reference date. Currently, the only offsets defined are expected to be expressed as either calendar or business day offsets. This entity inherits from a base entity, FpML_Interval.

businessDayConvention (exactly one occurrence; of type *string*, an enumerated domain value defined by *businessDayConventionScheme*)

- The convention for adjusting a date if it would otherwise fall on a day that is not a business day. If the business day convention value is NONE then the businessCentersReference or businessCenters element should still be included if the dayType element contains a value of Business since the business centers defined are those used for determining good business days.

Zero or one occurrence of either

businessCentersReference (exactly one occurrence; an *empty* element containing an *href* attribute)

- A pointer style reference to a set of financial business centers defined elsewhere in the document. This set of business centers is used to determine whether a particular day is a business day or not.

Or

businessCenters (zero or one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_BusinessCenters)

- A container for a set of financial business centers. This set of business centers is used to determine whether a day is a business day or not.

dateRelativeTo (exactly one occurrence; of type *string*, an enumerated domain value defined by *dateRelativeToScheme*)

- Specifies the anchor date. This element also carries an href attribute. The href attribute value will be a pointer style reference to the element or component elsewhere in the document where the anchor date is defined.

Used by:

FpML_RelativeDates
cashSettlementValuationDate
feePaymentDate
fixingDateOffset
fixingDates
relativeDate
varyingNotionalFixingDates
varyingNotionalInterimExchangePaymentDates

DTD Fragment:

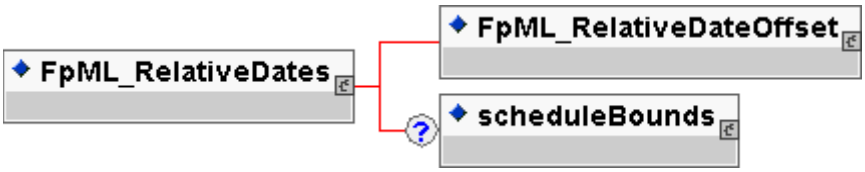
```
<!ENTITY % FpML_RelativeDateOffset "(%FpML_Offset; , businessDayConvention ,
(businessCentersReference | businessCenters)? , dateRelativeTo)">
```

FpML_RelativeDates

Description:

An entity to define a set of dates defined as relative to another set of dates.

Figure:



Contents:

inherited element(s) (this entity inherits the element(s) defined by exactly one occurrence of the entity FpML_RelativeDateOffset)

- An entity for defining a date (referred to as the derived date) as a relative offset from another date (referred to as the anchor date). If the anchor date is itself an adjustable date then the offset is assumed to be calculated from the adjusted anchor date. A number of different scenarios can be supported, namely; 1) the derived date may simply be a number of calendar periods (days, weeks, months or years) preceding or following the anchor date; 2) the unadjusted derived date may be a number of calendar periods (days, weeks, months, years) preceding or following the anchor date with the resulting unadjusted derived date subject to adjustment in accordance with a specified business day convention, i.e. the derived date must fall on a good business day; 3) the derived date may be a number of business days preceding or following the anchor date. Note that the businessDayConvention element specifies any required adjustment to the unadjusted derived date. A negative or positive value in the periodMultiplier element indicates whether the unadjusted derived date precedes or follows the anchor date. The businessDayConvention element should contain a value of NONE if the dayType element contains a value of Business (since specifying a negative or positive business days offset would already guarantee that the derived date would fall on a good business day in the specified business centers).

scheduleBounds (zero or one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_DateRange)

- The first and last dates of a schedule. This can be used to restrict the range of values in a reference series of dates.

Used by:

relativeDates

DTD Fragment:

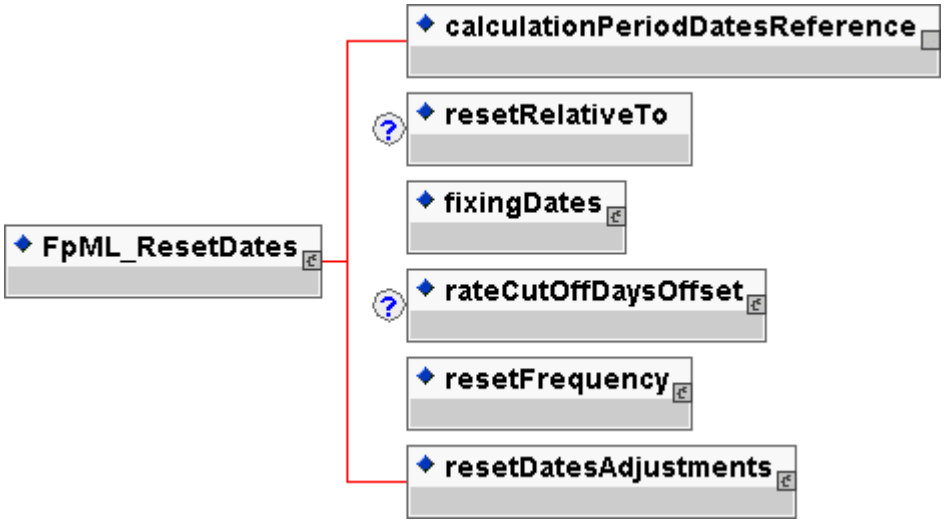
```
<!ENTITY % FpML_RelativeDates "(%FpML_RelativeDateOffset; ,
scheduleBounds?)">
```

FpML_ResetDates

Description:

An entity for defining the parameters used to generate the reset dates schedule and associated fixing dates. The reset dates are determined relative to the calculation periods schedule dates.

Figure:



Contents:

calculationPeriodDatesReference (exactly one occurrence; an *empty* element containing an *href* attribute)

- A pointer style reference to the associated calculation period dates component defined elsewhere in the document.

resetRelativeTo (zero or one occurrence; of type *string*, an enumerated domain value defined by *resetRelativeToScheme*)

- Specifies whether the reset dates are determined with respect to each adjusted calculation period start date or adjusted calculation period end date. If the reset frequency is specified as daily this element must not be included.

fixingDates (exactly one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_RelativeDateOffset)

- Specifies the fixing date relative to each reset date in terms of a business days offset and an associated set of financial business centers. Normally these offset calculation rules will be those specified in the ISDA definition for the relevant floating rate index (ISDA's Floating Rate Option). However, non-standard offset calculation rules may apply for a trade if mutually agreed by the principal parties to the transaction. The href attribute on the dateRelativeTo element should reference the id attribute on the resetDates element.

rateCutOffDaysOffset (zero or one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_Offset)

- Specifies the number of business days before the period end date when the rate cut-off date is assumed to apply. The financial business centers associated with determining the rate cut-off date are those specified in the reset dates adjustments. The rate cut-off number of days must be a negative integer (a value of zero would imply no rate cut off applies in which case the rateCutOffDaysOffset element should not be included). The relevant rate for each reset date in the period from, and including, a rate cut-off date to, but excluding, the next applicable period end date (or, in the case of the last calculation period, the termination date) will (solely for purposes of calculating the floating amount payable on the next applicable payment date) be deemed to be the relevant rate in effect on that rate cut-off date. For example, if rate cut-off days for a daily averaging deal is -2 business days, then the refix rate applied on (period end date - 2 days) will also be applied as the reset on (period end date - 1 day), i.e. the actual number of reset dates remains the same but from the rate cut-off date until the period end date, the same refix rate is applied. Note that in the case of several calculation periods contributing to a single payment, the rate cut-off is assumed only to apply to the final calculation period contributing to that payment. The day type associated with the offset must imply a business days offset.

resetFrequency (exactly one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_ResetFrequency)

- The frequency at which reset dates occur. In the case of a weekly reset frequency, also specifies the day of the week that the reset occurs. If the reset frequency is greater than the calculation period frequency then this implies that more than one reset date is established for each calculation period and some form of rate averaging is applicable.

resetDatesAdjustments (exactly one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_BusinessDayAdjustments)

- The business day convention to apply to each reset date if it would otherwise fall on a day that is not a business day in the specified financial business centers.

Used by:

resetDates

DTD Fragment:

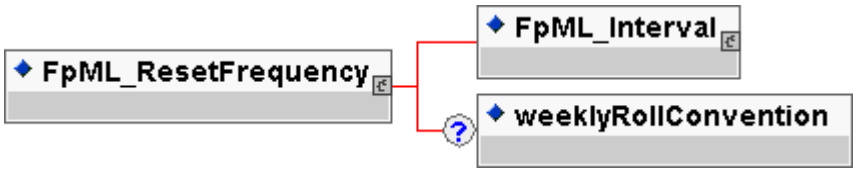
```
<!ENTITY % FpML_ResetDates "calculationPeriodDatesReference ,
resetRelativeTo? , fixingDates , rateCutOffDaysOffset? , resetFrequency ,
resetDatesAdjustments">
```

FpML_ResetFrequency

Description:

An entity for defining the reset frequency. In the case of a weekly reset, also specifies the day of the week that the reset occurs. This entity inherits from a base entity, FpML_Interval. If the reset frequency is greater than the calculation period frequency then this implies that more than one reset date is established for each calculation period and some form of rate averaging is applicable. The specific averaging method of calculation is specified in the entity FpML_FloatingRateCalculation.

Figure:



Contents:

inherited element(s) (this entity inherits the element(s) defined by exactly one occurrence of the entity FpML_Interval)

- An entity for defining a time interval or offset, e.g. one day, three months. Used for specifying frequencies at which events occur, the tenor of a floating rate or an offset relative to another date.

weeklyRollConvention (zero or one occurrence; of type *string*, an enumerated domain value defined by *weeklyRollConventionScheme*)

- The day of the week on which a weekly reset date occurs. This element must be included if the reset frequency is defined as weekly and not otherwise.

Used by:

resetFrequency

DTD Fragment:

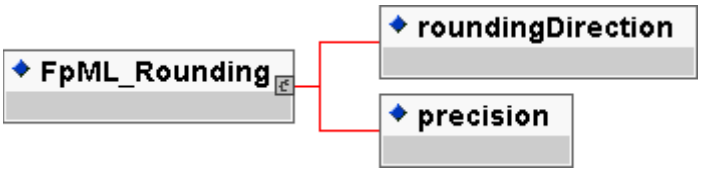
```
<!ENTITY % FpML_ResetFrequency "(%FpML_Interval; , weeklyRollConvention?)">
```

FpML_Rounding

Description:

An entity for defining a rounding direction and precision to be used in the rounding of a rate.

Figure:



Contents:

roundingDirection (exactly one occurrence; of type *string*, an enumerated domain value defined by *roundingDirectionScheme*)

- Specifies the rounding direction.

precision (exactly one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_nonNegativeInteger)

- Specifies the rounding precision in terms of a number of decimal places. Note how a percentage rate rounding of 5 decimal places is expressed as a rounding precision of 7 in the FpML document since the percentage is expressed as a decimal, e.g. 9.876543% (or 0.09876543) being rounded to the nearest 5 decimal places is 9.87654% (or 0.0987654).

Used by:

finalRateRounding

DTD Fragment:

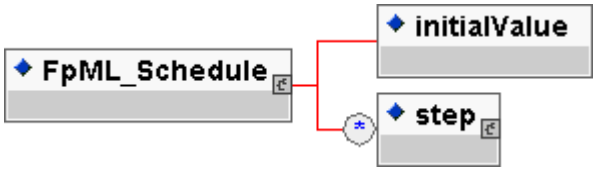
```
<!ENTITY % FpML_Rounding "roundingDirection , precision">
```

FpML_Schedule

Description:

An entity for defining a schedule of rate or amounts in terms of an initial value and then a series of step date and value pairs. On each step date the rate or amount changes to the new step value. The series of step date and value pairs are optional. If not specified, this implies that the initial value remains unchanged over time.

Figure:



Contents:

initialValue (exactly one occurrence; of type *decimal*)

- The initial rate or amount, as the case may be. An initial rate of 5% would be represented as 0.05.

step (zero or more occurrences; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_Step)

- The schedule of step date and value pairs. On each step date the associated step value becomes effective A list of steps may be ordered in the document by ascending step date. An FpML document containing an unordered list of steps is still regarded as a conformant document.

Used by:

FpML_AmountSchedule
FpML_StrikeRateSchedule
feeAmountSchedule
feeRateSchedule
fixedRateSchedule
floatingRateMultiplierSchedule
spreadSchedule

DTD Fragment:

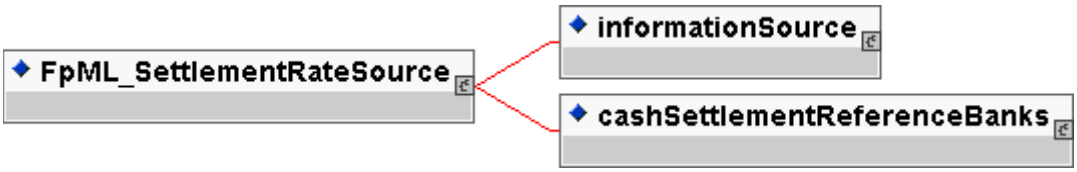
```
<!ENTITY % FpML_Schedule "initialValue , step*">
```

FpML_SettlementRateSource

Description:

An entity to describe the method for obtaining a settlement rate.

Figure:



Contents:

Either

informationSource (exactly one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_InformationSource)

- The information source where a published or displayed market rate will be obtained, e.g. Telerate Page 3750.

Or

cashSettlementReferenceBanks (exactly one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_CashSettlementReferenceBanks)

- A container for a set of reference institutions. These reference institutions may be called upon to provide rate quotations as part of the method to determine the applicable cash settlement amount.

Used by:

settlementRateSource

DTD Fragment:

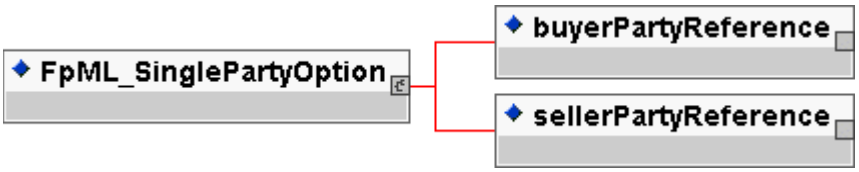
```
<!ENTITY % FpML_SettlementRateSource "informationSource |
cashSettlementReferenceBanks">
```

FpML_SinglePartyOption

Description:

An entity to describe the buyer and seller of a n option.

Figure:



Contents:

buyerPartyReference (exactly one occurrence; an *empty* element containing an *href* attribute)

- A pointer style reference to a party identifier defined elsewhere in the document. The party referenced is the buyer of the instrument, also known as the fixed rate payer.

sellerPartyReference (exactly one occurrence; an *empty* element containing an *href* attribute)

- A pointer style reference to a party identifier defined elsewhere in the document. The party referenced is the seller of the instrument, also known as the floating rate payer.

Used by:

singlePartyOption

DTD Fragment:

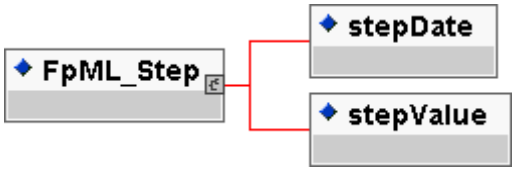
```
<!ENTITY % FpML_SinglePartyOption "buyerPartyReference ,
sellerPartyReference">
```

FpML_Step

Description:

An entity for defining a step date and step value pair. These step definitions are used to define varying rate or amount schedules, e.g. a notional amortization or a step-up coupon schedule.

Figure:



Contents:

stepDate (exactly one occurrence; of type *date*)

- The date on which the associated stepValue becomes effective. This day may be subject to adjustment in accordance with a business day convention.

stepValue (exactly one occurrence; of type *decimal*)

- The rate or amount which becomes effective on the associated stepDate. A rate of 5% would be represented as 0.05.

Used by:

step

DTD Fragment:

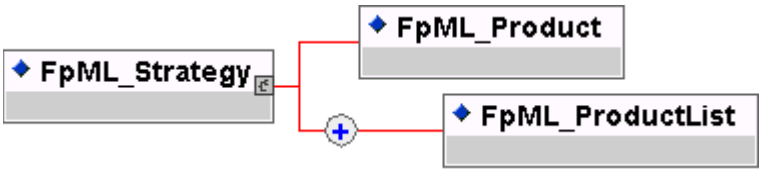
```
<!ENTITY % FpML_Step "stepDate , stepValue">
```

FpML_Strategy

Description:

An entity to define a group of products making up a single trade.

Figure:



Contents:

inherited element(s) (this entity inherits the element(s) defined by exactly one occurrence of the entity FpML_Product)

- The base entity which all FpML products extend.

One or more of:
inherited element(s) (this entity inherits the element(s) defined by exactly one occurrence of the entity FpML_ProductList)

- An entity defining the available product definitions.

Used by:

strategy

DTD Fragment:

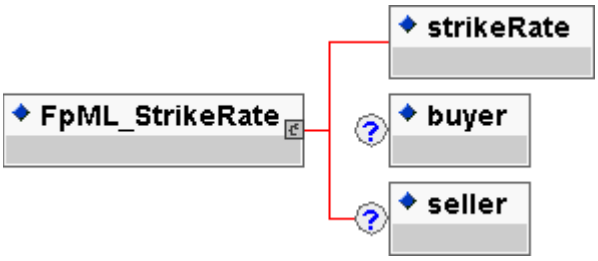
```
<!ENTITY % FpML_Strategy "%FpML_Product;, (%FpML_ProductList;)+">
```

FpML_StrikeRate

Description:

An entity to describe a single cap or floor rate.

Figure:



Contents:

strikeRate (exactly one occurrence; of type *decimal*)

- The rate for a cap or floor.

buyer (zero or one occurrence; of type *string*, an enumerated domain value defined by *payerReceiverScheme*)

- The buyer of the option

seller (zero or one occurrence; of type *string*, an enumerated domain value defined by *payerReceiverScheme*)

- The party that has sold.

Used by:

capRate
floorRate

DTD Fragment:

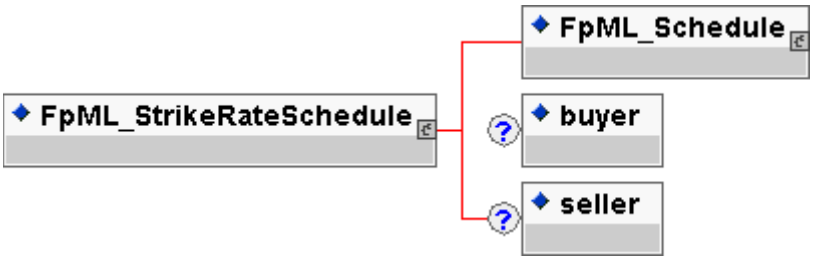
```
<!ENTITY % FpML_StrikeRate "strikeRate , buyer? , seller?">
```

FpML_StrikeRateSchedule

Description:

An entity to describe a schedule of cap or floor rates.

Figure:



Contents:

inherited element(s) (this entity inherits the element(s) defined by exactly one occurrence of the entity FpML_Schedule)

- An entity for defining a schedule of rate or amounts in terms of an initial value and then a series of step date and value pairs. On each step date the rate or amount changes to the new step value. The series of step date and value pairs are optional. If not specified, this implies that the initial value remains unchanged over time.

buyer (zero or one occurrence; of type *string*, an enumerated domain value defined by *payerReceiverScheme*)

- The buyer of the option

seller (zero or one occurrence; of type *string*, an enumerated domain value defined by *payerReceiverScheme*)

- The party that has sold.

Used by:

capRateSchedule
floorRateSchedule

DTD Fragment:

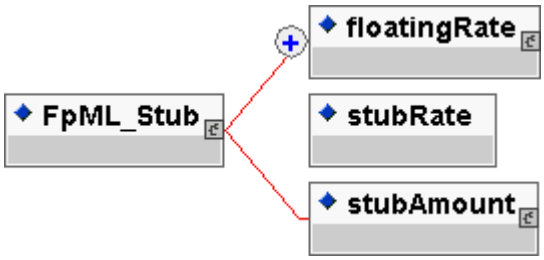
```
<!ENTITY % FpML_StrikeRateSchedule "(%FpML_Schedule; , buyer? , seller?)">
```

FpML_Stub

Description:

An entity for defining how a stub calculation period amount is calculated. A single floating rate tenor different to that used for the regular part of the calculation periods schedule may be specified, or two floating rate tenors may be specified. If two floating rate tenors are specified then Linear Interpolation (in accordance with the 2000 ISDA Definitions, Section 8.3. Interpolation) is assumed to apply. Alternatively, an actual known stub rate or stub amount may be specified.

Figure:



Contents:

Either

floatingRate (one or more occurrences; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_FloatingRate)

- The rates to be applied to the initial or final stub may be the linear interpolation of two different rates. While the majority of the time, the rate indices will be the same as that specified in the stream and only the tenor itself will be different, it is possible to specify two different rates. For example, a 2 month stub period may use the linear interpolation of a 1 month and 3 month rate. The different rates would be specified in this component. Note that a maximum of two rates can be specified. If a stub period uses the same floating rate index, including tenor, as the regular calculation periods then this should not be specified again within this component, i.e. the stub calculation period amount component may not need to be specified even if there is an initial or final stub period. If a stub period uses a different floating rate index compared to the regular calculation periods then this should be specified within this component. If specified here, they are likely to have id attributes, allowing them to be referenced from within the cashflows component.

Or

stubRate (exactly one occurrence; of type *decimal*)

- An actual rate to apply for the initial or final stub period may have been agreed between the principal parties (in a similar way to how an initial rate may have been agreed for the first regular period). If an actual stub rate has been agreed then it would be included in this component. It will be a per annum rate,

expressed as a decimal. A stub rate of 5% would be represented as 0.05.

Or

stubAmount (exactly one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_Money)

- An actual amount to apply for the initial or final stub period may have been agreed between the two parties. If an actual stub amount has been agreed then it would be included in this component.

Used by:

finalStub
initialStub

DTD Fragment:

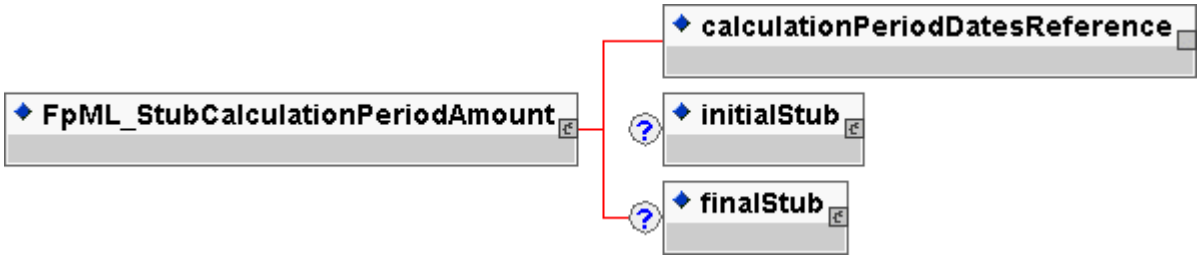
```
<!ENTITY % FpML_Stub "(floatingRate+ | stubRate | stubAmount)">
```

FpML_StubCalculationPeriodAmount

Description:

An entity for defining how the initial or final stub calculation period amounts is calculated. For example, the rate to be applied to the initial or final stub calculation period may be the linear interpolation of two different tenors for the floating rate index specified in the calculation period amount component, e.g. A two month stub period may use the linear interpolation of a one month and three month floating rate. The different rate tenors would be specified in this component. Note that a maximum of two rate tenors can be specified. If a stub period uses a single index tenor and this is the same as that specified in the calculation period amount component then the initial stub or final stub element, as the case may be, must not be included.

Figure:



Contents:

calculationPeriodDatesReference (exactly one occurrence; an *empty* element containing an *href* attribute)

- A pointer style reference to the associated calculation period dates component defined elsewhere in the document.

initialStub (zero or one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_Stub)

- Specifies how the initial stub amount is calculated. A single floating rate tenor different to that used for the regular part of the calculation periods schedule may be specified, or two floating tenors may be specified. If two floating rate tenors are specified then Linear Interpolation (in accordance with the 2000 ISDA Definitions, Section 8.3. Interpolation) is assumed to apply. Alternatively, an actual known stub rate or stub amount may be specified.

finalStub (zero or one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_Stub)

- Specifies how the final stub amount is calculated. A single floating rate tenor different to that used for the regular part of the calculation periods schedule may be specified, or two floating tenors may be specified. If two floating rate tenors are specified then Linear Interpolation (in accordance with the 2000 ISDA Definitions, Section 8.3. Interpolation) is assumed to apply. Alternatively, an actual known stub rate or stub amount may be specified.

Used by:

stubCalculationPeriodAmount

DTD Fragment:

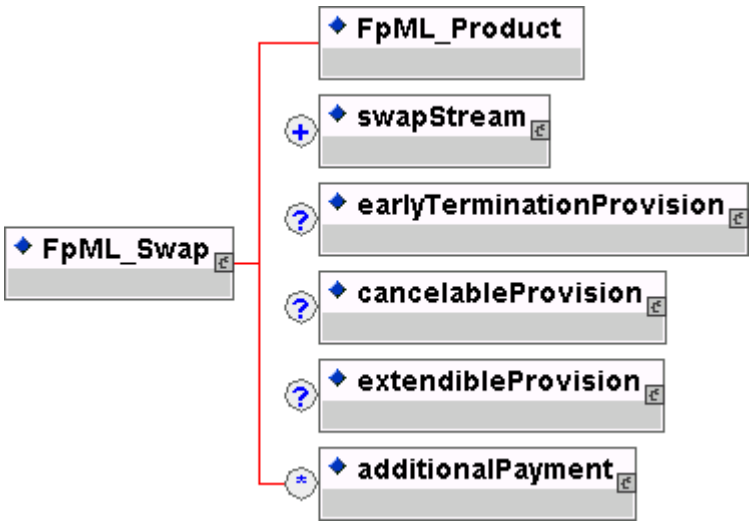
```
<!ENTITY % FpML_StubCalculationPeriodAmount "calculationPeriodDatesReference
, initialStub? , finalStub?">
```

FpML_Swap

Description:

An entity for defining swap streams and additional payments between the principal parties involved in the swap.

Figure:



Contents:

inherited element(s) (this entity inherits the element(s) defined by exactly one occurrence of the entity FpML_Product)

- The base entity which all FpML products extend.

swapStream (one or more occurrences; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_InterestRateStream)

- The swap streams.

earlyTerminationProvision (zero or one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_EarlyTerminationProvision)

- Parameters specifying provisions relating to the optional and mandatory early termination of a swap transaction.

cancelableProvision (zero or one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_CancelableProvision)

- A provision that allows the specification of an embedded option within a swap giving the buyer of the option the right to terminate the swap, in whole or in part, on the early termination date.

extendibleProvision (zero or one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_ExtendibleProvision)

- A provision that allows the specification of an embedded option within a swap giving the buyer of the option the right to extend the swap, in whole or in part, to the extended termination date.

additionalPayment (zero or more occurrences; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_Fee)

- Additional payments between the principal parties involved in the swap.

Used by:

swap

DTD Fragment:

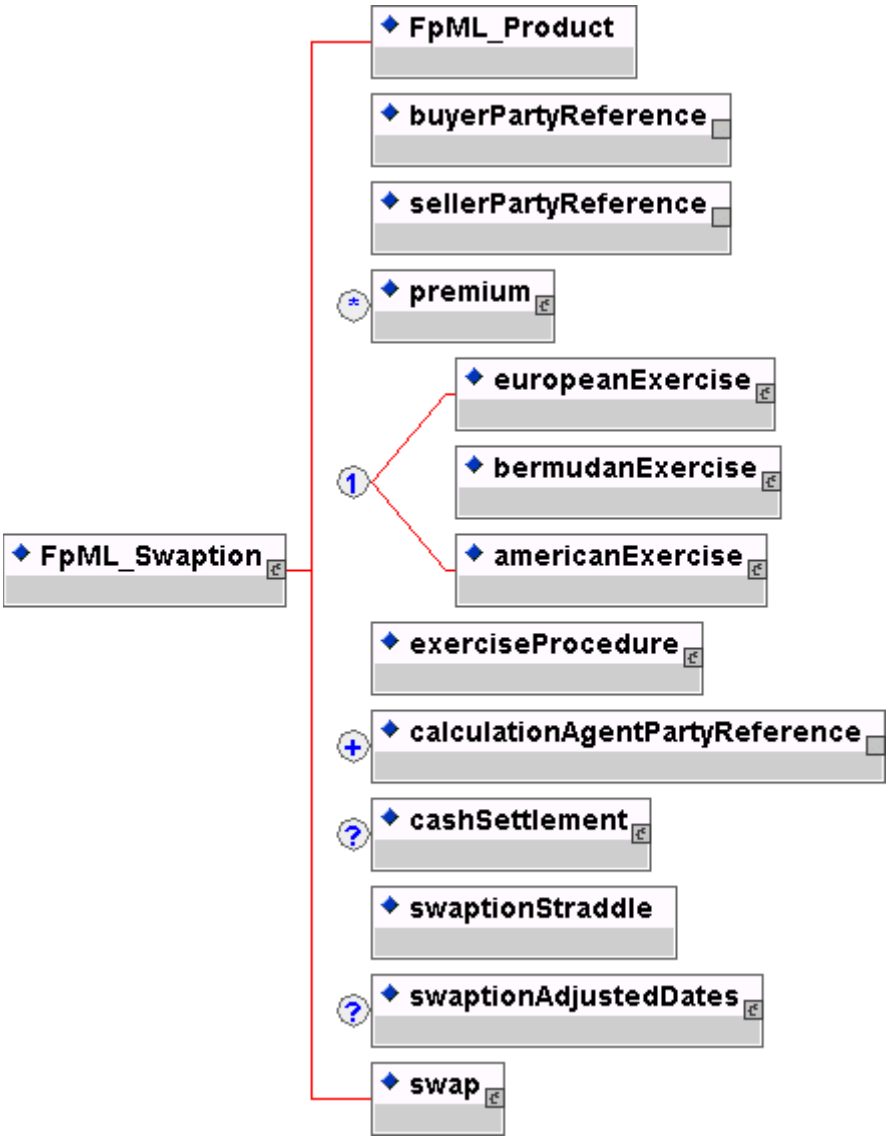
```
<!ENTITY % FpML_Swap "%FpML_Product;,swapStream+ , earlyTerminationProvision?
, cancelableProvision? , extendibleProvision? , additionalPayment*">
```

FpML_Swaption

Description:

An entity to define a option on a swap.

Figure:



Contents:

inherited element(s) (this entity inherits the element(s) defined by exactly one occurrence of the entity FpML_Product)

- The base entity which all FpML products extend.

buyerPartyReference (exactly one occurrence; an *empty* element containing an *href* attribute)

- A pointer style reference to a party identifier defined elsewhere in the document. The party referenced is the buyer of the instrument, also known as the fixed rate payer.

sellerPartyReference (exactly one occurrence; an *empty* element containing an *href* attribute)

- A pointer style reference to a party identifier defined elsewhere in the document. The party referenced is the seller of the instrument, also known as the floating rate payer.

premium (zero or more occurrences; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_Payment)

- The option premium amount payable by buyer to seller on the specified payment date.

Either

europeanExercise (exactly one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_EuropeanExercise)

- The parameters for defining the exercise period for a European style option together with any rules governing the notional amount of the underlying which can be exercised on any given exercise date and any associated exercise fees.

Or

bermudanExercise (exactly one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_BermudanExercise)

- The parameters for defining the exercise period for a Bermudan style option together with any rules governing the notional amount of the underlying which can be exercised on any given exercise date and any associated exercise fees.

Or

americanExercise (exactly one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_AmericanExercise)

- The parameters for defining the exercise period for an American style option together with any rules governing the notional amount of the underlying which can be exercised on any given exercise date and any associated exercise fees.

exerciseProcedure (exactly one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_ExerciseProcedure)

- A set of parameters defining procedures associated with the exercise.

calculationAgentPartyReference (one or more occurrences; an *empty* element containing an *href* attribute)

- A pointer style reference to a party identifier defined elsewhere in the document. The party referenced is the ISDA Calculation Agent for the trade. If more than one party is referenced then the parties are assumed to be co-calculation agents, i.e. they have joint responsibility.

cashSettlement (zero or one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_CashSettlement)

- If specified, this means that cash settlement is applicable to the transaction and defines the parameters associated with the cash settlement procedure. If not specified, then physical settlement is applicable.

swaptionStraddle (exactly one occurrence; of type *boolean*)

- Whether the option is a swaption or a swaption straddle

swaptionAdjustedDates (zero or one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_SwaptionAdjustedDates)

- The adjusted dates associated with swaption exercise. These dates have been adjusted for any applicable business day convention.

swap (exactly one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_Swap)

- A swap product definition.

Used by:

swaption

DTD Fragment:

```
<!ENTITY % FpML_Swaption "%FpML_Product;,buyerPartyReference ,
sellerPartyReference , premium* , (europeanExercise | bermudanExercise |
americanExercise) , exerciseProcedure , calculationAgentPartyReference+ ,
cashSettlement? , swaptionStraddle , swaptionAdjustedDates? , swap">
```

FpML_SwaptionAdjustedDates

Description:

An entity to describe the adjusted dates associated with swaption exercise and settlement.

Figure:

Contents:

exerciseEvent (one or more occurrences; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_ExerciseEvent)

- The adjusted dates associated with an individual swaption exercise date.

Used by:

swaptionAdjustedDates

DTD Fragment:

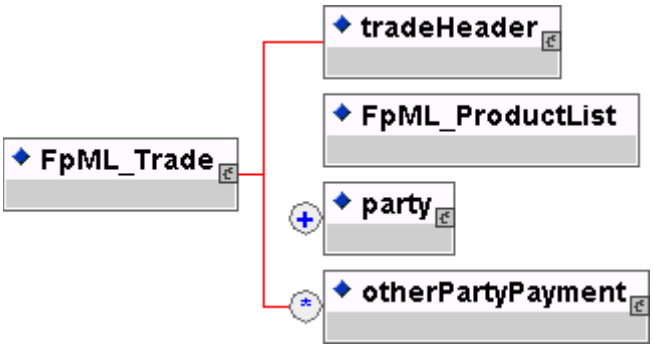
```
<!ENTITY % FpML_SwaptionAdjustedDates "exerciseEvent+">
```

FpML_Trade

Description:

An entity for defining an FpML trade.

Figure:



Contents:

tradeHeader (exactly one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_TradeHeader)

- The information on the trade which is not product specific, e.g. trade date.

inherited element(s) (this entity inherits the element(s) defined by exactly one occurrence of the entity FpML_ProductList)

- An entity defining the available product definitions.

party (one or more occurrences; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_Party)

- The parties obligated to make payments from time to time during the term of the trade. This will include, at a minimum, the principal parties involved in the swap or forward rate agreement. Other parties paying or receiving fees, commissions etc. must also be specified if referenced in other party payments.

otherPartyPayment (zero or more occurrences; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_Fee)

- Other fees or additional payments associated with the trade, e.g. broker commissions, where one or more of the parties involved are not principal parties involved in the trade.

Used by:

trade

DTD Fragment:

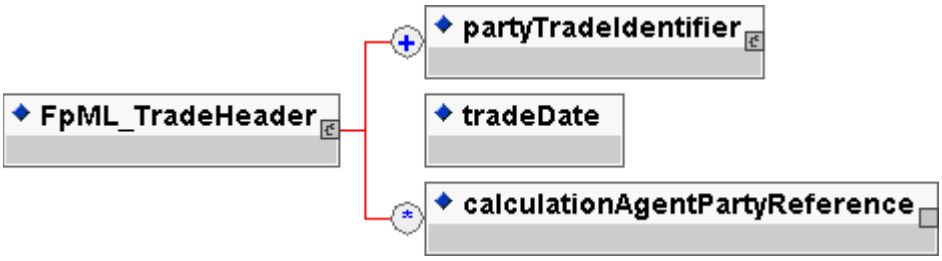
```
<!ENTITY % FpML_Trade "tradeHeader , %FpML_ProductList; , party+ ,
otherPartyPayment*">
```

FpML_TradeHeader

Description:

An entity for defining trade related information which is not product specific.

Figure:



Contents:

partyTradeIdentifier (one or more occurrences; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_PartyTradeIdentifier)

- The trade reference identifier(s) allocated to the trade by the parties involved.

tradeDate (exactly one occurrence; of type *date*)

- The trade date.

calculationAgentPartyReference (zero or more occurrences; an *empty* element containing an *href* attribute)

- A pointer style reference to a party identifier defined elsewhere in the document. The party referenced is the ISDA Calculation Agent for the trade. If more than one party is referenced then the parties are assumed to be co-calculation agents, i.e. they have joint responsibility.

Used by:

tradeHeader

DTD Fragment:

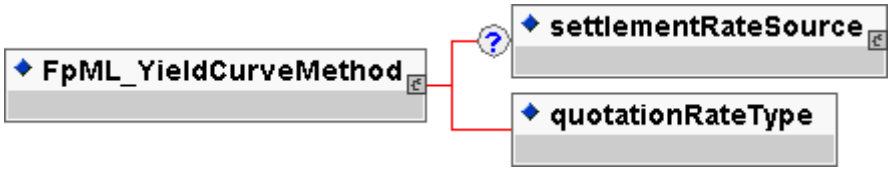
```
<!ENTITY % FpML_TradeHeader "partyTradeIdentifier+ , tradeDate ,
calculationAgentPartyReference*">
```

FpML_YieldCurveMethod

Description:

An entity to define the parameters required for each of the ISDA defined yield curve methods for cash settlement.

Figure:



Contents:

settlementRateSource (zero or one occurrence; contains the sub-element(s) defined by exactly one occurrence of the entity FpML_SettlementRateSource)

- The method for obtaining a settlement rate. This may be from some information source (e.g. Reuters) or from a set of reference banks.

quotationRateType (exactly one occurrence; of type *string*, an enumerated domain value defined by *quotationRateTypeScheme*)

- Which rate quote is to be observed, either Bid, Mid, Offer or Exercising Party Pays. The meaning of Exercising Party Pays is defined in the 2000 ISDA Definitions, Section 17.2. Certain Definitions Relating to Cash Settlement, paragraph (j)

Used by:

parYieldCurveAdjustedMethod
parYieldCurveUnadjustedMethod
zeroCouponYieldAdjustedMethod

DTD Fragment:

```
<!ENTITY % FpML_YieldCurveMethod "settlementRateSource? , quotationRateType">
```

6 DOCUMENT TYPE DEFINITION (DTD)

6.1 fpml-dtd-2-0

```
<?xml version='1.0' encoding='UTF-8' ?>

<!-- uri://www.fpml.org/spec/2001/fpml-dtd-2-0-2001-10-17 -->
<!ENTITY % FpML_AdjustableDate "unadjustedDate , dateAdjustments">

<!ENTITY % FpML_BusinessCenters "businessCenter+">

<!ENTITY % FpML_BusinessDayAdjustments "businessDayConvention , (businessCentersReference
| businessCenters)?">

<!ENTITY % FpML_Calculation "((notionalSchedule | fxLinkedNotionalSchedule) ,
(fixedRateSchedule | floatingRateCalculation) , dayCountFraction , discounting? ,
compoundingMethod?)">

<!ENTITY % FpML_CalculationPeriod "adjustedStartDate , adjustedEndDate , (notionalAmount
| fxLinkedNotionalAmount) , (floatingRateDefinition | fixedRate)">

<!ENTITY % FpML_CalculationPeriodAmount "calculation | knownAmountSchedule">

<!ENTITY % FpML_CalculationPeriodDates "effectiveDate , terminationDate ,
calculationPeriodDatesAdjustments , firstPeriodStartDate? , firstRegularPeriodStartDate?
, lastRegularPeriodEndDate? , calculationPeriodFrequency">

<!ENTITY % FpML_Cashflows "cashflowsMatchParameters , principalExchange* ,
paymentCalculationPeriod*">

<!ENTITY % FpML_Discounting "discountingType , discountRate? ,
discountRateDayCountFraction?">

<!ENTITY % FpML_Payment "payerPartyReference , receiverPartyReference , paymentAmount ,
paymentDate? , adjustedPaymentDate?">

<!ENTITY % FpML_Fee "%FpML_Payment; , paymentType?">

<!ENTITY % FpML_FloatingRate "floatingRateIndex , indexTenor? ,
floatingRateMultiplierSchedule? , spreadSchedule? , rateTreatment? , capRateSchedule* ,
floorRateSchedule*">

<!ENTITY % FpML_FloatingRateCalculation "(%FpML_FloatingRate; , initialRate? ,
finalRateRounding? , averagingMethod? , negativeInterestRateTreatment?)">

<!ENTITY % FpML_FloatingRateDefinition "calculatedRate? , rateObservation* ,
floatingRateMultiplier? , spread? , capRate* , floorRate*">

<!ENTITY % FpML_Fra "%FpML_Product; , buyerPartyReference , sellerPartyReference ,
adjustedEffectiveDate , adjustedTerminationDate , paymentDate , fixingDateOffset ,
dayCountFraction , calculationPeriodNumberOfDays , notional , fixedRate ,
floatingRateIndex , indexTenor+ , fraDiscounting">

<!ENTITY % FpML_InterestRateStream "payerPartyReference , receiverPartyReference ,
calculationPeriodDates , paymentDates , resetDates? , calculationPeriodAmount ,
stubCalculationPeriodAmount? , principalExchanges? , cashflows?">

<!ENTITY % FpML_Interval "periodMultiplier , period">

<!ENTITY % FpML_CalculationPeriodFrequency "(%FpML_Interval; , rollConvention)">

<!ENTITY % FpML_Money "currency , amount">

<!ENTITY % FpML_Notional "notionalStepSchedule , notionalStepParameters?">
```

```
<!ENTITY % FpML_NotionalStepRule "calculationPeriodDatesReference , stepFrequency ,
firstNotionalStepDate , lastNotionalStepDate , (notionalStepAmount | (notionalStepRate ,
stepRelativeTo))">

<!ENTITY % FpML_Offset "(%FpML_Interval; , dayType?)">

<!ENTITY % FpML_Party "partyId , partyName?">

<!ENTITY % FpML_PartyTradeIdentifier "partyReference , tradeId+ , linkId*">

<!ENTITY % FpML_PaymentCalculationPeriod "adjustedPaymentDate , (calculationPeriod+ |
fixedPaymentAmount)">

<!ENTITY % FpML_PaymentDates "((calculationPeriodDatesReference | resetDatesReference) ,
paymentFrequency , firstPaymentDate? , lastRegularPaymentDate? , payRelativeTo ,
paymentDaysOffset? , paymentDatesAdjustments)">

<!ENTITY % FpML_PrincipalExchange "adjustedPrincipalExchangeDate ,
principalExchangeAmount?">

<!ENTITY % FpML_PrincipalExchanges "initialExchange , finalExchange ,
intermediateExchange">

<!ENTITY % FpML_Product "productType?">

<!ENTITY % FpML_ProductList "(bulletPayment | capFloor | fra | swap | swaption |
strategy)">

<!ENTITY % FpML_Strategy "%FpML_Product; , (%FpML_ProductList;)+">

<!ENTITY % FpML_RateObservation "adjustedFixingDate , observedRate? , treatedRate? ,
observationWeight , rateReference?">

<!ENTITY % FpML_RelativeDateOffset "(%FpML_Offset; , businessDayConvention ,
(businessCentersReference | businessCenters)? , dateRelativeTo)">

<!ENTITY % FpML_ResetDates "calculationPeriodDatesReference , resetRelativeTo? ,
fixingDates , rateCutOffDaysOffset? , resetFrequency , resetDatesAdjustments">

<!ENTITY % FpML_ResetFrequency "(%FpML_Interval; , weeklyRollConvention?)">

<!ENTITY % FpML_Rounding "roundingDirection , precision">

<!ENTITY % FpML_Schedule "initialValue , step*">

<!ENTITY % FpML_AmountSchedule "(%FpML_Schedule; , currency)">

<!ENTITY % FpML_Step "stepDate , stepValue">

<!ENTITY % FpML_Stub "(floatingRate+ | stubRate | stubAmount)">

<!ENTITY % FpML_StubCalculationPeriodAmount "calculationPeriodDatesReference ,
initialStub? , finalStub?">

<!ENTITY % FpML_Swap "%FpML_Product; , swapStream+ , earlyTerminationProvision? ,
cancelableProvision? , extendibleProvision? , additionalPayment*">

<!ENTITY % FpML_Trade "tradeHeader , %FpML_ProductList; , party+ , otherPartyPayment*">

<!ENTITY % FpML_TradeHeader "partyTradeIdentifier+ , tradeDate ,
calculationAgentPartyReference*">

<!-- new version 2.0 entities required by FpML 1.0 elements below -->
<!ENTITY % FpML_StrikeRate "strikeRate , buyer? , seller?">

<!ENTITY % FpML_StrikeRateSchedule "(%FpML_Schedule; , buyer? , seller?)">

<!ELEMENT FpML (trade)>

<!ATTLIST FpML  version                                  (2-0 )  #REQUIRED
                  averagingMethodSchemeDefault            CDATA  #IMPLIED
```

```
                  businessCenterSchemeDefault                CDATA  #IMPLIED
                  businessDayConventionSchemeDefault         CDATA  #IMPLIED
                  calculationAgentPartySchemeDefault         CDATA  #IMPLIED
                  compoundingMethodSchemeDefault             CDATA  #IMPLIED
                  currencySchemeDefault                      CDATA  #IMPLIED
                  dateRelativeToSchemeDefault                CDATA  #IMPLIED
                  dayCountFractionSchemeDefault              CDATA  #IMPLIED
                  dayTypeSchemeDefault                       CDATA  #IMPLIED
                  discountingTypeSchemeDefault               CDATA  #IMPLIED
                  floatingRateIndexSchemeDefault             CDATA  #IMPLIED
                  informationProviderSchemeDefault           CDATA  #IMPLIED
                  linkIdSchemeDefault                        CDATA  #IMPLIED
                  negativeInterestRateTreatmentSchemeDefault CDATA  #IMPLIED
                  partyIdSchemeDefault                       CDATA  #IMPLIED
                  payerReceiverSchemeDefault                 CDATA  #IMPLIED
                  paymentTypeSchemeDefault                   CDATA  #IMPLIED
                  payRelativeToSchemeDefault                 CDATA  #IMPLIED
                  periodSchemeDefault                        CDATA  #IMPLIED
                  productTypeSchemeDefault                   CDATA  #IMPLIED
                  quotationRateTypeSchemeDefault             CDATA  #IMPLIED
                  rateSourcePageSchemeDefault                CDATA  #IMPLIED
                  rateTreatmentSchemeDefault                 CDATA  #IMPLIED
                  referenceBankIdSchemeDefault               CDATA  #IMPLIED
                  resetRelativeToSchemeDefault               CDATA  #IMPLIED
                  rollConventionSchemeDefault                CDATA  #IMPLIED
                  roundingDirectionSchemeDefault             CDATA  #IMPLIED
                  stepRelativeToSchemeDefault                CDATA  #IMPLIED
                  tradeIdSchemeDefault                       CDATA  #IMPLIED
                  weeklyRollConventionSchemeDefault          CDATA  #IMPLIED >
<!ELEMENT additionalPayment (%FpML_Fee;)>

<!ATTLIST additionalPayment  type NMTOKEN  #FIXED 'Fee'
                             id   ID       #IMPLIED >
<!ELEMENT adjustedEffectiveDate (#PCDATA)>

<!ATTLIST adjustedEffectiveDate  type NMTOKEN  #FIXED 'date'
                                 id   ID       #REQUIRED >
<!ELEMENT adjustedEndDate (#PCDATA)>

<!ATTLIST adjustedEndDate  type NMTOKEN  #FIXED 'date' >
<!ELEMENT adjustedFixingDate (#PCDATA)>

<!ATTLIST adjustedFixingDate  type NMTOKEN  #FIXED 'date' >
<!ELEMENT adjustedPaymentDate (#PCDATA)>

<!ATTLIST adjustedPaymentDate  type NMTOKEN  #FIXED 'date' >
<!ELEMENT adjustedPrincipalExchangeDate (#PCDATA)>

<!ATTLIST adjustedPrincipalExchangeDate  type NMTOKEN  #FIXED 'date' >
<!ELEMENT adjustedStartDate (#PCDATA)>

<!ATTLIST adjustedStartDate  type NMTOKEN  #FIXED 'date' >
<!ELEMENT adjustedTerminationDate (#PCDATA)>

<!ATTLIST adjustedTerminationDate  type NMTOKEN  #FIXED 'date' >
<!ELEMENT amount (#PCDATA)>

<!ATTLIST amount  type NMTOKEN  #FIXED 'decimal' >
<!ELEMENT averagingMethod (#PCDATA)>

<!ATTLIST averagingMethod  type                   NMTOKEN  #FIXED 'string'
                           averagingMethodScheme CDATA    #IMPLIED >
<!ELEMENT businessCenter (#PCDATA)>

<!ATTLIST businessCenter  type                 NMTOKEN  #FIXED 'string'
                          id                   ID       #IMPLIED
                          businessCenterScheme CDATA    #IMPLIED >
<!ELEMENT businessCenters (%FpML_BusinessCenters;)>

<!ATTLIST businessCenters  type NMTOKEN  #FIXED 'BusinessCenters'
                           id   ID       #IMPLIED >
```

```
<!ELEMENT businessCentersReference EMPTY>

<!ATTLIST businessCentersReference  href CDATA  #IMPLIED >
<!ELEMENT businessDayConvention (#PCDATA)>

<!ATTLIST businessDayConvention  type                        NMTOKEN  #FIXED 'string'
                                 businessDayConventionScheme CDATA    #IMPLIED >
<!ELEMENT buyerPartyReference EMPTY>

<!ATTLIST buyerPartyReference  href CDATA  #REQUIRED >
<!ELEMENT calculatedRate (#PCDATA)>

<!ATTLIST calculatedRate  type NMTOKEN  #FIXED 'decimal' >
<!ELEMENT calculation (%FpML_Calculation;)>

<!ATTLIST calculation  type NMTOKEN  #FIXED 'Calculation' >
<!ELEMENT calculationAgentPartyReference EMPTY>

<!ATTLIST calculationAgentPartyReference  href CDATA  #REQUIRED >
<!ELEMENT calculationPeriod (%FpML_CalculationPeriod;)>

<!ATTLIST calculationPeriod  type NMTOKEN  #FIXED 'CalculationPeriod'
                             id   ID       #IMPLIED >
<!ELEMENT calculationPeriodAmount (%FpML_CalculationPeriodAmount;)>

<!ATTLIST calculationPeriodAmount  type NMTOKEN  #FIXED 'CalculationPeriodAmount' >
<!ELEMENT calculationPeriodDates (%FpML_CalculationPeriodDates;)>

<!ATTLIST calculationPeriodDates  type NMTOKEN  #FIXED 'CalculationPeriodDates'
                                  id   ID       #REQUIRED >
<!ELEMENT calculationPeriodDatesAdjustments (%FpML_BusinessDayAdjustments;)>

<!ATTLIST calculationPeriodDatesAdjustments  type NMTOKEN  #FIXED
'BusinessDayAdjustments' >
<!ELEMENT calculationPeriodDatesReference EMPTY>

<!ATTLIST calculationPeriodDatesReference  href CDATA  #REQUIRED >
<!ELEMENT calculationPeriodFrequency (%FpML_CalculationPeriodFrequency;)>

<!ATTLIST calculationPeriodFrequency  type NMTOKEN  #FIXED 'CalculationPeriodFrequency'
                                      base NMTOKEN  #FIXED 'Interval' >
<!ELEMENT calculationPeriodNumberOfDays (#PCDATA)>

<!ATTLIST calculationPeriodNumberOfDays  type NMTOKEN  #FIXED 'positiveInteger' >
<!ELEMENT capRate (%FpML_StrikeRate;)>

<!ATTLIST capRate  type NMTOKEN  #FIXED 'StrikeRate'
                   id   ID       #IMPLIED >
<!ELEMENT capRateSchedule (%FpML_StrikeRateSchedule;)>

<!ATTLIST capRateSchedule  type NMTOKEN  #FIXED 'StrikeRateSchedule'
                           base NMTOKEN  #FIXED 'Schedule'
                           id   ID       #IMPLIED >
<!ELEMENT cashflows (%FpML_Cashflows;)>

<!ATTLIST cashflows  type NMTOKEN  #FIXED 'Cashflows' >
<!ELEMENT cashflowsMatchParameters (#PCDATA)>

<!ATTLIST cashflowsMatchParameters  type NMTOKEN  #FIXED 'boolean' >
<!ELEMENT compoundingMethod (#PCDATA)>

<!ATTLIST compoundingMethod  type                    NMTOKEN  #FIXED 'string'
                             compoundingMethodScheme CDATA    #IMPLIED >
<!ELEMENT currency (#PCDATA)>

<!ATTLIST currency  type           NMTOKEN  #FIXED 'string'
                    currencyScheme CDATA    #IMPLIED >
<!ELEMENT dateAdjustments (%FpML_BusinessDayAdjustments;)>

<!ATTLIST dateAdjustments  type NMTOKEN  #FIXED 'BusinessDayAdjustments' >
<!ELEMENT dateRelativeTo (#PCDATA)>
```

```
<!ATTLIST dateRelativeTo  type                NMTOKEN  #FIXED 'string'
                          href                  CDATA    #REQUIRED
                          dateRelativeToScheme CDATA    #IMPLIED >
<!ELEMENT dayCountFraction (#PCDATA)>

<!ATTLIST dayCountFraction  type                   NMTOKEN  #FIXED 'string'
                            dayCountFractionScheme CDATA    #IMPLIED >
<!ELEMENT dayType (#PCDATA)>

<!ATTLIST dayType  type          NMTOKEN  #FIXED 'string'
                   dayTypeScheme CDATA    #IMPLIED >
<!ELEMENT discounting (%FpML_Discounting;)>

<!ATTLIST discounting  type NMTOKEN  #FIXED 'Discounting' >
<!ELEMENT discountingType (#PCDATA)>

<!ATTLIST discountingType  type                   NMTOKEN  #FIXED 'string'
                           discountingTypeScheme CDATA    #IMPLIED >
<!ELEMENT discountRate (#PCDATA)>

<!ATTLIST discountRate  type NMTOKEN  #FIXED 'decimal' >
<!ELEMENT discountRateDayCountFraction (#PCDATA)>

<!ATTLIST discountRateDayCountFraction  type                   NMTOKEN  #FIXED 'string'
                                        dayCountFractionScheme CDATA    #IMPLIED >
<!ELEMENT effectiveDate (%FpML_AdjustableDate;)>

<!ATTLIST effectiveDate  type NMTOKEN  #FIXED 'AdjustableDate' >
<!ELEMENT finalExchange (#PCDATA)>

<!ATTLIST finalExchange  type NMTOKEN  #FIXED 'boolean' >
<!ELEMENT finalRateRounding (%FpML_Rounding;)>

<!ATTLIST finalRateRounding  type NMTOKEN  #FIXED 'Rounding' >
<!ELEMENT finalStub (%FpML_Stub;)>

<!ATTLIST finalStub  type NMTOKEN  #FIXED 'Stub' >
<!ELEMENT firstNotionalStepDate (#PCDATA)>

<!ATTLIST firstNotionalStepDate  type NMTOKEN  #FIXED 'date' >
<!ELEMENT firstPaymentDate (#PCDATA)>

<!ATTLIST firstPaymentDate  type NMTOKEN  #FIXED 'date' >
<!ELEMENT firstPeriodStartDate (%FpML_AdjustableDate;)>

<!ATTLIST firstPeriodStartDate  type NMTOKEN  #FIXED 'AdjustableDate' >
<!ELEMENT firstRegularPeriodStartDate (#PCDATA)>

<!ATTLIST firstRegularPeriodStartDate  type NMTOKEN  #FIXED 'date' >
<!ELEMENT fixedPaymentAmount (#PCDATA)>

<!ATTLIST fixedPaymentAmount  type NMTOKEN  #FIXED 'decimal' >
<!ELEMENT fixedRate (#PCDATA)>

<!ATTLIST fixedRate  type NMTOKEN  #FIXED 'decimal' >
<!ELEMENT fixedRateSchedule (%FpML_Schedule;)>

<!ATTLIST fixedRateSchedule  type NMTOKEN  #FIXED 'Schedule' >
<!ELEMENT fixingDateOffset (%FpML_RelativeDateOffset;)>

<!ATTLIST fixingDateOffset  type NMTOKEN  #FIXED 'RelativeDateOffset'
                            base NMTOKEN  #FIXED 'Offset' >
<!ELEMENT fixingDates (%FpML_RelativeDateOffset;)>

<!ATTLIST fixingDates  type NMTOKEN  #FIXED 'RelativeDateOffset'
                       base NMTOKEN  #FIXED 'Offset' >
<!ELEMENT floatingRate (%FpML_FloatingRate;)>

<!ATTLIST floatingRate  type NMTOKEN  #FIXED 'FloatingRate'
                        id   ID       #IMPLIED >
```

```
<!ELEMENT floatingRateCalculation (%FpML_FloatingRateCalculation;)>

<!ATTLIST floatingRateCalculation  type NMTOKEN  #FIXED 'FloatingRateCalculation'
                                   base NMTOKEN  #FIXED 'FloatingRate' >
<!ELEMENT floatingRateDefinition (%FpML_FloatingRateDefinition;)>

<!ATTLIST floatingRateDefinition  type NMTOKEN  #FIXED 'FloatingRateDefinition' >
<!ELEMENT floatingRateIndex (#PCDATA)>

<!ATTLIST floatingRateIndex  type                    NMTOKEN  #FIXED 'string'
                             floatingRateIndexScheme CDATA    #IMPLIED >
<!ELEMENT floorRate (%FpML_StrikeRate;)>

<!ATTLIST floorRate  type NMTOKEN  #FIXED 'StrikeRate'
                     id   ID       #IMPLIED >
<!ELEMENT floorRateSchedule (%FpML_StrikeRateSchedule;)>

<!ATTLIST floorRateSchedule  type NMTOKEN  #FIXED 'StrikeRateSchedule'
                             base NMTOKEN  #FIXED 'Schedule'
                             id   ID       #IMPLIED >
<!ELEMENT fra (%FpML_Fra;)>

<!ATTLIST fra  type NMTOKEN  #FIXED 'Fra'
               id   ID       #IMPLIED >
<!ELEMENT fraDiscounting (#PCDATA)>

<!ATTLIST fraDiscounting  type NMTOKEN  #FIXED 'boolean' >
<!ELEMENT indexTenor (%FpML_Interval;)>

<!ATTLIST indexTenor  type NMTOKEN  #FIXED 'Interval'
                      id   ID       #IMPLIED >
<!ELEMENT initialExchange (#PCDATA)>

<!ATTLIST initialExchange  type NMTOKEN  #FIXED 'boolean' >
<!ELEMENT initialRate (#PCDATA)>

<!ATTLIST initialRate  type NMTOKEN  #FIXED 'decimal' >
<!ELEMENT initialStub (%FpML_Stub;)>

<!ATTLIST initialStub  type NMTOKEN  #FIXED 'Stub' >
<!ELEMENT initialValue (#PCDATA)>

<!ATTLIST initialValue  type NMTOKEN  #FIXED 'decimal' >
<!ELEMENT intermediateExchange (#PCDATA)>

<!ATTLIST intermediateExchange  type NMTOKEN  #FIXED 'boolean' >
<!ELEMENT knownAmountSchedule (%FpML_AmountSchedule;)>

<!ATTLIST knownAmountSchedule  type NMTOKEN  #FIXED 'AmountSchedule'
                               base NMTOKEN  #FIXED 'Schedule' >
<!ELEMENT lastNotionalStepDate (#PCDATA)>

<!ATTLIST lastNotionalStepDate  type NMTOKEN  #FIXED 'date' >
<!ELEMENT lastRegularPaymentDate (#PCDATA)>

<!ATTLIST lastRegularPaymentDate  type NMTOKEN  #FIXED 'date' >
<!ELEMENT lastRegularPeriodEndDate (#PCDATA)>

<!ATTLIST lastRegularPeriodEndDate  type NMTOKEN  #FIXED 'date' >
<!ELEMENT linkId (#PCDATA)>

<!ATTLIST linkId  type         NMTOKEN  #FIXED 'string'
                  id           ID       #IMPLIED
                  linkIdScheme CDATA    #IMPLIED >
<!ELEMENT negativeInterestRateTreatment (#PCDATA)>

<!ATTLIST negativeInterestRateTreatment  type                                NMTOKEN
#FIXED 'string'
                                         negativeInterestRateTreatmentScheme CDATA
#IMPLIED >
<!ELEMENT notional (%FpML_Money;)>
```

```
<!ATTLIST notional  type NMTOKEN  #FIXED 'Money' >
<!ELEMENT notionalAmount (#PCDATA)>

<!ATTLIST notionalAmount  type NMTOKEN  #FIXED 'decimal' >
<!ELEMENT notionalSchedule (%FpML_Notional;)>

<!ATTLIST notionalSchedule  type NMTOKEN  #FIXED 'Notional'
                            id   ID       #IMPLIED >
<!ELEMENT notionalStepAmount (#PCDATA)>

<!ATTLIST notionalStepAmount  type NMTOKEN  #FIXED 'decimal' >
<!ELEMENT notionalStepParameters (%FpML_NotionalStepRule;)>

<!ATTLIST notionalStepParameters  type NMTOKEN  #FIXED 'NotionalStepRule' >
<!ELEMENT notionalStepRate (#PCDATA)>

<!ATTLIST notionalStepRate  type NMTOKEN  #FIXED 'decimal' >
<!ELEMENT notionalStepSchedule (%FpML_AmountSchedule;)>

<!ATTLIST notionalStepSchedule  type NMTOKEN  #FIXED 'AmountSchedule'
                                base NMTOKEN  #FIXED 'Schedule' >
<!ELEMENT observationWeight (#PCDATA)>

<!ATTLIST observationWeight  type NMTOKEN  #FIXED 'positiveInteger' >
<!ELEMENT observedRate (#PCDATA)>

<!ATTLIST observedRate  type NMTOKEN  #FIXED 'decimal' >
<!ELEMENT otherPartyPayment (%FpML_Fee;)>

<!ATTLIST otherPartyPayment  type NMTOKEN  #FIXED 'Fee'
                             id   ID       #IMPLIED >
<!ELEMENT party (%FpML_Party;)>

<!ATTLIST party  type NMTOKEN  #FIXED 'Party'
                 id   ID       #REQUIRED >
<!ELEMENT partyId (#PCDATA)>

<!ATTLIST partyId  type          NMTOKEN  #FIXED 'string'
                   partyIdScheme CDATA    #IMPLIED >
<!ELEMENT partyName (#PCDATA)>

<!ATTLIST partyName  type NMTOKEN  #FIXED 'string' >
<!ELEMENT partyReference EMPTY>

<!ATTLIST partyReference  href CDATA  #REQUIRED >
<!ELEMENT partyTradeIdentifier (%FpML_PartyTradeIdentifier;)>

<!ATTLIST partyTradeIdentifier  type NMTOKEN  #FIXED 'PartyTradeIdentifier'
                                id   ID       #IMPLIED >
<!ELEMENT payerPartyReference EMPTY>

<!ATTLIST payerPartyReference  href CDATA  #REQUIRED >
<!ELEMENT paymentAmount (%FpML_Money;)>

<!ATTLIST paymentAmount  type NMTOKEN  #FIXED 'Money' >
<!ELEMENT paymentCalculationPeriod (%FpML_PaymentCalculationPeriod;)>

<!ATTLIST paymentCalculationPeriod  type NMTOKEN  #FIXED 'PaymentCalculationPeriod'
                                    id   ID       #IMPLIED >
<!ELEMENT paymentDate (%FpML_AdjustableDate;)>

<!ATTLIST paymentDate  type NMTOKEN  #FIXED 'AdjustableDate'
                       id   ID       #IMPLIED >
<!ELEMENT paymentDates (%FpML_PaymentDates;)>

<!ATTLIST paymentDates  type NMTOKEN  #FIXED 'PaymentDates'
                        id   ID       #IMPLIED >
<!ELEMENT paymentDatesAdjustments (%FpML_BusinessDayAdjustments;)>

<!ATTLIST paymentDatesAdjustments  type NMTOKEN  #FIXED 'BusinessDayAdjustments' >
```

```
<!ELEMENT paymentDaysOffset (%FpML_Offset;)>

<!ATTLIST paymentDaysOffset  type NMTOKEN  #FIXED 'Offset'
                             base NMTOKEN  #FIXED 'Interval' >
<!ELEMENT paymentFrequency (%FpML_Interval;)>

<!ATTLIST paymentFrequency  type NMTOKEN  #FIXED 'Interval' >
<!ELEMENT paymentType (#PCDATA)>

<!ATTLIST paymentType  type              NMTOKEN  #FIXED 'string'
                       paymentTypeScheme CDATA    #IMPLIED >
<!ELEMENT payRelativeTo (#PCDATA)>

<!ATTLIST payRelativeTo  type                NMTOKEN  #FIXED 'string'
                         payRelativeToScheme CDATA    #IMPLIED >
<!ELEMENT period (#PCDATA)>

<!ATTLIST period  type         NMTOKEN  #FIXED 'string'
                  periodScheme CDATA    #IMPLIED >
<!ELEMENT periodMultiplier (#PCDATA)>

<!ATTLIST periodMultiplier  type NMTOKEN  #FIXED 'integer' >
<!ELEMENT precision (#PCDATA)>

<!ATTLIST precision  type NMTOKEN  #FIXED 'nonNegativeInteger' >
<!ELEMENT principalExchange (%FpML_PrincipalExchange;)>

<!ATTLIST principalExchange  type NMTOKEN  #FIXED 'PrincipalExchange'
                             id   ID       #IMPLIED >
<!ELEMENT principalExchangeAmount (#PCDATA)>

<!ATTLIST principalExchangeAmount  type NMTOKEN  #FIXED 'decimal' >
<!ELEMENT principalExchanges (%FpML_PrincipalExchanges;)>

<!ATTLIST principalExchanges  type NMTOKEN  #FIXED 'PrincipalExchanges' >
<!ELEMENT rateCutOffDaysOffset (%FpML_Offset;)>

<!ATTLIST rateCutOffDaysOffset  type NMTOKEN  #FIXED 'Offset'
                                base NMTOKEN  #FIXED 'Interval' >
<!ELEMENT rateObservation (%FpML_RateObservation;)>

<!ATTLIST rateObservation  type NMTOKEN  #FIXED 'RateObservation'
                           id   ID       #IMPLIED >
<!ELEMENT rateReference EMPTY>

<!ATTLIST rateReference  href CDATA  #REQUIRED >
<!ELEMENT rateTreatment (#PCDATA)>

<!ATTLIST rateTreatment  type               NMTOKEN  #FIXED 'string'
                         rateTreatmentScheme CDATA    #IMPLIED >
<!ELEMENT receiverPartyReference EMPTY>

<!ATTLIST receiverPartyReference  href CDATA  #REQUIRED >
<!ELEMENT resetDates (%FpML_ResetDates;)>

<!ATTLIST resetDates  type NMTOKEN  #FIXED 'ResetDates'
                      id   ID       #REQUIRED >
<!ELEMENT resetDatesAdjustments (%FpML_BusinessDayAdjustments;)>

<!ATTLIST resetDatesAdjustments  type NMTOKEN  #FIXED 'BusinessDayAdjustments' >
<!ELEMENT resetDatesReference EMPTY>

<!ATTLIST resetDatesReference  href CDATA  #REQUIRED >
<!ELEMENT resetFrequency (%FpML_ResetFrequency;)>

<!ATTLIST resetFrequency  type NMTOKEN  #FIXED 'ResetFrequency'
                          base NMTOKEN  #FIXED 'Interval' >
<!ELEMENT resetRelativeTo (#PCDATA)>

<!ATTLIST resetRelativeTo  type                 NMTOKEN  #FIXED 'string'
                           resetRelativeToScheme CDATA    #IMPLIED >
```

```
<!ELEMENT rollConvention (#PCDATA)>

<!ATTLIST rollConvention  type                NMTOKEN  #FIXED 'string'
                          rollConventionScheme CDATA    #IMPLIED >
<!ELEMENT roundingDirection (#PCDATA)>

<!ATTLIST roundingDirection  type                   NMTOKEN  #FIXED 'string'
                             roundingDirectionScheme CDATA    #IMPLIED >
<!ELEMENT sellerPartyReference EMPTY>

<!ATTLIST sellerPartyReference  href CDATA  #REQUIRED >
<!ELEMENT spread (#PCDATA)>

<!ATTLIST spread  type NMTOKEN  #FIXED 'decimal' >
<!ELEMENT spreadSchedule (%FpML_Schedule;)>

<!ATTLIST spreadSchedule  type NMTOKEN  #FIXED 'Schedule' >
<!ELEMENT step (%FpML_Step;)>

<!ATTLIST step  type NMTOKEN  #FIXED 'Step'
                id   ID       #IMPLIED >
<!ELEMENT stepDate (#PCDATA)>

<!ATTLIST stepDate  type NMTOKEN  #FIXED 'date' >
<!ELEMENT stepFrequency (%FpML_Interval;)>

<!ATTLIST stepFrequency  type NMTOKEN  #FIXED 'Interval' >
<!ELEMENT stepRelativeTo (#PCDATA)>

<!ATTLIST stepRelativeTo  type                NMTOKEN  #FIXED 'string'
                          stepRelativeToScheme CDATA    #IMPLIED >
<!ELEMENT stepValue (#PCDATA)>

<!ATTLIST stepValue  type NMTOKEN  #FIXED 'decimal' >
<!ELEMENT stubAmount (%FpML_Money;)>

<!ATTLIST stubAmount  type NMTOKEN  #FIXED 'Money' >
<!ELEMENT stubCalculationPeriodAmount (%FpML_StubCalculationPeriodAmount;)>

<!ATTLIST stubCalculationPeriodAmount  type NMTOKEN  #FIXED 'StubCalculationPeriodAmount'
>
<!ELEMENT stubRate (#PCDATA)>

<!ATTLIST stubRate  type NMTOKEN  #FIXED 'decimal' >
<!ELEMENT swap (%FpML_Swap;)>

<!ATTLIST swap  type NMTOKEN  #FIXED 'Swap'
                id   ID       #IMPLIED >
<!ELEMENT swapStream (%FpML_InterestRateStream;)>

<!ATTLIST swapStream  type NMTOKEN  #FIXED 'InterestRateStream'
                      id   ID       #IMPLIED >
<!ELEMENT terminationDate (%FpML_AdjustableDate;)>

<!ATTLIST terminationDate  type NMTOKEN  #FIXED 'AdjustableDate' >
<!ELEMENT trade (%FpML_Trade;)>

<!ATTLIST trade  type NMTOKEN  #FIXED 'Trade'
                 id   ID       #IMPLIED >
<!ELEMENT tradeDate (#PCDATA)>

<!ATTLIST tradeDate  type NMTOKEN  #FIXED 'date' >
<!ELEMENT tradeHeader (%FpML_TradeHeader;)>

<!ATTLIST tradeHeader  type NMTOKEN  #FIXED 'TradeHeader' >
<!ELEMENT tradeId (#PCDATA)>

<!ATTLIST tradeId  type          NMTOKEN  #FIXED 'string'
                   id            ID       #IMPLIED
                   tradeIdScheme CDATA    #IMPLIED >
<!ELEMENT treatedRate (#PCDATA)>
```

```
<!ATTLIST treatedRate  type NMTOKEN  #FIXED 'decimal' >
<!ELEMENT unadjustedDate (#PCDATA)>

<!ATTLIST unadjustedDate  type NMTOKEN  #FIXED 'date'
                          id   ID       #IMPLIED >
<!ELEMENT weeklyRollConvention (#PCDATA)>

<!ATTLIST weeklyRollConvention  type                        NMTOKEN  #FIXED 'string'
                                weeklyRollConventionScheme CDATA    #IMPLIED >
<!-- new version 2.0 entities below -->
<!ENTITY % FpML_AdjustableDates "unadjustedDate+ , dateAdjustments">

<!ENTITY % FpML_AdjustableOrRelativeDate "adjustableDate | relativeDate">

<!ENTITY % FpML_AdjustableOrRelativeDates "adjustableDates | relativeDates">

<!ENTITY % FpML_AmericanExercise "commencementDate , expirationDate ,
relevantUnderlyingDate , earliestExerciseTime , latestExerciseTime? , expirationTime ,
multipleExercise? , exerciseFeeSchedule?">

<!ENTITY % FpML_AutomaticExercise "thresholdRate">

<!ENTITY % FpML_BermudanExercise "bermudanExerciseDates , relevantUnderlyingDate ,
earliestExerciseTime , latestExerciseTime? , expirationTime , multipleExercise? ,
exerciseFeeSchedule?">

<!ENTITY % FpML_BusinessCenterTime "hourMinuteTime , businessCenter">

<!ENTITY % FpML_CalculationAgent "calculationAgentPartyReference+ |
calculationAgentParty">

<!ENTITY % FpML_CancelableProvision "buyerPartyReference , sellerPartyReference ,
(europeanExercise | bermudanExercise | americanExercise) , exerciseNoticePartyReference ,
exerciseNoticeBusinessCenter , followUpConfirmation , cancelableProvisionAdjustedDates?">

<!ENTITY % FpML_CancelableProvisionAdjustedDates "cancellationEvent+">

<!ENTITY % FpML_CancellationEvent "adjustedExerciseDate , adjustedEarlyTerminationDate">

<!ENTITY % FpML_CapFloor "%FpML_Product; , capFloorStream , additionalPayment*">

<!ENTITY % FpML_CashPriceMethod "cashSettlementReferenceBanks? , cashSettlementCurrency ,
quotationRateType">

<!ENTITY % FpML_CashSettlement "cashSettlementValuationTime , cashSettlementValuationDate
, cashSettlementPaymentDate? , (cashPriceMethod | cashPriceAlternateMethod |
parYieldCurveAdjustedMethod | zeroCouponYieldAdjustedMethod |
parYieldCurveUnadjustedMethod)">

<!ENTITY % FpML_CashSettlementPaymentDate "adjustableDates | relativeDate |
businessDateRange">

<!ENTITY % FpML_CashSettlementReferenceBanks "referenceBank+">

<!ENTITY % FpML_DateRange "unadjustedFirstDate , unadjustedLastDate">

<!ENTITY % FpML_BusinessDateRange "(%FpML_DateRange; , businessDayConvention ,
(businessCentersReference | businessCenters)?)">

<!ENTITY % FpML_EarlyTerminationEvent "adjustedExerciseDate ,
adjustedEarlyTerminationDate , adjustedCashSettlementValuationDate ,
adjustedCashSettlementPaymentDate , adjustedExerciseFeePaymentDate?">

<!ENTITY % FpML_EarlyTerminationProvision "mandatoryEarlyTermination |
optionalEarlyTermination">

<!ENTITY % FpML_EuropeanExercise "expirationDate , relevantUnderlyingDate? ,
earliestExerciseTime , expirationTime , partialExercise? , exerciseFee?">
```

```
<!ENTITY % FpML_ExerciseEvent "adjustedExerciseDate , adjustedRelevantSwapEffectiveDate ,
adjustedCashSettlementValuationDate? , adjustedCashSettlementPaymentDate? ,
adjustedExerciseFeePaymentDate?">

<!ENTITY % FpML_ExerciseFee "notionalReference , (feeAmount | feeRate) , feePaymentDate">

<!ENTITY % FpML_ExerciseFeeSchedule "notionalReference , (feeAmountSchedule |
feeRateSchedule) , feePaymentDate">

<!ENTITY % FpML_ExerciseProcedure "(manualExercise | automaticExercise) ,
followUpConfirmation">

<!ENTITY % FpML_ExtendibleProvision "buyerPartyReference , sellerPartyReference ,
(europeanExercise | bermudanExercise | americanExercise) , exerciseNoticePartyReference ,
exerciseNoticeBusinessCenter , followUpConfirmation , extendibleProvisionAdjustedDates?">

<!ENTITY % FpML_ExtendibleProvisionAdjustedDates "extensionEvent+">

<!ENTITY % FpML_ExtensionEvent "adjustedExerciseDate , adjustedExtendedTerminationDate">

<!ENTITY % FpML_FxLinkedNotionalAmount "adjustedFxSpotFixingDate , observedFxSpotRate? ,
notionalAmount?">

<!ENTITY % FpML_FxLinkedNotionalSchedule "constantNotionalScheduleReference ,
varyingNotionalCurrency , varyingNotionalFixingDates , fxSpotRateSource ,
varyingNotionalInterimExchangePaymentDates">

<!ENTITY % FpML_FxSpotRateSource "informationSource , fixingTime">

<!ENTITY % FpML_InformationSource "rateSource , rateSourcePage? ,
rateSourcePageHeading?">

<!ENTITY % FpML_MandatoryEarlyTermination "mandatoryEarlyTerminationDate ,
calculationAgentPartyReference+ , cashSettlement ,
mandatoryEarlyTerminationAdjustedDates?">

<!ENTITY % FpML_MandatoryEarlyTerminationAdjustedDates "adjustedEarlyTerminationDate ,
adjustedCashSettlementValuationDate , adjustedCashSettlementPaymentDate">

<!ENTITY % FpML_ManualExercise "exerciseNoticePartyReference ,
exerciseNoticeBusinessCenter , fallbackExercise?">

<!ENTITY % FpML_OptionalEarlyTermination "singlePartyOption? , (europeanExercise |
bermudanExercise | americanExercise) , exerciseNoticePartyReference ,
exerciseNoticeBusinessCenter , followUpConfirmation , calculationAgent , cashSettlement ,
optionalEarlyTerminationAdjustedDates?">

<!ENTITY % FpML_OptionalEarlyTerminationAdjustedDates "earlyTerminationEvent+">

<!ENTITY % FpML_PartialExercise "notionalReference+ , integralMultipleAmount? ,
minimumNotionalAmount">

<!ENTITY % FpML_MultipleExercise "%FpML_PartialExercise; , maximumNotionalAmount?">

<!ENTITY % FpML_ReferenceBank "referenceBankId , referenceBankName?">

<!ENTITY % FpML_RelativeDates "(%FpML_RelativeDateOffset; , scheduleBounds?)">

<!ENTITY % FpML_SettlementRateSource "informationSource | cashSettlementReferenceBanks">

<!ENTITY % FpML_SinglePartyOption "buyerPartyReference , sellerPartyReference">

<!ENTITY % FpML_Swaption "%FpML_Product; , buyerPartyReference , sellerPartyReference ,
premium* , (europeanExercise | bermudanExercise | americanExercise) , exerciseProcedure ,
calculationAgentPartyReference+ , cashSettlement? , swaptionStraddle ,
swaptionAdjustedDates? , swap">

<!ENTITY % FpML_BulletPayment "%FpML_Product; , %FpML_Payment;">

<!ENTITY % FpML_SwaptionAdjustedDates "exerciseEvent+">
```

```
<!ENTITY % FpML_YieldCurveMethod "settlementRateSource? , quotationRateType">

<!ELEMENT adjustableDate (%FpML_AdjustableDate;)>

<!ATTLIST adjustableDate  type NMTOKEN  #FIXED 'AdjustableDate' >
<!ELEMENT adjustableDates (%FpML_AdjustableDates;)>

<!ATTLIST adjustableDates  type NMTOKEN  #FIXED 'AdjustableDates' >
<!ELEMENT adjustedCashSettlementPaymentDate (#PCDATA)>

<!ATTLIST adjustedCashSettlementPaymentDate  type NMTOKEN  #FIXED 'date' >
<!ELEMENT adjustedCashSettlementValuationDate (#PCDATA)>

<!ATTLIST adjustedCashSettlementValuationDate  type NMTOKEN  #FIXED 'date' >
<!ELEMENT adjustedEarlyTerminationDate (#PCDATA)>

<!ATTLIST adjustedEarlyTerminationDate  type NMTOKEN  #FIXED 'date' >
<!ELEMENT adjustedExerciseDate (#PCDATA)>

<!ATTLIST adjustedExerciseDate  type NMTOKEN  #FIXED 'date' >
<!ELEMENT adjustedExerciseFeePaymentDate (#PCDATA)>

<!ATTLIST adjustedExerciseFeePaymentDate  type NMTOKEN  #FIXED 'date' >
<!ELEMENT adjustedExtendedTerminationDate (#PCDATA)>

<!ATTLIST adjustedExtendedTerminationDate  type NMTOKEN  #FIXED 'date' >
<!ELEMENT adjustedFxSpotFixingDate (#PCDATA)>

<!ATTLIST adjustedFxSpotFixingDate  type NMTOKEN  #FIXED 'date' >
<!ELEMENT adjustedRelevantSwapEffectiveDate (#PCDATA)>

<!ATTLIST adjustedRelevantSwapEffectiveDate  type NMTOKEN  #FIXED 'date' >
<!ELEMENT americanExercise (%FpML_AmericanExercise;)>

<!ATTLIST americanExercise  type NMTOKEN  #FIXED 'AmericanExercise'
                            id   ID       #IMPLIED >
<!ELEMENT automaticExercise (%FpML_AutomaticExercise;)>

<!ATTLIST automaticExercise  type NMTOKEN  #FIXED 'AutomaticExercise' >
<!ELEMENT bermudanExercise (%FpML_BermudanExercise;)>

<!ATTLIST bermudanExercise  type NMTOKEN  #FIXED 'BermudanExercise'
                            id   ID       #IMPLIED >
<!ELEMENT bermudanExerciseDates (%FpML_AdjustableOrRelativeDates;)>

<!ATTLIST bermudanExerciseDates  type NMTOKEN  #FIXED 'AdjustableOrRelativeDates' >
<!ELEMENT bulletPayment (%FpML_BulletPayment;)>

<!ATTLIST bulletPayment  type NMTOKEN  #FIXED 'BulletPayment'
                         id   ID       #IMPLIED >
<!ELEMENT businessDateRange (%FpML_BusinessDateRange;)>

<!ATTLIST businessDateRange  type NMTOKEN  #FIXED 'BusinessDateRange' >
<!ELEMENT buyer (#PCDATA)>

<!ATTLIST buyer  type               NMTOKEN  #FIXED 'string'
                 id                   ID       #IMPLIED
                 payerReceiverScheme CDATA    #IMPLIED >
<!ELEMENT calculationAgent (%FpML_CalculationAgent;)>

<!ATTLIST calculationAgent  type NMTOKEN  #FIXED 'CalculationAgent' >
<!ELEMENT calculationAgentParty (#PCDATA)>

<!ATTLIST calculationAgentParty  type                        NMTOKEN  #FIXED 'string'
                                 calculationAgentPartyScheme CDATA    #IMPLIED >
<!ELEMENT cancelableProvision (%FpML_CancelableProvision;)>

<!ATTLIST cancelableProvision  type NMTOKEN  #FIXED 'CancelableProvision' >
<!ELEMENT cancelableProvisionAdjustedDates (%FpML_CancelableProvisionAdjustedDates;)>
```

```
<!ATTLIST cancelableProvisionAdjustedDates  type NMTOKEN  #FIXED
'CancelableProvisionAdjustedDates' >
<!ELEMENT cancellationEvent (%FpML_CancellationEvent;)>

<!ATTLIST cancellationEvent  type NMTOKEN  #FIXED 'CancellationEvent'
                               id   ID       #IMPLIED >
<!ELEMENT capFloor (%FpML_CapFloor;)>

<!ATTLIST capFloor  type NMTOKEN  #FIXED 'CapFloor'
                      id   ID       #IMPLIED >
<!ELEMENT capFloorStream (%FpML_InterestRateStream;)>

<!ATTLIST capFloorStream  type NMTOKEN  #FIXED 'InterestRateStream' >
<!ELEMENT cashPriceAlternateMethod (%FpML_CashPriceMethod;)>

<!ATTLIST cashPriceAlternateMethod  type NMTOKEN  #FIXED 'CashPriceMethod' >
<!ELEMENT cashPriceMethod (%FpML_CashPriceMethod;)>

<!ATTLIST cashPriceMethod  type NMTOKEN  #FIXED 'CashPriceMethod' >
<!ELEMENT cashSettlement (%FpML_CashSettlement;)>

<!ATTLIST cashSettlement  type NMTOKEN  #FIXED 'CashSettlement'
                            id   ID       #IMPLIED >
<!ELEMENT cashSettlementCurrency (#PCDATA)>

<!ATTLIST cashSettlementCurrency  type           NMTOKEN  #FIXED 'string'
                                    currencyScheme CDATA    #IMPLIED >
<!ELEMENT cashSettlementPaymentDate (%FpML_CashSettlementPaymentDate;)>

<!ATTLIST cashSettlementPaymentDate  type NMTOKEN  #FIXED 'CashSettlementPaymentDate' >
<!ELEMENT cashSettlementReferenceBanks (%FpML_CashSettlementReferenceBanks;)>

<!ATTLIST cashSettlementReferenceBanks  type NMTOKEN  #FIXED
'CashSettlementReferenceBanks'
                                          id   ID       #IMPLIED >
<!ELEMENT cashSettlementValuationDate (%FpML_RelativeDateOffset;)>

<!ATTLIST cashSettlementValuationDate  type NMTOKEN  #FIXED 'RelativeDateOffset' >
<!ELEMENT cashSettlementValuationTime (%FpML_BusinessCenterTime;)>

<!ATTLIST cashSettlementValuationTime  type NMTOKEN  #FIXED 'BusinessCenterTime' >
<!ELEMENT commencementDate (%FpML_AdjustableOrRelativeDate;)>

<!ATTLIST commencementDate  type NMTOKEN  #FIXED 'AdjustableOrRelativeDate' >
<!ELEMENT constantNotionalScheduleReference EMPTY>

<!ATTLIST constantNotionalScheduleReference  href CDATA  #REQUIRED >
<!ELEMENT earliestExerciseTime (%FpML_BusinessCenterTime;)>

<!ATTLIST earliestExerciseTime  type NMTOKEN  #FIXED 'BusinessCenterTime' >
<!ELEMENT earlyTerminationEvent (%FpML_EarlyTerminationEvent;)>

<!ATTLIST earlyTerminationEvent  type NMTOKEN  #FIXED 'EarlyTerminationEvent'
                                   id   ID       #IMPLIED >
<!ELEMENT earlyTerminationProvision (%FpML_EarlyTerminationProvision;)>

<!ATTLIST earlyTerminationProvision  type NMTOKEN  #FIXED 'EarlyTerminationProvision'
                                       id   ID       #IMPLIED >
<!ELEMENT europeanExercise (%FpML_EuropeanExercise;)>

<!ATTLIST europeanExercise  type NMTOKEN  #FIXED 'EuropeanExercise'
                              id   ID       #IMPLIED >
<!ELEMENT exerciseEvent (%FpML_ExerciseEvent;)>

<!ATTLIST exerciseEvent  type NMTOKEN  #FIXED 'ExerciseEvent'
                           id   ID       #IMPLIED >
<!ELEMENT exerciseFee (%FpML_ExerciseFee;)>

<!ATTLIST exerciseFee  type NMTOKEN  #FIXED 'ExerciseFee' >
<!ELEMENT exerciseFeeSchedule (%FpML_ExerciseFeeSchedule;)>
```

```
<!ATTLIST exerciseFeeSchedule  type NMTOKEN  #FIXED 'ExerciseFeeSchedule' >
<!ELEMENT exerciseNoticeBusinessCenter (#PCDATA)>

<!ATTLIST exerciseNoticeBusinessCenter  type                NMTOKEN  #FIXED 'string'
                                        id                    ID       #IMPLIED
                                        businessCenterScheme CDATA    #IMPLIED >
<!ELEMENT exerciseNoticePartyReference EMPTY>

<!ATTLIST exerciseNoticePartyReference  href CDATA  #REQUIRED >
<!ELEMENT exerciseProcedure (%FpML_ExerciseProcedure;)>

<!ATTLIST exerciseProcedure  type NMTOKEN  #FIXED 'ExerciseProcedure' >
<!ELEMENT expirationDate (%FpML_AdjustableOrRelativeDate;)>

<!ATTLIST expirationDate  type NMTOKEN  #FIXED 'AdjustableOrRelativeDate' >
<!ELEMENT expirationTime (%FpML_BusinessCenterTime;)>

<!ATTLIST expirationTime  type NMTOKEN  #FIXED 'BusinessCenterTime' >
<!ELEMENT extendibleProvision (%FpML_ExtendibleProvision;)>

<!ATTLIST extendibleProvision  type NMTOKEN  #FIXED 'ExtendibleProvision' >
<!ELEMENT extendibleProvisionAdjustedDates (%FpML_ExtendibleProvisionAdjustedDates;)>

<!ATTLIST extendibleProvisionAdjustedDates  type NMTOKEN  #FIXED
'ExtendibleProvisionAdjustedDates' >
<!ELEMENT extensionEvent (%FpML_ExtensionEvent;)>

<!ATTLIST extensionEvent  type NMTOKEN  #FIXED 'ExtensionEvent'
                          id   ID       #IMPLIED >
<!ELEMENT fallbackExercise (#PCDATA)>

<!ATTLIST fallbackExercise  type NMTOKEN  #FIXED 'boolean' >
<!ELEMENT feeAmount (#PCDATA)>

<!ATTLIST feeAmount  type NMTOKEN  #FIXED 'decimal' >
<!ELEMENT feeAmountSchedule (%FpML_Schedule;)>

<!ATTLIST feeAmountSchedule  type NMTOKEN  #FIXED 'AmountSchedule' >
<!ELEMENT feePaymentDate (%FpML_RelativeDateOffset;)>

<!ATTLIST feePaymentDate  type NMTOKEN  #FIXED 'RelativeDateOffset'
                          base NMTOKEN  #FIXED 'Offset' >
<!ELEMENT feeRate (#PCDATA)>

<!ATTLIST feeRate  type NMTOKEN  #FIXED 'decimal' >
<!ELEMENT feeRateSchedule (%FpML_Schedule;)>

<!ATTLIST feeRateSchedule  type NMTOKEN  #FIXED 'Schedule' >
<!ELEMENT fixingTime (%FpML_BusinessCenterTime;)>

<!ATTLIST fixingTime  type NMTOKEN  #FIXED 'BusinessCenterTime' >
<!ELEMENT floatingRateMultiplier (#PCDATA)>

<!ATTLIST floatingRateMultiplier  type NMTOKEN  #FIXED 'decimal' >
<!ELEMENT floatingRateMultiplierSchedule (%FpML_Schedule;)>

<!ATTLIST floatingRateMultiplierSchedule  type NMTOKEN  #FIXED 'Schedule' >
<!ELEMENT followUpConfirmation (#PCDATA)>

<!ATTLIST followUpConfirmation  type NMTOKEN  #FIXED 'boolean' >
<!ELEMENT fxLinkedNotionalAmount (%FpML_FxLinkedNotionalAmount;)>

<!ATTLIST fxLinkedNotionalAmount  type NMTOKEN  #FIXED 'FxLinkedNotionalAmount' >
<!ELEMENT fxLinkedNotionalSchedule (%FpML_FxLinkedNotionalSchedule;)>

<!ATTLIST fxLinkedNotionalSchedule  type NMTOKEN  #FIXED 'FxLinkedNotionalSchedule' >
<!ELEMENT fxSpotRateSource (%FpML_FxSpotRateSource;)>

<!ATTLIST fxSpotRateSource  type NMTOKEN  #FIXED 'FxSpotRateSource' >
<!ELEMENT hourMinuteTime (#PCDATA)>
```

```
<!ATTLIST hourMinuteTime  type NMTOKEN  #FIXED 'time' >
<!ELEMENT informationSource (%FpML_InformationSource;)>

<!ATTLIST informationSource  type NMTOKEN  #FIXED 'InformationSource' >
<!ELEMENT integralMultipleAmount (#PCDATA)>

<!ATTLIST integralMultipleAmount  type NMTOKEN  #FIXED 'decimal' >
<!ELEMENT latestExerciseTime (%FpML_BusinessCenterTime;)>

<!ATTLIST latestExerciseTime  type NMTOKEN  #FIXED 'BusinessCenterTime' >
<!ELEMENT mandatoryEarlyTermination (%FpML_MandatoryEarlyTermination;)>

<!ATTLIST mandatoryEarlyTermination  type NMTOKEN  #FIXED 'MandatoryEarlyTermination'
                                      id   ID       #IMPLIED >
<!ELEMENT mandatoryEarlyTerminationAdjustedDates
(%FpML_MandatoryEarlyTerminationAdjustedDates;)>

<!ATTLIST mandatoryEarlyTerminationAdjustedDates  type NMTOKEN  #FIXED
'MandatoryEarlyTerminationAdjustedDates' >
<!ELEMENT mandatoryEarlyTerminationDate (%FpML_AdjustableDate;)>

<!ATTLIST mandatoryEarlyTerminationDate  type NMTOKEN  #FIXED 'AdjustableDate' >
<!ELEMENT manualExercise (%FpML_ManualExercise;)>

<!ATTLIST manualExercise  type NMTOKEN  #FIXED 'ManualExercise' >
<!ELEMENT maximumNotionalAmount (#PCDATA)>

<!ATTLIST maximumNotionalAmount  type NMTOKEN  #FIXED 'decimal' >
<!ELEMENT minimumNotionalAmount (#PCDATA)>

<!ATTLIST minimumNotionalAmount  type NMTOKEN  #FIXED 'decimal' >
<!ELEMENT multipleExercise (%FpML_MultipleExercise;)>

<!ATTLIST multipleExercise  type NMTOKEN  #FIXED 'MultipleExercise' >
<!ELEMENT notionalReference EMPTY>

<!ATTLIST notionalReference  href CDATA  #REQUIRED >
<!ELEMENT observedFxSpotRate (#PCDATA)>

<!ATTLIST observedFxSpotRate  type NMTOKEN  #FIXED 'decimal' >
<!ELEMENT optionalEarlyTermination (%FpML_OptionalEarlyTermination;)>

<!ATTLIST optionalEarlyTermination  type NMTOKEN  #FIXED 'OptionalEarlyTermination' >
<!ELEMENT optionalEarlyTerminationAdjustedDates
(%FpML_OptionalEarlyTerminationAdjustedDates;)>

<!ATTLIST optionalEarlyTerminationAdjustedDates  type NMTOKEN  #FIXED
'OptionalEarlyTerminationAdjustedDates' >
<!ELEMENT partialExercise (%FpML_PartialExercise;)>

<!ATTLIST partialExercise  type NMTOKEN  #FIXED 'PartialExercise' >
<!ELEMENT parYieldCurveAdjustedMethod (%FpML_YieldCurveMethod;)>

<!ATTLIST parYieldCurveAdjustedMethod  type NMTOKEN  #FIXED 'YieldCurveMethod' >
<!ELEMENT parYieldCurveUnadjustedMethod (%FpML_YieldCurveMethod;)>

<!ATTLIST parYieldCurveUnadjustedMethod  type NMTOKEN  #FIXED 'YieldCurveMethod' >
<!ELEMENT premium (%FpML_Payment;)>

<!ATTLIST premium  type NMTOKEN  #FIXED 'Payment'
                     id   ID       #IMPLIED >
<!ELEMENT productType (#PCDATA)>

<!ATTLIST productType  type              NMTOKEN  #FIXED 'string'
                         productTypeScheme CDATA    #IMPLIED >
<!ELEMENT quotationRateType (#PCDATA)>

<!ATTLIST quotationRateType  type                    NMTOKEN  #FIXED 'string'
                               quotationRateTypeScheme CDATA    #IMPLIED >
<!ELEMENT rateSource (#PCDATA)>
```

```
<!ATTLIST rateSource  type                      NMTOKEN  #FIXED 'string'
                        informationProviderScheme CDATA    #IMPLIED >
<!ELEMENT rateSourcePage (#PCDATA)>

<!ATTLIST rateSourcePage  type                NMTOKEN  #FIXED 'string'
                          rateSourcePageScheme CDATA    #IMPLIED >
<!ELEMENT rateSourcePageHeading (#PCDATA)>

<!ATTLIST rateSourcePageHeading  type NMTOKEN  #FIXED 'string' >
<!ELEMENT referenceBank (%FpML_ReferenceBank;)>

<!ATTLIST referenceBank  type NMTOKEN  #FIXED 'ReferenceBank'
                         id   ID       #IMPLIED >
<!ELEMENT referenceBankId (#PCDATA)>

<!ATTLIST referenceBankId  type                  NMTOKEN  #FIXED 'string'
                           referenceBankIdScheme CDATA    #IMPLIED >
<!ELEMENT referenceBankName (#PCDATA)>

<!ATTLIST referenceBankName  type NMTOKEN  #FIXED 'string' >
<!ELEMENT relativeDate (%FpML_RelativeDateOffset;)>

<!ATTLIST relativeDate  type NMTOKEN  #FIXED 'RelativeDateOffset'
                        base NMTOKEN  #FIXED 'Offset' >
<!ELEMENT relativeDates (%FpML_RelativeDates;)>

<!ATTLIST relativeDates  type NMTOKEN  #FIXED 'RelativeDates'
                         base NMTOKEN  #FIXED 'RelativeDate' >
<!ELEMENT relevantUnderlyingDate (%FpML_AdjustableOrRelativeDates;)>

<!ATTLIST relevantUnderlyingDate  type NMTOKEN  #FIXED 'AdjustableOrRelativeDates' >
<!ELEMENT scheduleBounds (%FpML_DateRange;)>

<!ATTLIST scheduleBounds  type NMTOKEN  #FIXED 'DateRange' >
<!ELEMENT seller (#PCDATA)>

<!ATTLIST seller  type               NMTOKEN  #FIXED 'string'
                  id                   ID       #IMPLIED
                  payerReceiverScheme CDATA    #IMPLIED >
<!ELEMENT settlementRateSource (%FpML_SettlementRateSource;)>

<!ATTLIST settlementRateSource  type NMTOKEN  #FIXED 'SettlementRateSource' >
<!ELEMENT singlePartyOption (%FpML_SinglePartyOption;)>

<!ATTLIST singlePartyOption  type NMTOKEN  #FIXED 'SinglePartyOption' >
<!ELEMENT strategy (%FpML_Strategy;)>

<!ATTLIST strategy  type NMTOKEN  #FIXED 'Strategy'
                    id   ID       #IMPLIED >
<!ELEMENT strikeRate (#PCDATA)>

<!ATTLIST strikeRate  type NMTOKEN  #FIXED 'decimal' >
<!ELEMENT swaption (%FpML_Swaption;)>

<!ATTLIST swaption  type NMTOKEN  #FIXED 'Swaption'
                    id   ID       #IMPLIED >
<!ELEMENT swaptionAdjustedDates (%FpML_SwaptionAdjustedDates;)>

<!ATTLIST swaptionAdjustedDates  type NMTOKEN  #FIXED 'SwaptionAdjustedDates' >
<!ELEMENT swaptionStraddle (#PCDATA)>

<!ATTLIST swaptionStraddle  type NMTOKEN  #FIXED 'boolean' >
<!ELEMENT thresholdRate (#PCDATA)>

<!ATTLIST thresholdRate  type NMTOKEN  #FIXED 'decimal' >
<!ELEMENT unadjustedFirstDate (#PCDATA)>

<!ATTLIST unadjustedFirstDate  type NMTOKEN  #FIXED 'date' >
<!ELEMENT unadjustedLastDate (#PCDATA)>

<!ATTLIST unadjustedLastDate  type NMTOKEN  #FIXED 'date' >
```

```
<!ELEMENT varyingNotionalCurrency (#PCDATA)>

<!ATTLIST varyingNotionalCurrency  type           NMTOKEN  #FIXED 'string'
                                   currencyScheme CDATA    #IMPLIED >
<!ELEMENT varyingNotionalFixingDates (%FpML_RelativeDateOffset;)>

<!ATTLIST varyingNotionalFixingDates  type NMTOKEN  #FIXED 'RelativeDateOffset'
                                      base NMTOKEN  #FIXED 'Offset' >
<!ELEMENT varyingNotionalInterimExchangePaymentDates (%FpML_RelativeDateOffset;)>

<!ATTLIST varyingNotionalInterimExchangePaymentDates  type NMTOKEN  #FIXED
'RelativeDateOffset'
                                                      base NMTOKEN  #FIXED 'Offset' >
<!ELEMENT zeroCouponYieldAdjustedMethod (%FpML_YieldCurveMethod;)>

<!ATTLIST zeroCouponYieldAdjustedMethod  type NMTOKEN  #FIXED 'YieldCurveMethod' >
```

7 DATA DICTIONARY

7.1 Element Definitions

Element/Description	Used By
additionalPayment ; entity type: FpML_Fee Additional payments between the principal parties involved in the swap.	FpML_Swap FpML_CapFloor
adjustableDate ; entity type: FpML_AdjustableDate A date that shall be subject to adjustment if it would otherwise fall on a day that is not a business day in the specified business centers, together with the convention for adjusting the date.	FpML_AdjustableOrRelativeDate
adjustableDates ; entity type: FpML_AdjustableDates A series of dates that shall be subject to adjustment if they would otherwise fall on a day that is not a business day in the specified business centers, together with the convention for adjusting the date.	FpML_AdjustableOrRelativeDates FpML_CashSettlementPaymentDate
adjustedCashSettlementPaymentDate ; built-in datatype: date The date on which the cash settlement amount is paid. This date should already be adjusted for any applicable business day convention.	FpML_EarlyTerminationEvent FpML_ExerciseEvent FpML_MandatoryEarlyTerminationAdjustedDates
adjustedCashSettlementValuationDate ; built-in datatype: date	FpML_EarlyTerminationEvent FpML_ExerciseEvent FpML_MandatoryEarlyTerminationAdj

The date by which the cash settlement amount must be agreed. This date should already be adjusted for any applicable business day convention.	ustedDates
adjustedEarlyTerminationDate ; built-in datatype: date The early termination date that is applicable if an early termination provision is exercised. This date should already be adjusted for any applicable business day convention.	FpML_CancellationEvent FpML_EarlyTerminationEvent FpML_MandatoryEarlyTerminationAdj ustedDates
adjustedEffectiveDate ; built-in datatype: date The start date of the calculation period. This date should already be adjusted for any applicable business day convention. This is also the date when the observed rate is applied, the reset date.	FpML_Fra
adjustedEndDate ; built-in datatype: date The calculation period end date, adjusted according to any relevant business day convention.	FpML_CalculationPeriod
adjustedExerciseDate ; built-in datatype: date The date on which option exercise takes place. This date should already be adjusted for any applicable business day convention.	FpML_CancellationEvent FpML_EarlyTerminationEvent FpML_ExerciseEvent FpML_ExtensionEvent
adjustedExerciseFeePaymentDate ; built-in datatype: date The date on which the exercise fee amount is paid. This date should already be adjusted for any	FpML_EarlyTerminationEvent FpML_ExerciseEvent

applicable business day convention.	
adjustedExtendedTerminationDate ; built-in datatype: `date` The termination date if an extendible provision is exercised. This date should already be adjusted for any applicable business day convention.	FpML_ExtensionEvent
adjustedFixingDate ; built-in datatype: `date` The adjusted fixing date, i.e. the actual date the rate is observed. This date should already be adjusted for any applicable business day convention.	FpML_RateObservation
adjustedFxSpotFixingDate ; built-in datatype: `date` The date on which the fx spot rate is observed. This date should already be adjusted for any applicable business day convention.	FpML_FxLinkedNotionalAmount
adjustedPaymentDate ; built-in datatype: `date` The adjusted payment date. This date should already be adjusted for any applicable business day convention. (FpML_Fee usage) This element is not intended for use in trade confirmation but may be specified to allow the fee structure to also serve as a cashflow type component (all dates in the FpML_Cashflows entity are adjusted payment dates).	FpML_Fee FpML_PaymentCalculationPeriod
adjustedPrincipalExchangeDate ;	FpML_PrincipalExchange

built-in datatype: `date` The principal exchange date. This date should already be adjusted for any applicable business day convention.	
`adjustedRelevantSwapEffectiveDate` ; built-in datatype: `date` The effective date of the underlying swap associated with a given exercise date. This date should already be adjusted for any applicable business day convention.	`FpML_ExerciseEvent`
`adjustedStartDate` ; built-in datatype: `date` The calculation period start date, adjusted according to any relevant business day convention.	`FpML_CalculationPeriod`
`adjustedTerminationDate` ; built-in datatype: `date` The end date of the calculation period. This date should already be adjusted for any applicable business day convention.	`FpML_Fra`

`americanExercise` ; entity type: FpML_AmericanExercise
 	
The parameters for defining the exercise period for an American style option together with any rules governing the notional amount of the underlying which can be exercised on any given exercise date and any associated exercise fees.
	`FpML_CancelableProvision`
	`FpML_ExtendibleProvision`
	`FpML_OptionalEarlyTermination`
	`FpML_Swaption`

`amount` ; built-in datatype: `decimal`	`FpML_Money`

The monetary quantity in currency units.	
automaticExercise ; entity type: FpML_AutomaticExercise If automatic exercise is specified then the notional amount of the underlying swap, not previously exercised under the swaption, will be automatically exercised at the expiration time on the expiration date if at such time the buyer is in-the-money, provided that the difference between the settlement rate and the fixed rate under the relevant underlying swap is not less than the specified thresholdRate. The term In-the-money is assumed to have the meaning defined in the 2000 ISDA Definitions, Section 17.4. In-the-money.	FpML_ExerciseProcedure
averagingMethod ; built-in datatype: `string` ; coding scheme: `averagingMethodScheme` If averaging is applicable, this element specifies whether a weighted or unweighted average method of calculation is to be used. The element must only be included when averaging applies.	FpML_FloatingRateCalculation
bermudanExercise ; entity type: FpML_BermudanExercise The parameters for defining the exercise period for a Bermudan style option together with any rules governing the notional amount of the underlying which can be exercised on any given exercise date and any associated exercise fees.	FpML_CancelableProvision FpML_ExtendibleProvision FpML_OptionalEarlyTermination FpML_Swaption
bermudanExerciseDates ; entity type: FpML_AdjustableOrRelativeDates The dates that define the bermudan option exercise dates and the expiration date. The last specified exercise date is assumed to be the	FpML_BermudanExercise

expiration date. The dates can either be specified as a series of explicit dates and associated adjustments or as a series of dates defined relative to another schedule of dates, for example, the calculation period start dates. Where a relative series of dates are defined the first and last possible exercise dates can be separately specified.	
`bulletPayment` ; entity type: FpML_BulletPayment A product to represent one or more known payments.	`FpML_ProductList`
`businessCenter` ; built-in datatype: `string` ; coding scheme: `businessCenterScheme` A code identifying a financial business center location. A list of business centers may be ordered in the document alphabetically based on business center code. An FpML document containing an unordered business center list is still regarded as a conformant document.	`FpML_BusinessCenters` `FpML_BusinessCenterTime`

`businessCenters` ; entity type: FpML_BusinessCenters
 	
A container for a set of financial business centers. This set of business centers is used to determine whether a day is a business day or not.
	`FpML_BusinessDayAdjustments`
	`FpML_RelativeDateOffset`
	`FpML_BusinessDateRange`

`businessCentersReference` ; empty element
 	
A pointer style reference to a set of financial business centers defined elsewhere in the document. This set of business centers is used to determine whether a particular day is a business day or not.
	`FpML_BusinessDayAdjustments`
	`FpML_RelativeDateOffset`
	`FpML_BusinessDateRange`

businessDateRange ; entity type: FpML_BusinessDateRange A range of contiguous business days.	FpML_CashSettlementPaymentDate
businessDayConvention ; built-in datatype: `string` ; coding scheme: `businessDayConventionScheme` The convention for adjusting a date if it would otherwise fall on a day that is not a business day. (FpML_BusinessDayAdjustments usage) If the business day convention value is NONE then neither the businessCentersReference or businessCenters element should be included (FpML_RelativeDateOffset usage) If the business day convention value is NONE then the businessCentersReference or businessCenters element should still be included if the dayType element contains a value of Business since the business centers defined are those used for determining good business days.	FpML_BusinessDayAdjustments FpML_RelativeDateOffset FpML_BusinessDateRange
buyer ; built-in datatype: `string` ; coding scheme: `payerReceiverScheme` The buyer of the option	FpML_StrikeRate FpML_StrikeRateSchedule
buyerPartyReference ; empty element A pointer style reference to a party identifier defined elsewhere in the document. The party referenced is the buyer of the instrument, also known as the fixed rate payer.	FpML_Fra FpML_CancelableProvision FpML_ExtendibleProvision FpML_SinglePartyOption FpML_Swaption
calculatedRate ; built-in datatype: `decimal`	FpML_FloatingRateDefinition

The final calculated rate for a calculation period after any required averaging of rates. A calculated rate of 5% would be represented as 0.05.	
calculation ; entity type: FpML_Calculation The parameters used in the calculation of fixed or floating rate calculation period amounts.	FpML_CalculationPeriodAmount
calculationAgent ; entity type: FpML_CalculationAgent The ISDA Calculation Agent responsible for performing duties associated with an optional early termination.	FpML_OptionalEarlyTermination
calculationAgentParty ; built-in datatype: string ; coding scheme: calculationAgentPartyScheme The ISDA Calculation Agent where the actual party responsible for performing the duties associated with an optional early termination provision will be determined at exercise. For example, the Calculation Agent may be defined as being the Non-exercising Party.	FpML_CalculationAgent
calculationAgentPartyReference ; empty element A pointer style reference to a party identifier defined elsewhere in the document. The party referenced is the ISDA Calculation Agent for the trade. If more than one party is referenced then the parties are assumed to be co-calculation agents, i.e. they have joint responsibility.	FpML_TradeHeader FpML_CalculationAgent FpML_MandatoryEarlyTermination FpML_Swaption
calculationPeriod ; entity type:	FpML_PaymentCalculationPeriod

FpML_CalculationPeriod The parameters used in the calculation of a fixed or floating rate calculation period amount. A list of calculation period elements may be ordered in the document by ascending adjusted start date. An FpML document which contains an unordered list of calculation periods is still regarded as a conformant document.	
calculationPeriodAmount ; entity type: FpML_CalculationPeriodAmount The calculation period amount parameters.	FpML_InterestRateStream
calculationPeriodDates ; entity type: FpML_CalculationPeriodDates The calculation periods dates schedule.	FpML_InterestRateStream
calculationPeriodDatesAdjustments ; entity type: FpML_BusinessDayAdjustments The business day convention to apply to each calculation period end date if it would otherwise fall on a day that is not a business day in the specified financial business centers.	FpML_CalculationPeriodDates
calculationPeriodDatesReference ; empty element A pointer style reference to the associated calculation period dates component defined elsewhere in the document.	FpML_NotionalStepRule FpML_PaymentDates FpML_ResetDates FpML_StubCalculationPeriodAmount
calculationPeriodFrequency ; entity type: FpML_CalculationPeriodFrequency The frequency at which calculation period end	FpML_CalculationPeriodDates

dates occur within the regular part of the calculation period schedule and their roll date convention.	
calculationPeriodNumberOfDays ; entity type: FpML_positiveInteger The number of days from the adjusted effective date to the adjusted termination date calculated in accordance with the applicable day count fraction.	FpML_Fra
cancelableProvision ; entity type: FpML_CancelableProvision A provision that allows the specification of an embedded option within a swap giving the buyer of the option the right to terminate the swap, in whole or in part, on the early termination date.	FpML_Swap
cancelableProvisionAdjustedDates ; entity type: FpML_CancelableProvisionAdjustedDates The adjusted dates associated with a cancelable provision. These dates have been adjusted for any applicable business day convention.	FpML_CancelableProvision
cancellationEvent ; entity type: FpML_CancellationEvent The adjusted dates for an individual cancellation date.	FpML_CancelableProvisionAdjustedDates
capFloor ; entity type: FpML_CapFloor A cap, floor or cap floor structures product definition.	FpML_Product

capFloorStream ; entity type: FpML_InterestRateStream A cap, floor or cap floor structure stream.	FpML_CapFloor
capRate ; entity type: FpML_StrikeRate The cap rate, if any, which applies to the floating rate for the calculation period. The cap rate (strike) is only required where the floating rate on a swap stream is capped at a certain strike level. The cap rate is assumed to be exclusive of any spread and is a per annum rate, expressed as a decimal. A cap rate of 5% would be represented as 0.05.	FpML_FloatingRateDefinition
capRateSchedule ; entity type: FpML_StrikeRateSchedule The cap rate or cap rate schedule, if any, which applies to the floating rate. The cap rate (strike) is only required where the floating rate on a swap stream is capped at a certain strike level. A cap rate schedule is expressed as explicit cap rates and dates and the step dates may be subject to adjustment in accordance with any adjustments specified in calculationPeriodDatesAdjustments. The cap rate is assumed to be exclusive of any spread and is a per annum rate, expressed as a decimal. A cap rate of 5% would be represented as 0.05.	FpML_FloatingRate
cashflows ; entity type: FpML_Cashflows The cashflows representation of the swap stream.	FpML_InterestRateStream
cashflowsMatchParameters ; built-in datatype: boolean A true/false flag to indicate whether the cashflows	FpML_Cashflows

match the parametric definition of the stream, i.e. whether the cashflows could be regenerated from the parameters without loss of information.	
cashPriceAlternateMethod ; entity type: FpML_CashPriceMethod An ISDA defined cash settlement method used for the determination of the applicable cash settlement amount. The method is defined in the 2000 ISDA Definitions, Section 17.3. Cash Settlement Methods, paragraph (b).	FpML_CashSettlement
cashPriceMethod ; entity type: FpML_CashPriceMethod An ISDA defined cash settlement method used for the determination of the applicable cash settlement amount. The method is defined in the 2000 ISDA Definitions, Section 17.3. Cash Settlement Methods, paragraph (a).	FpML_CashSettlement
cashSettlement ; entity type: FpML_CashSettlement If specified, this means that cash settlement is applicable to the transaction and defines the parameters associated with the cash settlement procedure. If not specified, then physical settlement is applicable.	FpML_MandatoryEarlyTermination FpML_OptionalEarlyTermination FpML_Swaption
cashSettlementCurrency ; built-in datatype: `string` ; coding scheme: `currencyScheme` The currency in which the cash settlement amount will be specified.	FpML_CashPriceMethod
cashSettlementPaymentDate ; entity type:	FpML_CashSettlement

FpML_CashSettlementPaymentDate The date on which the cash settlement amount will be paid, subject to adjustment in accordance with any applicable business day convention.This element would not be present for a mandatory early termination provision where the cash settlement payment date is the mandatory early termination date.	
cashSettlementReferenceBanks ; entity type: FpML_CashSettlementReferenceBanks A container for a set of reference institutions. These reference institutions may be called upon to provide rate quotations as part of the method to determine the applicable cash settlement amount.	FpML_CashPriceMethod FpML_SettlementRateSource
cashSettlementValuationDate ; entity type: FpML_RelativeDateOffset The date on which the cash settlement amount will be determined according to the cash settlement method if the parties have not otherwise been able to agree the cash settlement amount.	FpML_CashSettlement
cashSettlementValuationTime ; entity type: FpML_BusinessCenterTime The time on the cash settlement valuation date when the cash settlement amount will be determined according to the cash settlement method if the parties have not otherwise been able to agree the cash settlement amount.	FpML_CashSettlement
commencementDate ; entity type: FpML_AdjustableOrRelativeDate The first day of the exercise period for an American style option.	FpML_AmericanExercise

compoundingMethod ; built-in datatype: `string` ; coding scheme: `compoundingMethodScheme` If more than one calculation period contributes to a single payment amount this element specifies whether compounding is applicable, and if so, what compounding method is to be used. This element must only be included when more than one calculation period contributes to a single payment amount.	FpML_Calculation
constantNotionalScheduleReference ; empty element A pointer style reference to the associated constant notional schedule defined elsewhere in the document which contains the currency amounts which will be converted into the varying notional currency amounts using the spot currency exchange rate.	FpML_FxLinkedNotionalSchedule
currency ; built-in datatype: `string` ; coding scheme: `currencyScheme` The currency in which an amount is denominated.	FpML_Money FpML_AmountSchedule
dateAdjustments ; entity type: FpML_BusinessDayAdjustments The business day convention and financial business centers used for adjusting the date if it would otherwise fall on a day that is not a business day in the specified business centers.	FpML_AdjustableDate FpML_AdjustableDates
dateRelativeTo ; built-in datatype: `string` ; coding scheme: `dateRelativeToScheme` Specifies the anchor date. This element also	FpML_RelativeDateOffset

carries an href attribute. The href attribute value will be a pointer style reference to the element or component elsewhere in the document where the anchor date is defined.	
dayCountFraction ; built-in datatype: string ; coding scheme: dayCountFractionScheme The day count fraction.	FpML_Calculation FpML_Fra
dayType ; built-in datatype: string ; coding scheme: dayTypeScheme In the case of an offset specified as a number of days, this element defines whether consideration is given as to whether a day is a good business day or not. If a day type of business days is specified then non-business days are ignored when calculating the offset. The financial business centers to use for determination of business days are implied by the context in which this element is used. This element must only be included when the offset is specified as a number of days. If the offset is zero days then the dayType element should not be included.	FpML_Offset
discounting ; entity type: FpML_Discounting The parameters specifying any discounting conventions that may apply. This element must only be included if discounting applies.	FpML_Calculation
discountingType ; built-in datatype: string ; coding scheme: discountingTypeScheme The discounting method that is applicable.	FpML_Discounting

discountRate ; built-in datatype: decimal A discount rate, expressed as a decimal, to be used in the calculation of a discounted amount. A discount rate of 5% would be represented as 0.05.	FpML_Discounting
discountRateDayCountFraction ; built-in datatype: string ; coding scheme: dayCountFractionScheme A discount day count fraction to be used in the calculation of a discounted amount.	FpML_Discounting
earliestExerciseTime ; entity type: FpML_BusinessCenterTime The earliest time at which notice of exercise can be given by the buyer to the seller (or seller's agent) i) on the expiration date, in the case of a European style option, (ii) on each bermuda option exercise date and the expiration date, in the case of a Bermudan style option and (iii) all days that are exercise business days from and including the commencement date to, and including, the expiration date, in the case of an American style option.	FpML_AmericanExercise FpML_BermudanExercise FpML_EuropeanExercise
earlyTerminationEvent ; entity type: FpML_EarlyTerminationEvent The adjusted dates associated with an individual early termination date.	FpML_OptionalEarlyTerminationAdjustedDates
earlyTerminationProvision ; entity type: FpML_EarlyTerminationProvision Parameters specifying provisions relating to the optional and mandatory early termination of a swap transaction.	FpML_Swap

effectiveDate ; entity type: FpML_AdjustableDate The first day of the term of the trade. This day may be subject to adjustment in accordance with a business day convention.	FpML_CalculationPeriodDates
europeanExercise ; entity type: FpML_EuropeanExercise The parameters for defining the exercise period for a European style option together with any rules governing the notional amount of the underlying which can be exercised on any given exercise date and any associated exercise fees.	FpML_CancelableProvision FpML_ExtendibleProvision FpML_OptionalEarlyTermination FpML_Swaption
exerciseEvent ; entity type: FpML_ExerciseEvent The adjusted dates associated with an individual swaption exercise date.	FpML_SwaptionAdjustedDates
exerciseFee ; entity type: FpML_ExerciseFee A fee to be paid on exercise. This could be represented as an amount or a rate and notional reference on which to apply the rate.	FpML_EuropeanExercise
exerciseFeeSchedule ; entity type: FpML_ExerciseFeeSchedule The fees associated with an exercise date. The fees are conditional on the exercise occurring. The fees can be specified as actual currency amounts or as percentages of the notional amount being exercised.	FpML_AmericanExercise FpML_BermudanExercise
exerciseNoticeBusinessCenter ; built-	FpML_CancelableProvision

in datatype: `string` ; coding scheme: `businessCenterScheme`
 	
The business center location where notice of exercise should be given to the seller or seller's agent.
	`FpML_ExtendibleProvision`
	`FpML_ManualExercise`
	`FpML_OptionalEarlyTermination`

`exerciseNoticePartyReference` ; empty element
 	
A pointer style reference to a party identifier defined elsewhere in the document. The party referenced is the party to which notice of exercise should be given by the buyer.
	`FpML_CancelableProvision`
	`FpML_ExtendibleProvision`
	`FpML_ManualExercise`
	`FpML_OptionalEarlyTermination`

`exerciseProcedure` ; entity type: FpML_ExerciseProcedure A set of parameters defining procedures associated with the exercise.	`FpML_Swaption`
`expirationDate` ; entity type: FpML_AdjustableOrRelativeDate The last day within an exercise period for a Bermudan or American style option. For a European style option it is the only day within the exercise period.	`FpML_AmericanExercise` `FpML_EuropeanExercise`

`expirationTime` ; entity type: FpML_BusinessCenterTime
 	
The latest time for expiration on expirationDate.
	`FpML_AmericanExercise`
	`FpML_BermudanExercise`
	`FpML_EuropeanExercise`

`extendibleProvision` ; entity type: FpML_ExtendibleProvision A provision that allows the specification of an embedded option within a swap giving the buyer	`FpML_Swap`

of the option the right to extend the swap, in whole or in part, to the extended termination date.	
extendibleProvisionAdjustedDates ; entity type: FpML_ExtendibleProvisionAdjustedDates The adjusted dates associated with a extendible provision. These dates have been adjusted for any applicable business day convention.	FpML_ExtendibleProvision
extensionEvent ; entity type: FpML_ExtensionEvent The adjusted dates associated with a single extendible exercise date.	FpML_ExtendibleProvisionAdjustedDates
fallbackExercise ; built-in datatype: boolean If fallback exercise is specified then the notional amount of the underlying swap, not previously exercised under the swaption, will be automatically exercised at the expiration time on the expiration date if at such time the buyer is in-the-money, provided that the difference between the settlement rate and the fixed rate under the relevant underlying swap is not less than one tenth of a percentage point (0.10% or 0.001). The term In-the-money is assumed to have the meaning defined in the 2000 ISDA Definitions, Section 17.4. In-the-money.	FpML_ManualExercise
feeAmount ; entity type: decimal The amount of fee to be paid on exercise. The currency of this fee is the currency of the referenced notional	FpML_ExerciseFee

feeAmountSchedule ; entity type: FpML_AmountSchedule The exercise fee amount schedule. The fees are expressed as currency amounts. The currency of the fee is assumed to be that of the notional schedule referenced.	FpML_ExerciseFeeSchedule
feePaymentDate ; entity type: FpML_RelativeDateOffset The date on which exercise fees will be paid. It can be specified as a reference date or a relative date.	FpML_ExerciseFee FpML_ExerciseFeeSchedule
feeRate ; entity type: decimal A fee represented as a percentage of some referenced notional	FpML_ExerciseFee
feeRateSchedule ; entity type: FpML_Schedule The exercise fee rate schedule. The fees are expressed as percentage rates of the notional being exercised. The currency of the fee is assumed to be that of the notional schedule referenced.	FpML_ExerciseFeeSchedule
finalExchange ; built-in datatype: boolean A true/false flag to indicate whether there is a final exchange of principal on the termination date.	FpML_PrincipalExchanges
finalRateRounding ; entity type: FpML_Rounding The rounding convention to apply to the final rate used in determination of a calculation period	FpML_FloatingRateCalculation

amount.	
finalStub ; entity type: FpML_Stub Specifies how the final stub amount is calculated. A single floating rate tenor different to that used for the regular part of the calculation periods schedule may be specified, or two floating tenors may be specified. If two floating rate tenors are specified then Linear Interpolation (in accordance with the 2000 ISDA Definitions, Section 8.3. Interpolation) is assumed to apply. Alternatively, an actual known stub rate or stub amount may be specified.	FpML_StubCalculationPeriodAmount
firstNotionalStepDate ; built-in datatype: date The unadjusted calculation period start date of the first change in notional. This day may be subject to adjustment in accordance with any adjustments specified in calculationPeriodDatesAdjustments.	FpML_NotionalStepRule
firstPaymentDate ; built-in datatype: date The first unadjusted payment date. This day may be subject to adjustment in accordance with any business day convention specified in paymentDatesAdjustments. This element must only be included if there is an initial stub. This date will normally correspond to an unadjusted calculation period start or end date. This is true even if early or delayed payment is specified to be applicable since the actual first payment date will be the specified number of days before or after the applicable adjusted calculation period start or end date with the resulting payment date then being adjusted in accordance with any business day convention specified in paymentDatesAdjustments.	FpML_PaymentDates
	FpML_CalculationPeriodDates

firstPeriodStartDate ; entity type: FpML_AdjustableDate The start date of the first calculation period if the date falls before the effective date. It must only be specified if it is not equal to the effective date. This day may be subject to adjustment in accordance with a business day convention.	
firstRegularPeriodStartDate ; built-in datatype: `date` The start date of the regular part of the calculation period schedule. It must only be specified if there is an initial stub calculation period. This day may be subject to adjustment in accordance with any adjustments specified in calculationPeriodDatesAdjustments.	FpML_CalculationPeriodDates
fixedPaymentAmount ; built-in datatype: `decimal` A known fixed payment amount.	FpML_PaymentCalculationPeriod
fixedRate ; built-in datatype: `decimal` The calculation period fixed rate. A per annum rate, expressed as a decimal. A fixed rate of 5% would be represented as 0.05.	FpML_CalculationPeriod FpML_Fra
fixedRateSchedule ; entity type: FpML_Schedule The fixed rate or fixed rate schedule expressed as explicit fixed rates and dates. In the case of a schedule, the step dates may be subject to adjustment in accordance with any adjustments specified in calculationPeriodDatesAdjustments.	FpML_Calculation

fixingDateOffset ; entity type: FpML_RelativeDateOffset Specifies the fixing date relative to the reset date in terms of a business days offset and an associated set of financial business centers. Normally these offset calculation rules will be those specified in the ISDA definition for the relevant floating rate index (ISDA's Floating Rate Option). However, non-standard offset calculation rules may apply for a trade if mutually agreed by the principal parties to the transaction. The href attribute on the dateRelativeTo element should reference the id attribute on the adjustedEffectiveDate element.	FpML_Fra
fixingDates ; entity type: FpML_RelativeDateOffset Specifies the fixing date relative to each reset date in terms of a business days offset and an associated set of financial business centers. Normally these offset calculation rules will be those specified in the ISDA definition for the relevant floating rate index (ISDA's Floating Rate Option). However, non-standard offset calculation rules may apply for a trade if mutually agreed by the principal parties to the transaction. The href attribute on the dateRelativeTo element should reference the id attribute on the resetDates element.	FpML_ResetDates
fixingTime ; entity type: FpML_BusinessCenterTime The time at which the spot currency exchange rate will be observed. It is specified as a time in a specific business center, e.g. 11:00 am London time.	FpML_FxSpotRateSource
floatingRate ; entity type: FpML_FloatingRate The rates to be applied to the initial or final stub	FpML_Stub

may be the linear interpolation of two different rates. While the majority of the time, the rate indices will be the same as that specified in the stream and only the tenor itself will be different, it is possible to specify two different rates. For example, a 2 month stub period may use the linear interpolation of a 1 month and 3 month rate. The different rates would be specified in this component. Note that a maximum of two rates can be specified. If a stub period uses the same floating rate index, including tenor, as the regular calculation periods then this should not be specified again within this component, i.e. the stub calculation period amount component may not need to be specified even if there is an initial or final stub period. If a stub period uses a different floating rate index compared to the regular calculation periods then this should be specified within this component. If specified here, they are likely to have id attributes, allowing them to be referenced from within the cashflows component.	
`floatingRateCalculation` ; entity type: FpML_FloatingRateCalculation The floating rate calculation definitions.	`FpML_Calculation`
`floatingRateDefinition` ; entity type: FpML_FloatingRateDefinition The floating rate reset information for the calculation period.	`FpML_CalculationPeriod`
`floatingRateIndex` ; built-in datatype: `string` ; coding scheme: `floatingRateIndexScheme` The ISDA Floating Rate Option, i.e. the floating rate index.	`FpML_FloatingRate` `FpML_Fra`

floatingRateMultiplier ; built-in datatype: decimal An amount by which the floating rate is multiplied by.	FpML_FloatingRateDefinition
floatingRateMultiplierSchedule ; entity type: FpML_Schedule A schedule of values by which the floating rate will be multiplied by.	FpML_FloatingRate
floorRate ; entity type: FpML_StrikeRate The floor rate, if any, which applies to the floating rate for the calculation period. The floor rate (strike) is only required where the floating rate on a swap stream is floored at a certain strike level. The floor rate is assumed to be exclusive of any spread and is a per annum rate, expressed as a decimal. A floor rate of 5% would be represented as 0.05.	FpML_FloatingRateDefinition
floorRateSchedule ; entity type: FpML_StrikeRateSchedule The floor rate or floor rate schedule, if any, which applies to the floating rate. The floor rate (strike) is only required where the floating rate on a swap stream is floored at a certain strike level. A floor rate schedule is expressed as explicit floor rates and dates and the step dates may be subject to adjustment in accordance with any adjustments specified in calculationPeriodDatesAdjustments. The floor rate is assumed to be exclusive of any spread and is a per annum rate, expressed as a decimal. A floor rate of 5% would be represented as 0.05.	FpML_FloatingRate
followUpConfirmation ; built-in datatype: boolean	FpML_CancelableProvision FpML_ExerciseProcedure

A flag to indicate whether follow-up confirmation of exercise (written or electronic) is required following telephonic notice by the buyer to the seller or seller's agent.	FpML_ExtendibleProvision FpML_OptionalEarlyTermination
fra ; entity type: FpML_Fra A forward rate agreement product definition.	FpML_Product
fraDiscounting ; built-in datatype: boolean A true/false flag to indicate whether ISDA FRA Discounting applies. If false, then the calculation will be based on a par value and no discounting will apply.	FpML_Fra
fxLinkedNotionalAmount ; entity type: FpML_FxLinkedNotionalAmount The amount that a cashflow will accrue interest on. This is the calculated amount of the fx linked notional - ie the other currency notional amount multiplied by the appropriate fx spot rate.	FpML_CalculationPeriod
fxLinkedNotionalSchedule ; entity type: FpML_FxLinkedNotionalSchedule A notional amount schedule where each notional that applies to a calculation period is calculated with reference to a notional amount or notional amount schedule in a different currency by means of a spot currency exchange rate which is normally observed at the beginning of each period.	FpML_Calculation
fxSpotRateSource ; entity type: FpML_FxSpotRateSource	FpML_FxLinkedNotionalSchedule

The information source and time at which the spot currency exchange rate will be observed.	
hourMinuteTime ; built-in datatype: time A time specified in hh:mm:ss format where the second component must be '00', e.g. 11am would be represented as 11:00:00.	FpML_BusinessCenterTime
indexTenor ; entity type: FpML_Interval The ISDA Designated Maturity, i.e. the tenor of the floating rate.	FpML_FloatingRate FpML_Fra
informationSource ; entity type: FpML_InformationSource The information source where a published or displayed market rate will be obtained, e.g. Telerate Page 3750.	FpML_FxSpotRateSource FpML_SettlementRateSource
initialExchange ; built-in datatype: boolean A true/false flag to indicate whether there is an initial exchange of principal on the effective date.	FpML_PrincipalExchanges
initialRate ; built-in datatype: decimal The initial floating rate reset agreed between the principal parties involved in the trade. This is assumed to be the first required reset rate for the first regular calculation period. It should only be included when the rate is not equal to the rate published on the source implied by the floating rate index. An initial rate of 5% would be represented as 0.05.	FpML_FloatingRateCalculation

initialStub ; entity type: FpML_Stub Specifies how the initial stub amount is calculated. A single floating rate tenor different to that used for the regular part of the calculation periods schedule may be specified, or two floating tenors may be specified. If two floating rate tenors are specified then Linear Interpolation (in accordance with the 2000 ISDA Definitions, Section 8.3. Interpolation) is assumed to apply. Alternatively, an actual known stub rate or stub amount may be specified.	FpML_StubCalculationPeriodAmount
initialValue ; built-in datatype: decimal The initial rate or amount, as the case may be. An initial rate of 5% would be represented as 0.05.	FpML_Schedule
integralMultipleAmount ; built-in datatype: decimal A notional amount which restricts the amount of notional that can be exercised when partial exercise or multiple exercise is applicable. The integral multiple amount defines a lower limit of notional that can be exercised and also defines a unit multiple of notional that can be exercised, i.e. only integer multiples of this amount can be exercised.	FpML_PartialExercise
intermediateExchange ; built-in datatype: boolean A true/false flag to indicate whether there are intermediate or interim exchanges of principal during the term of the swap.	FpML_PrincipalExchanges
knownAmountSchedule ; entity type: FpML_AmountSchedule	FpML_CalculationPeriodAmount

The known calculation period amount or a known amount schedule expressed as explicit known amounts and dates. In the case of a schedule, the step dates may be subject to adjustment in accordance with any adjustments specified in calculationPeriodDatesAdjustments.	
lastNotionalStepDate ; built-in datatype: date The unadjusted calculation period end date of the last change in notional. This day may be subject to adjustment in accordance with any adjustments specified in calculationPeriodDatesAdjustments.	FpML_NotionalStepRule
lastRegularPaymentDate ; built-in datatype: date The last regular unadjusted payment date. This day may be subject to adjustment in accordance with any business day convention specified in paymentDatesAdjustments. This element must only be included if there is a final stub. All calculation periods after this date contribute to the final payment. The final payment is made relative to the final set of calculation periods or the final reset date as the case may be. This date will normally correspond to an unadjusted calculation period start or end date. This is true even if early or delayed payment is specified to be applicable since the actual last regular payment date will be the specified number of days before or after the applicable adjusted calculation period start or end date with the resulting payment date then being adjusted in accordance with any business day convention specified in paymentDatesAdjustments.	FpML_PaymentDates
lastRegularPeriodEndDate ; built-in datatype: date The end date of the regular part of the calculation period schedule. It must only be specified if there	FpML_CalculationPeriodDates

is a final stub calculation period. This day may be subject to adjustment in accordance with any adjustments specified in calculationPeriodDatesAdjustments.	
latestExerciseTime ; entity type: FpML_BusinessCenterTime For a Bermudan or American style options, the latest time on an exercise business day (excluding the expiration date) within the exercise period that notice of exercise can be given by buyer to the seller or seller's agent. Notice of exercise given after this time will be deemed to have been given on the next exercise business day.	FpML_AmericanExercise FpML_BermudanExercise
linkId ; built-in datatype: `string` ; coding scheme: `linkIdScheme` A link identifier allowing the trade to be associated with other related trades, e.g. the linkId may contain a tradeId for an associated trade or several related trades may be given the same linkId. FpML does not define the domain values associated with this element. Note that the domain values for this element are not strictly an enumerated list.	FpML_PartyTradeIdentifier
mandatoryEarlyTermination ; entity type: FpML_MandatoryEarlyTermination A mandatory early termination provision to terminate the trade at fair value.	FpML_EarlyTerminationProvision
mandatoryEarlyTerminationAdjustedDates ; entity type: FpML_MandatoryEarlyTerminationAdjustedDates The adjusted dates associated with a mandatory early termination provision. These dates have	FpML_MandatoryEarlyTermination

been adjusted for any applicable business day convention.	
mandatoryEarlyTerminationDate ; entity type: FpML_AdjustableDate The early termination date associated with a mandatory early termination of a swap.	FpML_MandatoryEarlyTermination
manualExercise ; entity type: FpML_ManualExercise Specifies that the notice of exercise must be given by the buyer to the seller or seller's agent.	FpML_ExerciseProcedure
maximumNotionalAmount ; built-in datatype: `decimal` The maximum notional amount that can be exercised on a given exercise date.	FpML_MultipleExercise
minimumNotionalAmount ; built-in datatype: `decimal` The minimum notional amount that can be exercised on a given exercise date. See multipleExercise.	FpML_PartialExercise
multipleExercise ; entity type: FpML_MultipleExercise As defined in the 2000 ISDA Definitions, Section 12.4. Multiple Exercise, the buyer of the option has the right to exercise all or less than all the unexercised notional amount of the underlying swap on one or more days in the exercise period, but on any such day may not exercise less than the minimum notional amount or more than the maximum notional amount, and if an integral	FpML_AmericanExercise FpML_BermudanExercise

multiple amount is specified, the notional amount exercised must be equal to, or be an integral multiple of, the integral multiple amount.	
negativeInterestRateTreatment ; built-in datatype: `string` ; coding scheme: `negativeInterestRateTreatmentScheme` The specification of any provisions for calculating payment obligations when a floating rate is negative (either due to a quoted negative floating rate or by operation of a spread that is subtracted from the floating rate).	FpML_FloatingRateCalculation
notional ; entity type: FpML_Money The notional amount.	FpML_Fra
notionalAmount ; built-in datatype: `decimal` The calculation period notional amount. (FpML_FxLinkedNotionalAmount usage) The notional in the currency of the stream. This notional can be calculated once the FX Spot rate is known. It is optional since it should not be present prior to the fx spot reset date.	FpML_CalculationPeriod FpML_FxLinkedNotionalAmount
notionalReference ; empty element A pointer style reference to the associated notional schedule defined elsewhere in the document.	FpML_ExerciseFee FpML_ExerciseFeeSchedule FpML_PartialExercise
notionalSchedule ; entity type: FpML_Notional The notional amount or notional amount schedule.	FpML_Calculation

notionalStepAmount ; built-in datatype: `decimal` The explicit amount that the notional changes on each step date. This can be a positive or negative amount.	FpML_NotionalStepRule
notionalStepParameters ; entity type: FpML_NotionalStepRule A parametric representation of the notional step schedule, i.e. parameters used to generate the notional schedule.	FpML_Notional
notionalStepRate ; built-in datatype: `decimal` The percentage amount by which the notional changes on each step date. The percentage is either a percentage applied to the initial notional amount or the previous outstanding notional, depending on the value of the element stepRelativeTo. The percentage can be either positive or negative. A percentage of 5% would be represented as 0.05.	FpML_NotionalStepRule
notionalStepSchedule ; entity type: FpML_AmountSchedule The notional amount or notional amount schedule expressed as explicit outstanding notional amounts and dates. In the case of a schedule, the step dates may be subject to adjustment in accordance with any adjustments specified in calculationPeriodDatesAdjustments.	FpML_Notional
observationWeight ; entity type: FpML_positiveInteger	FpML_RateObservation

The number of days weighting to be associated with the rate observation, i.e. the number of days such rate is in effect. This is applicable in the case of a weighted average method of calculation where more than one reset date is established for a single calculation period.	
observedFxSpotRate ; built-in datatype: `decimal` The actual observed fx spot rate.	FpML_FxLinkedNotionalAmount
observedRate ; built-in datatype: `decimal` The actual observed rate before any required rate treatment is applied, e.g. before converting a rate quoted on a discount basis to an equivalent yield. An observed rate of 5% would be represented as 0.05.	FpML_RateObservation
optionalEarlyTermination ; entity type: FpML_OptionalEarlyTermination The option for either or both parties to terminate the swap at fair value.	FpML_EarlyTerminationProvision
optionalEarlyTerminationAdjustedDates ; entity type: FpML_OptionalEarlyTerminationAdjustedDates An early termination provision to terminate the trade at fair value where one or both parties have the right to decide on termination.	FpML_OptionalEarlyTermination
otherPartyPayment ; entity type: FpML_Fee Other fees or additional payments associated with the trade, e.g. broker commissions, where one or	FpML_Trade

more of the parties involved are not principal parties involved in the trade.	
partialExercise ; entity type: FpML_PartialExercise As defined in the 2000 ISDA Definitions, Section 12.3. Partial Exercise, the buyer of the option has the right to exercise all or less than all the notional amount of the underlying swap on the expiration date, but may not exercise less than the minimum notional amount, and if an integral multiple amount is specified, the notional amount exercised must be equal to, or be an integral multiple of, the integral multiple amount.	FpML_EuropeanExercise
party ; entity type: FpML_Party The parties obligated to make payments from time to time during the term of the trade. This will include, at a minimum, the principal parties involved in the swap or forward rate agreement. Other parties paying or receiving fees, commissions etc. must also be specified if referenced in other party payments.	FpML_Trade
partyId ; built-in datatype: `string` ; coding scheme: `partyIdScheme` A party identifier, e.g. a S.W.I.F.T. bank identifier code (BIC).	FpML_Party
partyName ; built-in datatype: `string` The name of the party. A free format string. FpML does not define usage rules for this element	FpML_Party
partyReference ; empty element	FpML_PartyTradeIdentifier

A pointer style reference to a party identifier defined elsewhere in the document. The party referenced has allocated the trade identifier.	
partyTradeIdentifier ; entity type: FpML_PartyTradeIdentifier The trade reference identifier(s) allocated to the trade by the parties involved.	FpML_TradeHeader
parYieldCurveAdjustedMethod ; entity type: FpML_YieldCurveMethod An ISDA defined cash settlement method used for the determination of the applicable cash settlement amount. The method is defined in the 2000 ISDA Definitions, Section 17.3. Cash Settlement Methods, paragraph (c).	FpML_CashSettlement
parYieldCurveUnadjustedMethod ; entity type: FpML_YieldCurveMethod An ISDA defined cash settlement method used for the determination of the applicable cash settlement amount. The method is defined in the 2000 ISDA Definitions, Section 17.3. Cash Settlement Methods, paragraph (e).	FpML_CashSettlement
payerPartyReference ; empty element A pointer style reference to a party identifier defined elsewhere in the document.	FpML_Fee FpML_InterestRateStream
paymentAmount ; entity type: FpML_Money The currency amount of the payment.	FpML_Fee

paymentCalculationPeriod ; entity type: FpML_PaymentCalculationPeriod The adjusted payment date and associated calculation period parameters required to calculate the actual or projected payment amount. A list of payment calculation period elements may be ordered in the document by ascending adjusted payment date. An FpML document containing an unordered list of payment calculation periods is still regarded as a conformant document.	FpML_Cashflows
paymentDate ; entity type: FpML_AdjustableDate The payment date. This date is subject to adjustment in accordance with any applicable business day convention. (FpML_Fee usage) This element is optional to allow the fee component to be used to capture commission amounts that might not have a known payment date associated with them, e.g. commissions may be invoiced and billed periodically.	FpML_Fee FpML_Fra
paymentDates ; entity type: FpML_PaymentDates The payment dates schedule.	FpML_InterestRateStream
paymentDatesAdjustments ; entity type: FpML_BusinessDayAdjustments The business day convention to apply to each payment date if it would otherwise fall on a day that is not a business day in the specified financial business centers.	FpML_PaymentDates
paymentDaysOffset ; entity type: FpML_Offset	FpML_PaymentDates

If early payment or delayed payment is required, specifies the number of days offset that the payment occurs relative to what would otherwise be the unadjusted payment date. The offset can be specified in terms of either calendar or business days. Even in the case of a calendar days offset, the resulting payment date, adjusted for the specified calendar days offset, will still be adjusted in accordance with the specified payment dates adjustments. This element should only be included if early or delayed payment is applicable, i.e. if the periodMultiplier element value is not equal to zero. An early payment would be indicated by a negative periodMultiplier element value and a delayed payment (or payment lag) would be indicated by a positive periodMultiplier element value.	
`paymentFrequency` ; entity type: FpML_Interval The frequency at which regular payment dates occur. If the payment frequency is equal to the frequency defined in the calculation period dates component then one calculation period contributes to each payment amount. If the payment frequency is less frequent than the frequency defined in the calculation period dates component then more than one calculation period will contribute to e payment amount. A payment frequency more frequent than the calculation period frequency or one that is not a multiple of the calculation period frequency is invalid.	FpML_PaymentDates
`paymentType` ; built-in datatype: `string` ; coding scheme: `paymentTypeScheme` A classification of the type of fee or additional payment, e.g. brokerage, upfront fee etc. FpML does not define domain values for this element.	FpML_Fee
`payRelativeTo` ; built-in datatype: `string` ; coding scheme: `payRelativeToScheme`	FpML_PaymentDates

Specifies whether the payments occur relative to each adjusted calculation period start date, adjusted calculation period end date or each reset date. The reset date is applicable in the case of certain euro (former French Franc) floating rate indices. Calculation period start date means relative to the start of the first calculation period contributing to a given payment. Similarly, calculation period end date means the end of the last calculation period contributing to a given payment.	
period ; built-in datatype: `string` ; coding scheme: `periodScheme` A time period, e.g. a day, week, month, year or term of the stream. If the periodMultiplier value is 0 (zero) then period must contain the value D (day).	FpML_Interval
periodMultiplier ; built-in datatype: `integer` A time period multiplier, e.g. 1, 2 or 3 etc. A negative value can be used when specifying an offset relative to another date, e.g. -2 days. If the period value is T (Term) then periodMultiplier must contain the value 1.	FpML_Interval
precision ; entity type: FpML_nonNegativeInteger Specifies the rounding precision in terms of a number of decimal places. Note how a percentage rate rounding of 5 decimal places is expressed as a rounding precision of 7 in the FpML document since the percentage is expressed as a decimal, e.g. 9.876543% (or 0.09876543) being rounded to the nearest 5 decimal places is 9.87654% (or 0.0987654).	FpML_Rounding

premium ; entity type: FpML_Payment The option premium amount payable by buyer to seller on the specified payment date.	FpML_Swaption
principalExchange ; entity type: FpML_PrincipalExchange The initial, intermediate and final principal exchange amounts. Typically required on cross currency interest rate swaps where actual exchanges of principal occur. A list of principal exchange elements may be ordered in the document by ascending adjusted principal exchange date. An FpML document containing an unordered principal exchange list is still regarded as a conformant document.	FpML_Cashflows
principalExchangeAmount ; built-in datatype: `decimal` The principal exchange amount. This amount should be positive if the stream payer is paying the exchange amount and signed negative if they are receiving it.	FpML_PrincipalExchange
principalExchanges ; entity type: FpML_PrincipalExchanges The true/false flags indicating whether initial, intermediate or final exchanges of principal should occur.	FpML_InterestRateStream
productType ; built-in datatype: `string` ; coding scheme: `productTypeScheme` A classification of the type of product. FpML does not define domain values for this element.	FpML_Product

quotationRateType ; built-in datatype: string ; coding scheme: quotationRateTypeScheme Which rate quote is to be observed, either Bid, Mid, Offer or Exercising Party Pays. The meaning of Exercising Party Pays is defined in the 2000 ISDA Definitions, Section 17.2. Certain Definitions Relating to Cash Settlement, paragraph (j)	FpML_CashPriceMethod FpML_YieldCurveMethod
rateCutOffDaysOffset ; entity type: FpML_Offset Specifies the number of business days before the period end date when the rate cut-off date is assumed to apply. The financial business centers associated with determining the rate cut-off date are those specified in the reset dates adjustments. The rate cut-off number of days must be a negative integer (a value of zero would imply no rate cut off applies in which case the rateCutOffDaysOffset element should not be included). The relevant rate for each reset date in the period from, and including, a rate cut-off date to, but excluding, the next applicable period end date (or, in the case of the last calculation period, the termination date) will (solely for purposes of calculating the floating amount payable on the next applicable payment date) be deemed to be the relevant rate in effect on that rate cut-off date. For example, if rate cut-off days for a daily averaging deal is -2 business days, then the refix rate applied on (period end date - 2 days) will also be applied as the reset on (period end date - 1 day), i.e. the actual number of reset dates remains the same but from the rate cut-off date until the period end date, the same refix rate is applied. Note that in the case of several calculation periods contributing to a single payment, the rate cut-off is assumed only to apply to the final calculation period contributing to that payment. The day type associated with the offset must imply a business days offset.	FpML_ResetDates
rateObservation ; entity type:	FpML_FloatingRateDefinition

FpML_RateObservation The details of a particular rate observation, including the fixing date and observed rate. A list of rate observation elements may be ordered in the document by ascending adjusted fixing date. An FpML document containing an unordered list of rate observations is still regarded as a conformant document.	
rateReference ; empty element A pointer style reference to a floating rate component defined as part of a stub calculation period amount component. It is only required when it is necessary to distinguish two rate observations for the same fixing date which could occur when linear interpolation of two different rates occurs for a stub calculation period.	FpML_RateObservation
rateSource ; built-in datatype: `string` ; coding scheme: `informationProviderScheme` An information source for obtaining a market rate. For example Bloomberg, Reuters, Telerate etc.	FpML_InformationSource
rateSourcePage ; built-in datatype: `string` ; coding scheme: `rateSourcePageScheme` A specific page for the rate source for obtaining a market rate.	FpML_InformationSource
rateSourcePageHeading ; built-in datatype: `string` The specific information source page for obtaining a market rate. For example, 3750 (Telerate), LIBO (Reuters) etc.	FpML_InformationSource

rateTreatment ; built-in datatype: `string` ; coding scheme: `rateTreatmentScheme` The specification of any rate conversion which needs to be applied to the observed rate before being used in any calculations. The two common conversions are for securities quoted on a bank discount basis which will need to be converted to either a Money Market Yield or Bond Equivalent Yield. See the Annex to the 2000 ISDA Definitions, Section 7.3. Certain General Definitions Relating to Floating Rate Options, paragraphs (g) and (h) for definitions of these terms.	FpML_FloatingRate
receiverPartyReference ; empty element A pointer style reference to a party identifier defined elsewhere in the document.	FpML_Fee FpML_InterestRateStream
referenceBank ; entity type: FpML_ReferenceBank An institution (party) identified by means of a coding scheme and an optional name.	FpML_CashSettlementReferenceBanks
referenceBankId ; built-in datatype: `string` ; coding scheme: `referenceBankIdScheme` An institution (party) identifier, e.g. a bank identifier code (BIC).	FpML_ReferenceBank
referenceBankName ; built-in datatype: `string` The name of the institution (party). A free format string. FpML does not define usage rules for the element.	FpML_ReferenceBank

relativeDate ; entity type: FpML_RelativeDateOffset A date specified as some offset to another date (the anchor date).	FpML_AdjustableOrRelativeDate FpML_CashSettlementPaymentDate
relativeDates ; entity type: FpML_RelativeDates A series of dates specified as some offset to another series of dates. (the anchor dates).	FpML_AdjustableOrRelativeDates
relevantUnderlyingDate ; entity type: FpML_AdjustableOrRelativeDates The date on the underlying set by the exercise of an option. What this date is depends on the option (eg in a swaption it is the effective date, in a extendible / cancelable provision is is the termination date).	FpML_AmericanExercise FpML_BermudanExercise FpML_EuropeanExercise
resetDates ; entity type: FpML_ResetDates The reset dates schedule. The reset dates schedule only applies for a floating rate stream.	FpML_InterestRateStream
resetDatesAdjustments ; entity type: FpML_BusinessDayAdjustments The business day convention to apply to each reset date if it would otherwise fall on a day that is not a business day in the specified financial business centers.	FpML_ResetDates
resetDatesReference ; empty element A pointer style reference to the associated reset	FpML_PaymentDates

dates component defined elsewhere in the document.	
resetFrequency ; entity type: FpML_ResetFrequency The frequency at which reset dates occur. In the case of a weekly reset frequency, also specifies the day of the week that the reset occurs. If the reset frequency is greater than the calculation period frequency then this implies that more than one reset date is established for each calculation period and some form of rate averaging is applicable.	FpML_ResetDates
resetRelativeTo ; built-in datatype: `string` ; coding scheme: `resetRelativeToScheme` Specifies whether the reset dates are determined with respect to each adjusted calculation period start date or adjusted calculation period end date. If the reset frequency is specified as daily this element must not be included.	FpML_ResetDates
rollConvention ; built-in datatype: `string` ; coding scheme: `rollConventionScheme` Used in conjunction with a frequency and the regular period start date of a calculation period, determines each calculation period end date within the regular part of a calculation period schedule.	FpML_CalculationPeriodFrequency
roundingDirection ; built-in datatype: `string` ; coding scheme: `roundingDirectionScheme` Specifies the rounding direction.	FpML_Rounding

scheduleBounds ; entity type: FpML_DateRange The first and last dates of a schedule. This can be used to restrict the range of values in a reference series of dates.	FpML_RelativeDates
seller ; built-in datatype: `string` ; coding scheme: `payerReceiverScheme` The party that has sold.	FpML_StrikeRate FpML_StrikeRateSchedule
sellerPartyReference ; empty element A pointer style reference to a party identifier defined elsewhere in the document. The party referenced is the seller of the instrument, also known as the floating rate payer.	FpML_Fra FpML_CancelableProvision FpML_ExtendibleProvision FpML_SinglePartyOption FpML_Swaption
settlementRateSource ; entity type: FpML_SettlementRateSource The method for obtaining a settlement rate. This may be from some information source (e.g. Reuters) or from a set of reference banks.	FpML_YieldCurveMethod
singlePartyOption ; entity type: FpML_SinglePartyOption If optional early termination is not available to both parties then this component specifies the buyer and seller of the option.	FpML_OptionalEarlyTermination
spread ; built-in datatype: `decimal` The ISDA Spread, if any, which applies for the calculation period. The spread is a per annum rate, expressed as a decimal. For purposes of	FpML_FloatingRateDefinition

determining a calculation period amount, if positive the spread will be added to the floating rate and if negative the spread will be subtracted from the floating rate. A positive 10 basis point (0.1%) spread would be represented as 0.001.	
spreadSchedule ; entity type: FpML_Schedule The ISDA Spread or a Spread schedule expressed as explicit spreads and dates. In the case of a schedule, the step dates may be subject to adjustment in accordance with any adjustments specified in calculationPeriodDatesAdjustments. The spread is a per annum rate, expressed as a decimal. For purposes of determining a calculation period amount, if positive the spread will be added to the floating rate and if negative the spread will be subtracted from the floating rate. A positive 10 basis point (0.1%) spread would be represented as 0.001.	FpML_FloatingRate
step ; entity type: FpML_Step The schedule of step date and value pairs. On each step date the associated step value becomes effective A list of steps may be ordered in the document by ascending step date. An FpML document containing an unordered list of steps is still regarded as a conformant document.	FpML_Schedule
stepDate ; built-in datatype: date The date on which the associated stepValue becomes effective. This day may be subject to adjustment in accordance with a business day convention.	FpML_Step
stepFrequency ; entity type: FpML_Interval The frequency at which the step changes occur.	FpML_NotionalStepRule

This frequency must be a multiple of the stream calculation period frequency.	
stepRelativeTo ; built-in datatype: `string` ; coding scheme: `stepRelativeToScheme` Specifies whether the notionalStepRate should be applied to the initial notional or the previous notional in order to calculate the notional step change amount.	FpML_NotionalStepRule
stepValue ; built-in datatype: `decimal` The rate or amount which becomes effective on the associated stepDate. A rate of 5% would be represented as 0.05.	FpML_Step
strategy ; entity type: FpML_Strategy A trade containing multiple products. It is envisaged that this will be used to represent structured products.	FpML_Product
strategyType ; built-in datatype: `string` ; coding scheme: `strategyTypeScheme` A string to define the type of the product strategy.	FpML_Strategy
strikeRate ; built-in datatype: `decimal` The rate for a cap or floor.	FpML_StrikeRate
stubAmount ; entity type: FpML_Money An actual amount to apply for the initial or final stub period may have been agreed between the two parties. If an actual stub amount has been	FpML_Stub

agreed then it would be included in this component.	
stubCalculationPeriodAmount ; entity type: FpML_StubCalculationPeriodAmount The stub calculation period amount parameters. This element must only be included if there is an initial or final stub calculation period. Even then, it must only be included if either the stub references a different floating rate tenor to the regular calculation periods, or if the stub is calculated as a linear interpolation of two different floating rate tenors, or if a specific stub rate or stub amount has been negotiated.	FpML_InterestRateStream
stubRate ; built-in datatype: decimal An actual rate to apply for the initial or final stub period may have been agreed between the principal parties (in a similar way to how an initial rate may have been agreed for the first regular period). If an actual stub rate has been agreed then it would be included in this component. It will be a per annum rate, expressed as a decimal. A stub rate of 5% would be represented as 0.05.	FpML_Stub
swap ; entity type: FpML_Swap A swap product definition.	FpML_Product FpML_Swaption
swapStream ; entity type: FpML_InterestRateStream The swap streams.	FpML_Swap
swaption ; entity type: FpML_Swaption A swaption product definition.	FpML_Product

swaptionAdjustedDates ; entity type: FpML_SwaptionAdjustedDates The adjusted dates associated with swaption exercise. These dates have been adjusted for any applicable business day convention.	FpML_Swaption
swaptionStraddle ; built-in datatype: boolean Whether the option is a swaption or a swaption straddle	FpML_Swaption
terminationDate ; entity type: FpML_AdjustableDate The last day of the term of the trade. This day may be subject to adjustment in accordance with a business day convention.	FpML_CalculationPeriodDates
thresholdRate ; built-in datatype: decimal A threshold rate. A threshold of 0.10% would be represented as 0.001.	FpML_AutomaticExercise
trade ; entity type: FpML_Trade The FpML trade definition.	FpML
tradeDate ; built-in datatype: date The trade date.	FpML_TradeHeader
tradeHeader ; entity type:	FpML_Trade

FpML_TradeHeader The information on the trade which is not product specific, e.g. trade date.	
tradeId ; built-in datatype: string ; coding scheme: tradeIdScheme A trade reference identifier allocated by a party. FpML does not define the domain values associated with this element. Note that the domain values for this element are not strictly an enumerated list.	FpML_PartyTradeIdentifier
treatedRate ; built-in datatype: decimal The observed rate after any required rate treatment is applied. A treated rate of 5% would be represented as 0.05.	FpML_RateObservation
unadjustedDate ; built-in datatype: date A date subject to adjustment.	FpML_AdjustableDate FpML_AdjustableDates
unadjustedFirstDate ; built-in datatype: date The first date of a date range.	FpML_DateRange
unadjustedLastDate ; built-in datatype: date The last date of a date range.	FpML_DateRange
varyingNotionalCurrency ; built-in datatype: string ; coding scheme: currencyScheme	FpML_FxLinkedNotionalSchedule

The currency of the varying notional amount, i.e. the notional amount being determined periodically based on observation of a spot currency exchange rate.	
varyingNotionalFixingDates ; entity type: FpML_RelativeDateOffset The dates on which spot currency exchange rates are observed for purposes of determining the varying notional currency amount that will apply to a calculation period.	FpML_FxLinkedNotionalSchedule
varyingNotionalInterimExchangePaymentDates ; entity type: FpML_RelativeDateOffset The dates on which interim exchanges of notional are paid. Interim exchanges will arise as a result of changes in the spot currency exchange amount or changes in the constant notional schedule (e.g. amortization).	FpML_FxLinkedNotionalSchedule
weeklyRollConvention ; built-in datatype: `string` ; coding scheme: `weeklyRollConventionScheme` The day of the week on which a weekly reset date occurs. This element must be included if the reset frequency is defined as weekly and not otherwise.	FpML_ResetFrequency
zeroCouponYieldAdjustedMethod ; entity type: FpML_YieldCurveMethod An ISDA defined cash settlement method used for the determination of the applicable cash settlement amount. The method is defined in the 2000 ISDA Definitions, Section 17.3. Cash Settlement Methods, paragraph (d).	FpML_CashSettlement

8 CHARACTER ENCODING AND CHARACTER REPERTOIRE

8.1 Character Encoding

Producers of FpML documents intended for interchange with other parties must encode such documents using either UTF-8 or UTF-16. Consumers of FpML documents must be able to process documents encoded using UTF-8, as well as documents encoded using UTF-16. For more information, see http://www.w3.org/TR/REC-xml#charencoding.

8.2 Character Repertoire

FpML element content, as well as values of the FpML `id` and `href` attributes, may use any valid XML characters. For more information, see http://www.w3.org/TR/REC-xml#charsets.

9 DATATYPES AND CODING SCHEMES

9.1 Datatypes

FpML 2.0 uses a subset of the built-in datatypes (both primitive and derived datatypes) as defined in XML Schema Part 2: Datatypes, W3C Recommendation 02 May 2001. The built-in datatypes are described at:

http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/ - built-in-datatypes

The built-in datatypes used in FpML 2.0 are the following:

- `boolean`
- `date`
- `decimal`
- `integer`
- `nonNegativeInteger`
- `positiveInteger`
- `string`
- `time`.

The set of valid literals for each datatype are those defined in the XML Schema specification as being its lexical space. Additional constraints are imposed by FpML 2.0 on the `date` and `time` built-in datatypes as described below.

9.1.1 date

All elements of type `date` in FpML must contain date values with the format CCYY-MM-DD where "CC" represents the century, "YY" the year, "MM" the month and "DD" the day. The CCYY field must have at least four digits, the MM and DD fields exactly two digits each; leading zeroes must be used if the field would otherwise have too few digits. A following time zone qualifier is not allowed and year values must be in the range 0001 to 9999. For example, 25 May 2000 would be represented in FpML as 2000-05-25.

9.1.2 time

All elements of type `time` in FpML must represent daily recurring instant of time values with the format hh:mm:ss where "hh", "mm" and "ss" represent hour, minute and second respectively. The hh, mm and ss fields must have exactly two digits each; leading zeroes must be used if the field would otherwise have too few digits. FpML imposes the further restriction that the second (ss) component must be '00' and a time zero qualifier is not allowed. For example, 00:00:00 (midnight), 01:00:00 (1:00am), 12:00:00 (midday), 23:30:00 (11:30pm).

9.2 Coding Schemes

9.2.1 Introduction

A number of data elements defined in the FpML 2.0 DTD are restricted to holding one of a limited set of possible values, e.g. `dayCountConvention`, `dayCountFraction`, `currency` etc. Such restricted sets of values are frequently referred to as domains. XML 1.0 has some limited support for the concept of domains through the use of enumerated attributes.

FpML has adopted the principle of not using attributes to hold business data. As a consequence, XML enumerations are not used and an alternative strategy has been defined by the Architecture Working Group referred to as 'Schemes'. Each Scheme is associated with a URI. Coding Schemes can be categorized as one of the following:

- An external coding Scheme, which has a well-known URI. In this case the URI is assigned by an external body, and may or may not have its own versioning, date syntax and semantics. The external body may be an open standards organization, or it may be a market participant

- An external coding Scheme, which does not have a well-known URI. In this case FpML assigns a URI as a proxy to refer to the concept of the external Scheme, but this URI will not be versioned or dated

- An FpML-defined coding Scheme. In this case the Scheme is fully under FpML control and the URI will change reflecting newer versions and revisions as the scheme evolves and changes.

In this section, the FpML-controlled Schemes and their associated URIs are defined, as well as URIs assigned by FpML to external coding schemes. The URI construction follows the FpML Architecture Version 1.0 recommendation.

Note that FpML does not define a coding Scheme or URI for the following Schemes:

- Link Identifier (linkIdScheme)
- Payment Type (paymentTypeScheme)
- Product Type (productTypeScheme)
- Rate Source Page (rateSourcePageScheme)
- Trade Identifier (tradeIdScheme).

These are currently assumed to be specific to individual organizations or FpML based implementations.

Although the initial set of Schemes are defined in this document we expect that new versions of Schemes will be released from time to time and published separately. Key benefits of using Schemes are that they allow:

- enumerations to be revised without requiring a re-issue of the FpML DTDs
- alternate Schemes to be used without requiring changes to the FpML DTDs.

9.2.2 Averaging Method Scheme (averagingMethodScheme)

Definition

The method of calculation to be used when averaging rates. Per ISDA 2000 Definitions, Section 6.2. Certain Definitions Relating to Floating Amounts.

URI

http://www.fpml.org/spec/2000/averaging-method-1-0

Coding Scheme

Code	Meaning
Unweighted	The arithmetic mean of the relevant rates for each reset date.
Weighted	The arithmetic mean of the relevant rates in effect for each day in a calculation period calculated by multiplying each relevant rate by the number of days such relevant rate is in effect, determining the sum of such products and dividing such sum by the number of days in the calculation period.

9.2.3 Business Center Scheme (businessCenterScheme)

Definition

A financial business center location.

URI

http://www.fpml.org/spec/2000/business-center-1-0

Code Construction

In general, the codes are based on the ISO country code and the English name of the location.

Additional location codes can be built according to the following rules. The first two characters represent the ISO country code, the next two characters represent a) if the location name is one word, the first two letters of the location b) if the location name consists of at least two words, the first letter of the first word followed by the first letter of the second word .

There are exceptions to this rule. For example, the TARGET (Trans-European Automated Real-time Gross settlement Express Transfer system) business center for Euro settlement has a code of EUTA.

This coding scheme is currently consistent with the S.W.I.F.T. Financial Centre scheme used in the MT340/MT360/MT361 message definitions, although FpML controls the Business Center Scheme and it should not be assumed that both schemes will remain synchronized.

Coding Scheme

Code	Meaning
ARBA	Buenos Aires
ATVI	Vienna
AUME	Melbourne
AUSY	Sydney

BEBR	Brussels
BRSP	São Paulo
CAMO	Montreal
CATO	Toronto
CHGE	Geneva
CHZU	Zürich
CLSA	Santiago
CNBE	Beijing
CZPR	Prague
DEFR	Frankfurt
DKCO	Copenhagen
EETA	Tallinn
ESMA	Madrid
EUTA	TARGET (euro 'Business Center')
FIHE	Helsinki
FRPA	Paris
GBLO	London
GRAT	Athens
HKHK	Hong Kong
HUBU	Budapest
IDJA	Jakarta
ILTA	Tel Aviv
ITMI	Milan
ITRO	Rome
JPTO	Tokyo
KRSE	Seoul
LBBE	Beirut
LULU	Luxembourg
MXMC	Mexico City
MYKL	Kuala Lumpur
NLAM	Amsterdam
NOOS	Oslo
NZAU	Auckland
NZWE	Wellington
PAPC	Panama City

PHMA	Manila
PLWA	Warsaw
RUMO	Moscow
SARI	Riyadh
SEST	Stockholm
SGSI	Singapore
SKBR	Bratislava
THBA	Bangkok
TRAN	Ankara
TWTA	Taipei
USCH	Chicago
USLA	Los Angeles
USNY	New York
ZAJO	Johannesburg

9.2.4 Business Day Convention Scheme (businessDayConventionScheme)

Definition

The convention for adjusting any relevant date if it would otherwise fall on a day that is not a valid business day. Note that FRN is included here as a type of business day convention although it does not strictly fall within ISDA's definition of a Business Day Convention and does not conform to the simple definition given above.

URI

http://www.fpml.org/spec/2000/business-day-convention-1-0

Coding Scheme

Code	Meaning
FOLLOWING	The non-business date will be adjusted to the first following day that is a business day.

FRN	Per 2000 ISDA Definitions, Section 4.11. FRN Convention; Eurodollar Convention, i.e. "FRN Convention" or "Eurodollar Convention" means, in respect of either Payment Dates or Period End Dates for a Swap Transaction and a party, that the Payment Dates or Period End Dates of that party will be each day during the term of the Swap Transaction that numerically corresponds to the preceding applicable Payment Date or Period End Date, as the case may be, of that party in the calendar month that is the specified number of months after the month in which the preceding applicable Payment Date or Period End Date occurred (or, in the case of the first applicable Payment Date or Period End Date, the day that numerically corresponds to the Effective Date in the calendar month that is the specified number of months after the month in which the Effective Date occurred), except that (a) if there is not any such numerically corresponding day in a calendar month in which a Payment Date or Period End Date, as the case may be, of that party should occur, then the Payment Date or Period End Date will be the last day that is a Business Day in that month, (b) if a Payment Date or Period End Date, as the case may be, of the party would otherwise fall on a day that is not a Business Day, then the Payment Date or Period End Date will be the first following day that is a Business Day unless that day falls in the next calendar month, in which case the Payment Date or Period End Date will be the first preceding day that is a Business Day, and (c) if the preceding applicable Payment Date or Period End Date, as the case may be, of that party occurred on the last day in a calendar month that was a Business Day, then all subsequent applicable Payment Dates or Period End Dates, as the case may be, of that party prior to the Termination Date will be the last day that is a Business Day in the month that is the specified number of months after the month in which the preceding applicable Payment Date or Period End Date occurred.
MODFOLLOWING	The non-business date will be adjusted to the first following day that is a business day unless that day falls in the next calendar month, in which case that date will be the first preceding day that is a business day.
PRECEDING	The non-business date will be adjusted to the first preceding day that is a business day
MODPRECEDING	The non-business date will be adjusted to the first preceding day that is a business day unless that day falls in the previous calendar month, in which case that date will be the first following day that is a business day.
NONE	The date will not be adjusted if it falls on a day that is not a business day.

9.2.5 Calculation Agent Party Scheme (calculationAgentPartyScheme)

Definition

The specification of how a calculation agent will be determined.

URI

http://www.fpml.org/spec/2001/calculation-agent-party-1-0

Coding Scheme

Code	Meaning

ExercisingParty	The party that gives notice of exercise. Per 2000 ISDA Definitions, Section 11.1. Parties, paragraph (d).
NonExercisingParty	The party that is given notice of exercise. Per 2000 ISDA Definitions, Section 11.1. Parties, paragraph (e).

9.2.6 Compounding Method Scheme (compoundingMethodScheme)

Definition

The compounding calculation method. Per 2000 ISDA Definitions, Section 6.3. Certain Definitions Relating to Compounding.

URI

http://www.fpml.org/spec/2000/compounding-method-1-0

Coding Scheme

Code	Meaning
Flat	Flat compounding.
None	No compounding is to be applied.
Straight	Straight compounding.

9.2.7 Currency Scheme (currencyScheme)

Definition

The code for representation of a currency.

URI

http://www.fpml.org/ext/iso4217

Coding Scheme

A valid currency code as defined by the ISO standard 4217 - Codes for representation of currencies and funds.

9.2.8 Date Relative To Scheme (dateRelativeToScheme)

Definition

The specification of the anchor date when calculating a derived date as a relative offset from this anchor date.

URI

http://www.fpml.org/spec/2001/date-relative-to-2-0

Coding Scheme

Code	Meaning
ResetDate	The derived date will be calculated as a relative offset from the reset date.
CashSettlementPaymentDate	The derived date will be calculated as a relative offset from the Cash Settlement Payment Date.

MandatoryEarlyTerminationDate	The derived date will be calculated as a relative offset from the Mandatory Early Termination Date.
ExerciseDate	The derived date will be calculated as a relative offset from the Exercise Date.

9.2.9 Day Count Fraction Scheme (dayCountFractionScheme)

Definition

The specification for how the number of days between two dates is calculated for purposes of calculation of a fixed or floating payment amount and the basis for how many days are assumed to be in a year. Day Count Fraction is an ISDA term. The equivalent AFB (Association Française des Banques) term is Calculation Basis.

URI

http://www.fpml.org/spec/2000/day-count-fraction-1-0

Coding Scheme

Code	Meaning
1/1	Per Annex to the 2000 ISDA Definitions (June 2000 Version), Section 4.16. Day Count Fraction, paragraph (a), i.e. if "1/1" is specified, 1.
ACT/365.ISDA	Per Annex to the 2000 ISDA Definitions (June 2000 Version), Section 4.16. Day Count Fraction, paragraph (b), i.e. If "Actual/365", "Act/365", "A/365", "Actual/Actual" or "Act/Act" is specified, the actual number of days in the Calculation Period or Compounding Period in respect of which the payment is being made divided by 365 (or, if any portion of that Calculation Period or Compounding Period falls in a leap year, the sum of (i) the actual number of days in that portion of the Calculation Period or Compounding Period falling in a leap year divided by 366 and (ii) the actual number of days in that portion of the Calculation Period or Compounding Period falling in a non-leap year divided by 365).
ACT/ACT.ISMA	The Fixed/Floating Amount will be calculated in accordance with Rule 251 of the statutes, by-laws, rules and recommendations of the International Securities Market Association, as published in April 1999, as applied to straight and convertible bonds issued after December 31, 1998, as though the Fixed/Floating Amount were the interest coupon on such a bond.
ACT/ACT.AFB	The Fixed/Floating Amount will be calculated in accordance with the "BASE EXACT/EXACT" day count fraction, as defined in the "Definitions Communes à plusieurs Additifs Techniques" published by the Association Française des Banques in September 1994.
ACT/365.FIXED	Per Annex to the 2000 ISDA Definitions (June 2000 Version), Section 4.16. Day Count Fraction, paragraph (c), i.e. if "Actual/365 (Fixed)", "Act/365 (Fixed)", "A/365 (Fixed)" or "A/365F" is specified, the actual number of days in the Calculation Period or Compounding Period in respect of which payment is being made divided by 365.

Code	Meaning
ACT/360	Per Annex to the 2000 ISDA Definitions (June 2000 Version), Section 4.16. Day Count Fraction, paragraph (d), i.e. if "Actual/360", "Act/360" or "A/360" is specified, the actual number of days in the Calculation Period or Compounding Period in respect of which payment is being made divided by 360.
30/360	Per Annex to the 2000 ISDA Definitions (June 2000 Version), Section 4.16. Day Count Fraction, paragraph (e), i.e. if "30/360", "360/360" or "Bond Basis" is specified, the number of days in the Calculation Period or Compounding Period in respect of which payment is being made divided by 360 (the number of days to be calculated on the basis of a year of 360 days with 12 30-day months (unless (i) the last day of the Calculation Period or Compounding Period is the 31st day of a month but the first day of the Calculation Period or Compounding Period is a day other than the 30th or 31st day of a month, in which case the month that includes that last day shall not be considered to be shortened to a 30-day month, or (ii) the last day of the Calculation Period or Compounding Period is the last day of the month of February, in which case the month of February shall not be considered to be lengthened to a 30-day month)).
30E/360	Per Annex to the 2000 ISDA Definitions (June 2000 Version), Section 4.16. Day Count Fraction, paragraph (f), i.e. if "30E/360" or "Eurobond Basis" is specified, the number of days in the Calculation Period or Compounding Period in respect of which payment is being made divided by 360 (the number of days to be calculated on the basis of a year of 360 days with 12 30-day months, without regard to the date of the first day or last day of the Calculation Period or Compounding Period unless, in the case of the final Calculation Period or Compounding Period, the Termination Date is the last day of the month of February, in which case the month of February shall not be considered to be lengthened to a 30-day month).

9.2.10 Day Type Scheme (dayTypeScheme)

Definition

A day type classification used in counting the number of days between two dates.

URI

http://www.fpml.org/spec/2000/day-type-1-0

Coding Scheme

Code	Meaning
Business	When calculating the number of days between two dates the count includes only business days.
Calendar	When calculating the number of days between two dates the count includes all calendar days.

9.2.11 Discounting Type Scheme (discountingTypeScheme)

Definition

The method of calculating discounted payment amounts.

URI

http://www.fpml.org/spec/2000/discounting-type-1-0

Coding Scheme

Code	Meaning
Standard	Standard Discounting. Per 2000 ISDA Definitions, Section 8.4. Discounting, paragraph (a).
FRA	FRA Discounting. Per 2000 ISDA Definitions, Section 8.4. Discounting, paragraph (b).

9.2.12 Floating Rate Index Scheme (floatingRateIndexScheme)

Definition

The specification of an ISDA Rate Option for purposes of determining a relevant rate on a given reset date. Several URIs are defined to allow floating rate index code definitions to be associated with specific definitions and provisions published by ISDA.

URI

http://www.fpml.org/ext/isda-2000-definitions

Coding Scheme

Valid ISDA Rate Options as published by ISDA in the Annex to the 2000 ISDA Definitions, Section 7.1. Rate Options, and amended and supplemented through to the `tradeDate` of the trade. Amendments and supplements to the Annex will be deemed to have been made when published by ISDA.

URI

http://www.fpml.org/ext/isda-2000-definitions-june-2000-version-annex

Coding Scheme

Valid ISDA Rate Options as published by ISDA in the Annex to the 2000 ISDA Definitions (June 2000 Version), Section 7.1. Rate Options.

URI

http://www.fpml.org/ext/isda-euro-definitions

Coding Scheme

Valid ISDA Euro Rate Options as published by ISDA in the 1998 ISDA Euro Definitions, Section 3.1. Euro Rate Options.

URI

http://www.fpml.org/ext/isda-1998-supplement

Coding Scheme

Valid ISDA Rate Options as published by ISDA in the 1998 Supplement to the 1991 ISDA Definitions, Section 7.1. Rate Options.

URI

http://www.fpml.org/ext/isda-1991-definitions

Coding Scheme

Valid ISDA Rate Options as published by ISDA in the 1991 ISDA Definitions, Section 7.1. Rate Options.

9.2.13 Information Provider Scheme (informationProviderScheme)

Definition

The specification of a list of information providers and vendors who publish financial markets information. Their information sources will typically be used to determine a relevant market rate, price or index. Note that the current list has been compiled incorporating the Annex to the 2000 ISDA Definitions Section 7.2 – Certain Published and Displayed Sources.

URI

http://www.fpml.org/spec/2001/information-provider-1-0

Coding Scheme

Code	Meaning
BankOfCanada	The central bank of Canada
BankOfJapan	The central bank of Japan
Bloomberg	Bloomberg LP.
FederalReserve	The Federal Reserve, the central bank of the United States.
FHLBSF	The Federal Home Loan Bank of San Francisco, or its successor.
ISDA	International Swaps and Derivatives Association, Inc.
Reuters	Reuters Group Plc.
SAFEX	South African Futures Exchange, or its successor.
Telerate	Telerate, Inc.

9.2.14 Negative Interest Rate Treatment Scheme (negativeInterestRateTreatmentScheme)

Definition

The method of calculating payment obligations when a floating rate is negative (either due to a quoted negative floating rate or by operation of a spread that is subtracted from the floating rate).

URI

http://www.fpml.org/spec/2001/negative-interest-rate-treatment-1-0

Coding Scheme

Code	Meaning
NegativeInterestRateMethod	Negative Interest Rate Method. Per 2000 ISDA Definitions, Section 6.4. Negative Interest Rates, paragraphs (b) and (c).

ZeroInterestRateMethod	Zero Interest Rate Method. Per 2000 ISDA Definitions, Section 6.4. Negative Interest Rates, paragraphs (d) and (e).

9.2.15 Party Identifier Scheme (partyIdScheme)

Definition

The code for identification of parties involved in a trade. Valid bank identifier codes (BICs).

URI

http://www.fpml.org/ext/iso9362

Coding Scheme

Valid BIC codes as defined by the ISO standard 9362 - Bank identifier codes (BIC).

S.W.I.F.T. is the designated registration authority for the assignment of BIC codes. They maintain an online BIC directory at http://www.swift.com/.

9.2.16 Payer Receiver Scheme (payerReceiverScheme)

Definition

The specification of an interest rate stream payer or receiver party.

URI

http://www.fpml.org/spec/2001/payer-receiver-1-0

Coding Scheme

Code	Meaning
Payer	The party identified as the stream payer.
Receiver	The party identified as the stream receiver.

9.2.17 Pay Relative To Scheme (payRelativeToScheme)

Definition

The specification of whether payments occur relative to the calculation period start or end date, or the reset date.

URI

http://www.fpml.org/spec/2000/pay-relative-to-1-0

Coding Scheme

Code	Meaning
CalculationPeriodStartDate	Payments will occur relative to the first day of each calculation period.
CalculationPeriodEndDate	Payments will occur relative to the last day of each calculation period.
ResetDate	Payments will occur relative to the reset date.

9.2.18 Period Scheme (periodScheme)

Definition

The specification of a time period.

URI

http://www.fpml.org/spec/2000/period-1-0

Coding Scheme

Code	Meaning
D	Day
W	Week
M	Month
Y	Year
T	Term. The period commencing on the effective date of the stream and ending on the termination date of the stream.

9.2.19 Quotation Rate Type Scheme (quotationRateTypeScheme)

Definition

The specification of the type of the quotation rate to be obtained from each cash settlement reference bank.

URI

http://www.fpml.org/spec/2000/period-1-0

Coding Scheme

Code	Meaning
Bid	A bid rate.
Ask	An ask rate.
Mid	A mid-market rate.
ExercisingPartyPays	If optional early termination is applicable to a swap transaction, the rate, which may be a bid or ask rate, which would result, if seller is in-the-money, in the higher absolute value of the cash settlement amount, or, is seller is out-of-the-money, in the lower absolute value of the cash settlement amount.

9.2.20 Rate Treatment Scheme (rateTreatmentScheme)

Definition

The specification of methods for converting rates from one basis to another.

URI

http://www.fpml.org/spec/2000/rate-treatment-1-0

Coding Scheme

Code	Meaning

BondEquivalentYield	Bond Equivalent Yield. Per Annex to the 2000 ISDA Definitions (June 2000 Version), Section 7.3. Certain General Definitions Relating to Floating Rate Options, paragraph (g).
MoneyMarketYield	Money Market Yield. Per Annex to the 2000 ISDA Definitions (June 2000 Version), Section 7.3. Certain General Definitions Relating to Floating Rate Options, paragraph (h).

9.2.21 Reference Bank Identifier Scheme (referenceBankIdScheme)

Definition

The code for identification of reference bank parties. Valid bank identifier codes (BICs).

URI

http://www.fpml.org/ext/iso9362

Coding Scheme

Valid BIC codes as defined by the ISO standard 9362 – Bank identifier codes (BIC).

S.W.I.F.T. is the designated registration authority for the assignment of BIC codes. They maintain an online BIC directory at http://www.swift.com

9.2.22 Reset Relative To Scheme (resetRelativeToScheme)

Definition

The specification of whether resets occur relative to the first or last day of a calculation period.

URI

http://www.fpml.org/spec/2000/reset-relative-to-1-0

Coding Scheme

Code	Meaning
CalculationPeriodStartDate	Resets will occur relative to the first day of each calculation period.
CalculationPeriodEndDate	Resets will occur relative to the last day of each calculation period.

9.2.23 Roll Convention Scheme (rollConventionScheme)

Definition

The convention for determining the sequence of calculation period end dates. It is used in conjunction with a specified frequency and the regular period start date of a calculation period, e.g. semi-annual IMM roll dates.

URI

http://www.fpml.org/spec/2000/roll-convention-1-0

Coding Scheme

Code	Meaning
EOM	Rolls on month end dates irrespective of the length of the month and the previous roll day.

Code	Meaning
FRN	Rolls days are determined according to the FRN Convention or Eurodollar Convention. Per 2000 ISDA Definitions, Section 4.11. FRN Convention; Eurodollar Convention.
IMM	IMM Settlement Dates. The third Wednesday of the (delivery) month. Per 2000 ISDA Definitions, Section 4.17. IMM Settlement Dates.
IMMCAD	The Monday before the third Wednesday of the (delivery) month.
SFE	Sydney Futures Exchange 90-Day Bank Accepted Bill Futures Settlement Dates. The second Friday of the (delivery) month. Per Sydney Futures Exchange Contract Specification.
NONE	The roll convention is not required. For example, in the case of a daily calculation frequency.
TBILL	13-week and 26-week U.S. Treasury Bill Auction Dates. Each Monday except for U.S. (New York) holidays when it will occur on a Tuesday.
1	Rolls on the 1st day of the month.
2	Rolls on the 2nd day of the month.
3	Rolls on the 3rd day of the month.
4	Rolls on the 4th day of the month.
5	Rolls on the 5th day of the month.
6	Rolls on the 6thday of the month.
7	Rolls on the 7th day of the month.
8	Rolls on the 8th day of the month.
9	Rolls on the 9th day of the month.
10	Rolls on the 10th day of the month.
11	Rolls on the 11th day of the month.
12	Rolls on the 12th day of the month.
13	Rolls on the 13th day of the month.
14	Rolls on the 14th day of the month.
15	Rolls on the 15th day of the month.
16	Rolls on the 16th day of the month.
17	Rolls on the 17thday of the month.
18	Rolls on the 18th day of the month.
19	Rolls on the 19th day of the month.
20	Rolls on the 20th day of the month.
21	Rolls on the 21st day of the month.
22	Rolls on the 22nd day of the month.
23	Rolls on the 23rd day of the month.
24	Rolls on the 24th day of the month.
25	Rolls on the 25th day of the month.
26	Rolls on the 26th day of the month.
27	Rolls on the 27th day of the month.
28	Rolls on the 28th day of the month.
29	Rolls on the 29th day of the month.
30	Rolls on the 30th day of the month.
MON	Rolls weekly on a Monday.
TUE	Rolls weekly on a Tuesday.
WED	Rolls weekly on a Wednesday.
THU	Rolls weekly on a Thursday.
FRI	Rolls weekly on a Friday.
SAT	Rolls weekly on a Saturday.

Code	Meaning
SUN	Rolls weekly on a Sunday.

9.2.24 Rounding Direction Scheme (roundingDirectionScheme)

Definition

The method of rounding a fractional number.

URI

http://www.fpml.org/spec/2000/rounding-direction-1-0

Coding Scheme

Code	Meaning
Up	A fractional number will be rounded up to the specified number of decimal places (the precision). For example, 5.21 and 5.25 rounded up to 1 decimal place are 5.3 and 5.3 respectively.
Down	A fractional number will be rounded down to the specified number of decimal places (the precision). For example, 5.29 and 5.25 rounded down to 1 decimal place are 5.2 and 5.2 respectively.
Nearest	A fractional number will be rounded either up or down to the specified number of decimal places (the precision) depending on its value. For example, 5.24 would be rounded down to 5.2 and 5.25 would be rounded up to 5.3 if a precision of 1 decimal place were specified.

9.2.25 Step Relative To Scheme (stepRelativeToScheme)

Definition

The specification of whether a percentage rate change, used to calculate a change in notional outstanding, is expressed as a percentage of the initial notional amount or the previously outstanding notional amount.

URI

http://www.fpml.org/spec/2000/step-relative-to-1-0

Coding Scheme

Code	Meaning
Initial	Change in notional to be applied is calculated by multiplying the percentage rate by the initial notional amount.
Previous	Change in notional to be applied is calculated by multiplying the percentage rate by the previously outstanding notional amount.

9.2.26 Weekly Roll Convention Scheme (weeklyRollConventionScheme)

Definition

The specification of a weekly roll day.

URI

http://www.fpml.org/spec/2000/weekly-roll-convention-1-0

Coding Scheme

Code	Meaning
MON	Monday
TUE	Tuesday
WED	Wednesday
THU	Thursday
FRI	Friday
SAT	Saturday
SUN	Sunday

10 SAMPLE FPML

10.1 Introduction

This section contains twenty eight example FpML trades. Each example illustrates how different product features are modeled in FpML.

The twenty eight examples are the following:

1. Fixed/floating single currency interest rate swap
2. Fixed/floating single currency interest rate swap with initial stub period and notional amortization
3. Fixed/floating single currency interest rate swap with compounding, payment delay and final rate rounding
4. Fixed/floating single currency interest rate swap with arrears reset, step-up coupon and upfront fee
5. Fixed/floating single currency interest rate swap with long initial stub and short final stub
6. Fixed/floating cross currency interest rate swap
7. Fixed/floating overnight interest rate swap (OIS)
8. Forward rate agreement.
9. European Swaption with physical settlement.
10. European Swaption with physical settlement and relative underlying effective date.
11. European Swaption with physical settlement, partial exercise and automatic exercise.
12. European Swaption Straddle with cash settlement.
13. European Swaption with cash settlement. Cashflows included.
14. Bermudan Swaption with physical settlement.
15. American Swaption with physical settlement.
16. Fixed/Floating single currency interest rate swap with a mandatory early termination clause.
17. Fixed/Floating single currency interest rate swap with a european style optional early termination clause.
18. Fixed/Floating single currency interest rate swap with bermudan style optional early termination clause. Cashflows included.
19. Fixed/Floating single currency interest rate swap with american style optional early termination clause.
20. Fixed/Floating single currency interest rate swap with european cancelable provision.
21. Fixed/Floating single currency interest rate swap with european extendible provision.
22. Interest Rate Cap
23. Interest Rate Floor
24. Interest Rate Collar
25. Fixed/Floating cross currency interest rate swap where the floating stream notional is reset based on the prevailing spot exchange rate. (FX Resetable Swap).
26. Fixed/Floating cross currency interest rate swap where the floating stream notional is reset based on the prevailing spot exchange rate. (FX Resetable Swap). Cashflows included
27. Inverse Floating single currency interest rate swap.
28. Bullet Payments.

Example 5 shows the defaulted 'type' attributes as part of the sample document. This illustrates the additional content model information available to a validating parser when processing an FpML document.

The sample xml document are available for download from the fpml.org website.

10.2 Example 1 - Fixed/Floating Single Currency Interest Rate Swap

File: example_1.xml

On 12 December, 1994 Chase New York and Barclays Bank London enter into an ISDA swap agreement with each other. The terms of the contract are:

- Effective Date: 14 December, 1994
- Termination Date: 14 December, 1999
- Notional Amount: DEM 50,000,000
- Chase pays the floating rate every 6 months, based on 6-month DEM-LIBOR-BBA, on an ACT/360 basis
- Barclays pays the 6% fixed rate every year on a 30E/360 basis
- The swap is non compounding, non amortizing and there are no stub periods. There is no averaging of rates. The business day convention for adjusting the calculation dates is the same as that used for payment date adjustments.

Note the following:

- This example is identical to the MT360 Example 1 message in the S.W.I.F.T. User Handbook (Page 361, Category 3 - Treasury Markets - Foreign Exchange, Money Markets & Derivatives - October 1998 Standards Release - August 1998 Edition)
- Optional cashflows are not included in this example
- The `floatingRateIndexScheme` refers to the 1991 ISDA Definitions.

10.3 Example 2 - Fixed/Floating Single Currency Interest Rate Swap with Initial Stub Period and Notional Amortization

File: example_2.xml

The swap contract is identical to Example 1 except that there is an initial stub period and the notional amortizes.

The rate for the stub period is the linear interpolation between the 4-month and 5-month DEM-LIBOR-BBA rates.

The stub period on the floating stream runs from 16 January, 1995 to 14 June, 1995, and on the fixed stream from 16 January, 1995 to 14 December, 1995.

The notional amount is decreased by DEM 10,000,000 each year.

Note the following:

- This example is identical to the MT360 Example 2 message in the S.W.I.F.T. User Handbook (Page 364, Category 3 - Treasury Markets - Foreign Exchange, Money Markets & Derivatives - October 1998 Standards Release - August 1998 Edition)

- Optional cashflows are included. An assumption that all weekdays are good business days has been made in calculating the adjusted dates in the cashflows

- The `floatingRateIndexScheme` refers to the 1991 ISDA Definitions.

10.4 Example 3 - Fixed/Floating Single Currency Interest Rate Swap with Compounding, Payment Delay and Final Rate Rounding

File: example_3.xml

On 25 April, 2000 Morgan Stanley Dean Witter and JPMorgan enter into an ISDA swap agreement with each other. The terms of the contract are:

- Effective Date: 27 April, 2000
- Termination Date: 27 April, 2002
- Notional Amount: USD 100,000,000
- JPMorgan pays the 5.85% fixed rate semi-annually on a 30/360 basis.
- Morgan Stanley Dean Witter pays the floating rate semi-annually, based on 3-month USD-LIBOR-BBA reset and compounded flat quarterly, on an ACT/360 basis. The compounded rate to be used for calculating each floating payment amount will be rounded to the nearest 5 decimal places. Note how a percentage rate rounding of 5 decimal places is expressed as a rounding precision of 7 in the FpML document since the percentage is expressed as a decimal, e.g. 9.876543% (or 0.09876543) being rounded to the nearest 5 decimal places is 9.87654% (or 0.0987654)
- The business day convention for adjusting the calculation dates is the same as that used for payment date adjustments. There is a payment delay of 5 business days.

Note the following:

- Optional cashflows are included. An assumption that all weekdays are good business days has been made in calculating the adjusted dates in the cashflows
- The `floatingRateIndexScheme` refers to the 1998 Supplement to the 1991 ISDA Definitions.

10.5 Example 4 - Fixed/Floating Single Currency Interest Rate Swap with Arrears Reset, Step-Up Coupon and Upfront Fee

File: example_4.xml

On 25 April, 2000 Morgan Stanley Dean Witter and JPMorgan enter into an ISDA swap agreement with each other. The terms of the contract are:

- Effective Date: 27 April, 2000
- Termination Date: 27 April, 2002
- Notional amount: USD 100,000,000
- JPMorgan pays a 6.0% fixed rate semi-annually on a 30/360 basis for the first year and a fixed rate of 6.5% for the final year
- Morgan Stanley Dean Witter pays the floating rate quarterly, based on 3-month USD-LIBOR-BBA reset in arrears, on an ACT/360 basis
- There is no adjustment to period end dates on the fixed stream, i.e. the business day convention used for adjusting the payment dates does not apply for adjusting the calculation dates
- There is an upfront fee of USD 15,000 payable by Morgan Stanley Dean Witter to JPMorgan on the Effective Date.

Note the following:

- Optional cashflows are not included in this example
- The `floatingRateIndexScheme` refers to the 1998 Supplement to the 1991 ISDA Definitions.

10.6 Example 5 - Fixed/Floating Single Currency Interest Rate Swap with Long Initial Stub and Short Final Stub

File: example_5.xml

On 3 April, 2000 Chase and UBS Warburg enter into an ISDA swap agreement with each other. The terms of the contract are:

- Effective Date: 5 April, 2000
- Termination Date: 5 January, 2005
- Notional Amount: EUR 75,000,000
- Chase pays the floating rate every 6 months, based on 6-month EUR-EURIBOR-Telerate plus 10 basis points spread, on an ACT/360 basis
- UBS Warburg pays the 5.25% fixed rate every year on a 30/360 basis
- There is a long initial stub period of 7 months. The first period runs from 5 March, 2000 to 5 October, 2000 and an initial stub rate of 5.125% has been agreed for this period on the floating stream
- There is a short final stub period of 3 months. The final period runs from 5 October, 2004 to 5 January, 2005 and the 3-month EUR-EURIBOR-Telerate rate will be used for this period on the floating stream
- The business day convention for adjusting the calculation dates is the same as that used for payment date adjustments.

Note the following:

- The optional cashflows are not shown in this example
- This example shows the defaulted 'type' attributes to illustrate the additional content model information available to a validating parser. Whilst it is not invalid to include this information in the XML document instance, it is not recommended to do so, as any inconsistencies between the type information specified in the document and that in the DTD will result in a well formed but invalid FpML document
- The `floatingRateIndexScheme` refers to the 1998 ISDA Euro Definitions.

10.7 Example 6 - Fixed/Floating Cross Currency Interest Rate Swap

File: example_6.xml

On 12 December, 1994 Chase New York and Barclays Bank London enter into an ISDA cross-currency swap agreement with each other. The terms of the contract are:

- Effective Date: 14 December, 1994
- Termination Date: 14 December, 1999
- Chase pays the floating rate every 6 months, based on 6-month USD-LIBOR-BBA, on USD 10,000,000 and an ACT/360 basis
- Barclays pays the 6% fixed rate every year on JPY 1,000,000,000 and a 30E/360 basis
- The swap is non compounding, non amortizing and there are no stub periods. There is no averaging of rates. The business day convention for adjusting the calculation dates is the same as that used for payment date adjustments.

Note the following:

- This example is identical to the MT361 Example 1 message in the S.W.I.F.T. User Handbook (Page 477, Category 3 - Treasury Markets - Foreign Exchange, Money Markets & Derivatives - October 1998 Standards Release - August 1998 Edition)
- Optional cashflows are included. An assumption that all weekdays are good business days has been made in calculating the adjusted dates in the cashflows
- The `floatingRateIndexScheme` refers to the 1991 ISDA Definitions.

10.8 Example 7 – Fixed/Floating Overnight Interest Rate Swap (OIS)

File: example_7.xml

On 25 January, 2001 Citibank and Mizuho Capital enter into an ISDA swap agreement with each other. The terms of the contract are:

- Effective Date: 29 January, 2001
- Termination Date: 29 April, 2001
- Notional Amount: EUR 100,000,000
- Citibank makes a single floating rate payment at maturity based on the self-compounding floating rate index EUR-EONIA-OIS-COMPOUND, on an ACT/360 basis. The payment is delayed by one TARGET settlement day
- Mizuho Capital makes a single fixed rate payment at maturity based on a fixed rate of 5.1%, on an ACT/360 basis. The payment is delayed by one TARGET settlement day.

Note the following:

- Optional cashflows are not included in this example
- The `floatingRateIndexScheme` refers to the 2000 ISDA Definitions
- The `calculationPeriodFrequency`, `paymentFrequency` and `resetFrequency` are all specified as 'Term' since payments on the fixed and floating streams occur only at maturity and there is a single calculation period. The `rollConvention` is specified as 'None'
- The floating rate reset date is the last day of the calculation period. The ISDA definition of the OIS floating rate index provides for the compounding of the overnight deposit rates to occur in the process of arriving at the floating rate. There is no need to specify compounding of the rate separately, i.e. `calculationPeriodFrequency` and `paymentFrequency` are the same and no `compoundingMethod` is specified
- The fixing date is equal to the reset date
- There is no `indexTenor` (designated maturity) specified for the OIS floating rate index
- The calculation agent is Citibank.

10.9 Example 8 - Forward Rate Agreement

File: example_8.xml

On 14 May, 1991 ABN AMRO Bank and Midland Bank enter a Forward Rate Agreement in which ABN AMRO is the seller of the notional contract amount and Midland the buyer. The terms of the contract are:

- Effective Date: 17 July, 1991
- Termination Date: 17 January, 1992
- Notional Amount: CHF 25,000,000
- Fixed Rate: 4.0%
- Day Count Fraction: Actual/360

Note the following:

- This example is identical to the MT340 Example message in the S.W.I.F.T. User Handbook (Page 243, Category 3 - Treasury Markets - Foreign Exchange, Money Markets & Derivatives - October 1998 Standards Release - August 1998 Edition).
- The `floatingRateIndexScheme` refers to the 1991 ISDA Definitions.

10.10 Example 9 – European Swaption, Physical Settlement, Explicit Underlying Effective Date

File: example_9.xml

On 30 August, 2000 Party buys from PartyB an option to exercise into an underlying ISDA swap. The terms of the contract are:

- PartyA pays to partyB a premium of EUR 100000, on 30 August, 2000.
- The Option Expires on 28th August, 2001.
- The Option should be exercised no earlier than 09:00 hours Brussels time, and no later than 11:00 hours Brussels time
- Follow-up confirmation of the exercise decision is required.
- Effective Date of the Underlying Swap: 30 August, 2001
- Termination Date of the Underlying Swap: 30 August, 2006
- Notional on the Underlying Swap Amount: EUR 100,000,000
- Should the option be exercised, PartyA makes semi-annual floating rate payments based on the floating rate index EUR-EURIBOR-Telerate, on an ACT/360 basis.
- Should the option be exercised, PartyB makes annual fixed rate payments based on a fixed rate of 5.0%, on an 30/360 basis.

Note the following:

- The Calculation agent is partyB
- The notification party is partyB, i.e. it is to partyB that notice of exercise must be given.
- The Swap is not specified with cashflows.
- The options settles physically.
- The effective date of the underlying swap is explicitly set as 30 August, 2001 by virtue of the fact that there is no relevantUnderlyingDate element set.

10.11 Example 10 – European Swaption, Physical Settlement, Relative Underlying Effective Date

File: example10.xml

On 30 August, 2000 Party buys from PartyB an option to exercise into an underlying ISDA swap. The terms of the contract are:

- PartyA pays to partyB a premium of EUR 100000, on 30 August, 2000.
- The Option Expires on 28th August, 2001.
- The Option should be exercised no earlier than 09:00 hours Brussels time, and no later than 11:00 hours Brussels time
- Follow-up confirmation of the exercise decision is required.
- Effective Date of the Underlying Swap is defined as being 2 days after the Exercise Date.
- Termination Date of the Underlying Swap: 30 August, 2006
- Notional on the Underlying Swap Amount: EUR 100,000,000
- Should the option be exercised, PartyA makes semi-annual floating rate payments based on the floating rate index EUR-EURIBOR-Telerate, on an ACT/360 basis.
- Should the option be exercised, PartyB makes annual fixed rate payments based on a fixed rate of 5.0%, on an 30/360 basis.

10.12 Example 11 – European Swaption, Physical Settlement, Partial Exercise, Automatic Exercise

File: example11.xml

On 30 August, 2000 Party buys from PartyB an option to exercise into an underlying ISDA swap. The terms of the contract are:

- PartyA pays to partyB a premium of EUR 100000, on 30 August, 2000.
- The Option Expires on 28th August, 2001.
- The option is exercised automatically where the threshold rate for exercise is set as 2 basis points.
- There is allowance for partial exercise, where the minimum notional amount is EUR 50,000,000 increasing in multiples of EUR 10,000,000.
- Effective Date of the Underlying Swap: 30 August 2001.
- Termination Date of the Underlying Swap: 30 August, 2006
- Notional on the Underlying Swap Amount: EUR 100,000,000
- Should the option be exercised, PartyA makes semi-annual floating rate payments based on the floating rate index EUR-EURIBOR-Telerate, on an ACT/360 basis.
- Should the option be exercised, PartyB makes annual fixed rate payments based on a fixed rate of 5.0%, on an 30/360 basis.

10.13 Example 12 – European Swaption, Cash Settlement, Swaption Straddle

File: example12.xml

On 30 August, 2000 Party buys from PartyB an option to exercise into an underlying ISDA swap. The terms of the contract are:

- PartyA pays to partyB a premium of EUR 100000, on 30 August, 2000.
- The Option Expires on 28th August, 2001.
- The Option should be exercised no earlier than 09:00 hours Brussels time, and no later than 11:00 hours Brussels time
- The exercise, settlement is made in cash with valuation being performed using the yield curve unadjusted method (rate source – ISDA, rate type – Mid).
- Follow-up confirmation of the exercise decision is required.
- Effective Date of the Underlying Swap: 30 August, 2001
- Termination Date of the Underlying Swap: 30 August, 2006
- Notional on the Underlying Swap Amount: EUR 100,000,000
- The Option held is a straddle, therefore, on exercise, PartyA will either
    - Make semi-annual floating rate payments based on the floating rate index EUR-EURIBOR-Telerate, on an ACT/360 basis, and receive annual fixed rate payments based on a fixed rate of 5.0%, on an 30/360 basis.

    or

    - Make annual fixed rate payments based on a fixed rate of 5.0%, on an 30/360 basis and receive semi-annual floating rate payments based on the floating rate index EUR-EURIBOR-Telerate, on an ACT/360 basis.

10.14 Example 13 – European Swaption, Cash Settled, cashflows included

File: example13.xml

On 30 August, 2000 Party buys from PartyB an option to exercise into an underlying ISDA swap. The terms of the contract are:

- PartyA pays to partyB a premium of EUR 100000, on 30 August, 2000.
- The Option Expires on 28th August, 2001.
- The Option should be exercised no earlier than 09:00 hours Brussels time, and no later than 11:00 hours Brussels time
- The exercise, settlement is made in cash with valuation being performed using the yield curve unadjusted method (rate source – ISDA, rate type – Mid).
- Follow-up confirmation of the exercise decision is required.
- Effective Date of the Underlying Swap: 30 August, 2001
- Termination Date of the Underlying Swap: 30 August, 2006
- Notional on the Underlying Swap Amount: EUR 100,000,000
- Should the option be exercised, PartyA makes semi-annual floating rate payments based on the floating rate index EUR-EURIBOR-Telerate, on an ACT/360 basis.
- Should the option be exercised, PartyB makes annual fixed rate payments based on a fixed rate of 5.0%, on an 30/360 basis.

Note the following:

- The Calculation agent is partyB
- The swaption is specified with its adjusted exercise date..
- The Swap is specified with cashflows included

10.15 Example 14 – Bermudan Swaption, Physical Settlement.

File: example14.xml

On 30 August, 2000 Party buys from PartyB an option to exercise into an underlying ISDA swap. The terms of the contract are:

- PartyA pays to partyB a premium of EUR 100000, on 30 August, 2000.
- The Option can be exercised the following dates….
  28 December, 2000
  28 April, 2000
  28 August, 2000
- The Option should be exercised on these dates no earlier than 09:00 hours Brussels time, and no later than 11:00 hours Brussels time
- Follow-up confirmation of the exercise decision is required.
- Effective Date of the Underlying Swap: 30 August, 2001
- Termination Date of the Underlying Swap: 30 August, 2006
- Notional on the Underlying Swap Amount: EUR 100,000,000
- Should the option be exercised, PartyA makes semi-annual floating rate payments based on the floating rate index EUR-EURIBOR-Telerate, on an ACT/360 basis.
- Should the option be exercised, PartyB makes annual fixed rate payments based on a fixed rate of 5.0%, on an 30/360 basis.

Note the following:

- The Calculation agent is partyB
- The options settles physically.

10.16 Example 15 – American Swaption, Physical Settlement.

File: example15.xml

On 30 August, 2000 Party buys from PartyB an option to exercise into an underlying ISDA swap. The terms of the contract are:

- PartyA pays to partyB a premium of EUR 100000, on 30 August, 2000.
- The Option can be exercised on any date from 30 August 2000 to 30 August 2002.
- The Option should be exercised on these dates no earlier than 09:00 hours Brussels time, and no later than 11:00 hours Brussels time
- Follow-up confirmation of the exercise decision is required.
- Effective Date of the Underlying Swap will be 2 days after the exercise date.
- Termination Date of the Underlying Swap: 30 August, 2006
- Notional on the Underlying Swap Amount: EUR 100,000,000
- Should the option be exercised, PartyA makes semi-annual floating rate payments based on the floating rate index EUR-EURIBOR-Telerate, on an ACT/360 basis.
- Should the option be exercised, PartyB makes annual fixed rate payments based on a fixed rate of 5.0%, on an 30/360 basis.

Note the following:

- The Calculation agent is partyB
- The options settles physically.

10.17 Example 16 – Fixed/Floating Single Currency IRS With Mandatory Early Termination.

File: example16.xml

On 30 August, 2000 PartyA and PartyB agree to enter into an ISDA swap with early termination provision. The terms of the contract are:

- Effective Date of the Swap: 30 August 2001.
- Termination Date of the Underlying Swap: 30 August, 2006
- Notional on the Underlying Swap Amount: EUR 100,000,000
- PartyA makes semi-annual floating rate payments based on the floating rate index EUR-EURIBOR-Telerate, on an ACT/360 basis.
- PartyB makes annual fixed rate payments based on a fixed rate of 5.0%, on an 30/360 basis.
- The will terminate on the 30 August 2001.
- Cash settlement will be made on this date with valuation taking place 2 days prior to settlement at 11:00 hours (Brussels time).
- The Swap will be valued at this lime using the cash-price method

Note the following:

- The partyA and partyB are joint calculation agents

10.18 Example 17 – Fixed/Floating Single Currency IRS With European Style Optional Early Termination.

File: example17.xml

On 30 August, 2000 PartyA and PartyB agree to enter into an ISDA swap with early termination provision. The terms of the contract are:

- Effective Date of the Swap: 30 August 2001.
- Termination Date of the Underlying Swap: 30 August, 2006
- Notional on the Underlying Swap Amount: EUR 100,000,000
- PartyA makes semi-annual floating rate payments based on the floating rate index EUR-EURIBOR-Telerate, on an ACT/360 basis.
- PartyB makes annual fixed rate payments based on a fixed rate of 5.0%, on an 30/360 basis.
- The partyA has a chance to terminate the swap early – cash-settling on 30 August 2001. Notification of this needs to be given 5 days prior to this date after 9:00 hours (Brussels time) and not after (11:00 hours Brussels time)
- Cash settlement will be made on this date with valuation taking place 2 days prior to settlement at 11:00 hours (Brussels time).
- The Swap will be valued at this time using the cash-price method

10.19 Example 18 – Fixed/Floating Single Currency IRS With Bermudan Style Optional Early Termination, Cashflows + optionalEarlyTerminationAdjustedDates.

File: example18.xml

On 30 August, 2000 PartyA and PartyB agree to enter into an ISDA swap with early termination provision. The terms of the contract are:

- Effective Date of the Swap: 30 August 2001.
- Termination Date of the Underlying Swap: 30 August, 2006
- Notional on the Underlying Swap Amount: EUR 100,000,000
- PartyA makes semi-annual floating rate payments based on the floating rate index EUR-EURIBOR-Telerate, on an ACT/360 basis.
- PartyB makes annual fixed rate payments based on a fixed rate of 5.0%, on an 30/360 basis.
- The partyA has a chance to terminate the swap early – cash-settling either 30 August 2003, or 30 August 2004. Notification of this needs to be given 5 days prior to this date after 9:00 hours (Brussels time) and not after (11:00 hours Brussels time)
- Cash settlement will be made on this date with valuation taking place 2 days prior to settlement at 11:00 hours (Brussels time).
- The Swap will be valued at this time using the cash-price method

Note the following:

- The swap is defined with cashflows.

10.20 Example 19 – Fixed/Floating Single Currency IRS With American Style Optional Early Termination.

File: example19.xml

On 30 August, 2000 PartyA and PartyB agree to enter into an ISDA swap with early termination provision. The terms of the contract are:

- Effective Date of the Swap: 30 August 2001.
- Termination Date of the Underlying Swap: 30 August, 2011
- Notional on the Underlying Swap Amount: EUR 100,000,000
- PartyA makes semi-annual floating rate payments based on the floating rate index EUR-EURIBOR-Telerate, on an ACT/360 basis.
- PartyB makes annual fixed rate payments based on a fixed rate of 5.0%, on an 30/360 basis.
- The partyA has a chance to terminate the swap early – cash-settling any time between 30 August 2001 and 30 August 2006. Notification of this needs to be given 5 days prior to this date after 9:00 hours (Brussels time) and not after (11:00 hours Brussels time)
- Cash settlement will be made on this date with valuation taking place 2 days prior to settlement at 11:00 hours (Brussels time).
- The Swap will be valued at this time using the cash-price method

10.21 Example 20 – Fixed/Floating Single Currency IRS With European Cancelable Provision.

File: example20.xml

On 30 August, 2000 PartyA and PartyB agree to enter into an ISDA swap with Cancelable provision. The terms of the contract are:

- Effective Date of the Swap: 30 August 2001.
- Termination Date of the Underlying Swap: 30 August, 2011
- Notional on the Underlying Swap Amount: EUR 100,000,000
- PartyB makes semi-annual floating rate payments based on the floating rate index EUR-EURIBOR-Telerate, on an ACT/360 basis.
- PartyA makes annual fixed rate payments based on a fixed rate of 5.0%, on an 30/360 basis.
- The partyB has a chance to cancel the swap after five years (30 August 2006) giving notification 15 days prior to this date after 9:00 hours (Brussels time) and not after (11:00 hours Brussels time)

10.22 Example 21 – Fixed/Floating Single Currency IRS With European Extendible Provision.

File: example21.xml

On 30 August, 2000 PartyA and PartyB agree to enter into an ISDA swap with Extendible provision. The terms of the contract are:

- Effective Date of the Swap: 30 August 2001.
- Termination Date of the Underlying Swap: 30 August, 2006
- Notional on the Underlying Swap Amount: EUR 100,000,000
- PartyB makes semi-annual floating rate payments based on the floating rate index EUR-EURIBOR-Telerate, on an ACT/360 basis.
- PartyA makes annual fixed rate payments based on a fixed rate of 5.0%, on an 30/360 basis.
- The partyA has a chance to extend the swap after five years (30 August 2006) giving notification 15 days prior to this date after 9:00 hours (Brussels time) and not after (11:00 hours Brussels time). If extended, the swap will continue until 30 August 2011

10.23 Example 22 – Interest Rate Cap

File: example22.xml

On 30 August, 2000 PartyA sells to PartyB an interest rate cap. The terms of the contract are:

- Effective Date of the Swap: 30 August 2001.
- Termination Date of the Underlying Swap: 30 August, 2006
- Notional on the Underlying Swap Amount: EUR 100,000,000
- PartyA sells partyB a stepped cap (initial rate of 6%) on semi-annual floating rate payments based on the floating rate index EUR-EURIBOR-Telerate, on an ACT/360 basis (partyB being the payer of the floating rate).

Note the following:

- The cap rate schedule defines annual 'step up' intervals hence keeping the same strike for 2 successive caplets.

10.24 Example 23 – Interest Rate Floor

File: example23.xml

On 30 August, 2000 PartyA sells to PartyB an interest rate floor. The terms of the contract are:

- Effective Date of the Swap: 30 August 2001.
- Termination Date of the Underlying Swap: 30 August, 2006
- Notional on the Underlying Swap Amount: EUR 100,000,000
- PartyA sells partyB a stepped floor (initial rate of 4%) on semi-annual floating rate payments based on the floating rate index EUR-EURIBOR-Telerate, on an ACT/360 basis (partyB being the receiver of the floating rate).

Note the following:

- The floor rate schedule defines annual ‘step up’ intervals hence keeping the same strike for 2 successive floorlets..

10.25 Example 24 – Interest Rate Collar

File: example24.xml

On 30 August, 2000 PartyA sells to PartyB an interest rate floor. The terms of the contract are:

- Effective Date of the Swap: 30 August 2001.
- Termination Date of the Underlying Swap: 30 August, 2006
- Notional on the Underlying Swap Amount: EUR 100,000,000
- PartyA sells partyB a stepped collar (initially between 4% and 6%) on semi-annual floating rate payments based on the floating rate index EUR-EURIBOR-Telerate, on an ACT/360 basis (partyB being the payer of the floating rate).

Note the following:

- The cap and floor rate schedule defines annual 'step up' intervals hence keeping the same strike for 2 successive caplets and floorlets respectively.

10.26 Example 25 – Fixed/Floating IRS Where The Floating Stream Notional Is Reset Based On Prevailing Spot Exchange Rate.

File: example25.xml

On 9 January, 2001 PartyA and PartyB agree to enter into an FX Reseting interest rate swap. The terms of the contract are:

- Effective Date of the Swap: 11 January 2006.
- Termination Date of the Underlying Swap: 11 January, 2011
- PartyB makes semi-annual fixed rate payments based on a fixed rate of 1.0%, on an ACT/360-Fixed basis.
- Notional on the fixed leg of the Swap: JPY 100,000,000
- PartyA makes quarterly floating rate payments based on the floating rate index `USD-LIBOR-BBA`, on an ACT/360 basis.
- Notional on the floating leg of the swap has a Ccy of USD and is FX Linked to the fixed leg JPY notional. The conversion rate for each cashflow is that observed on payment day at 17:00 hours from the Bank of Japan information source.

10.27 Example 26 – Example 25 – Fixed/Floating IRS Where The Floating Stream Notional Is Reset Based On Prevailing Spot Exchange Rate - Cashflows.

File: example26.xml

On 9 January, 2001 PartyA and PartyB agree to enter into a forward starting FX Reseting interest rate swap. The terms of the contract are:

- Effective Date of the Swap: 11 January, 2006.
- Termination Date of the Underlying Swap: 11 January, 201
- PartyB makes semi-annual fixed rate payments based on a fixed rate of 1.0%, on an ACT/360-Fixed basis.
- Notional on the fixed leg of the Swap: JPY 100,000,000
- PartyA makes quarterly floating rate payments based on the floating rate index `USD-LIBOR-BBA`, on an ACT/360 basis.
- Notional on the floating leg of the swap has a Ccy of USD and is FX Linked to the fixed leg JPY notional. The conversion rate for each cashflow is that observed on payment day at 17:00 hours from the Bank of Japan information source.

Things to note:

- The Swap stream is defined with cashflows

10.28 Example 27 – Inverse Floater

File: example27.xml

On 30 August, 2000 PartyA and PartyB agree to enter into an ISDA. The terms of the contract are:

- Effective Date of the Swap: 30 August 2001.
- Termination Date of the Underlying Swap: 30 August, 2006
- Notional on the Underlying Swap Amount: USD 100,000,000
- PartyA makes quarterly payments with floating rate payments derived as (8.5% - floating rate index EUR-EURIBOR-Telerate), on an ACT/360 basis.
- PartyB makes semi-annual fixed rate payments based on a fixed rate of 4.5%, on an 30/360 basis.

Things to note:

- The use of the floatingRateMultiplierSchedule to invert the floating USD rate.

10.29 Example 28 - BulletPayments

File: example28.xml

On 29 April, 2000 PartyA agrees the payment of a single cashlow to PartyB. The terms of the contract are:

- The payment has an unadjusted payment date of 27 July 2001.
- The amount to be paid is USD 15,000.
- Payment dates are adjusted to London and NY business centers for both payments

APPENDIX

This appendix documents changes introduced between the FpML 2.0 Working Draft, 23 August 2001 (http://www.fpml.org/spec/2001/wd-fpml-2-0-2001-08-23) and FpML 2.0 Working Draft, 17 October 2001 (http://www.fpml.org/spec/2001/wd-fpml-2-0-2001-10-17).

1. All products changed to extend FpML_Product

DTD Changes

- FpML_Product changed to only contain an optional productType
  <!ENTITY % FpML_Product "productType?">
- All products (bulletPayment, capFloor , fra, strategy, swap and swaption) changed to extend FpML_Product
- Added a FpML_BulletPayment entity:
  <!ENTITY %FpML_BulletPayment "%FpML_Product;,%FpML_Payment;">
- Changed bulletPayment to be of type FpML_BulletPayment
- Removed StrategyTypeSchemeDefault from the FpML element.
- Removed strategyType from FpML_Strategy (this is covered by productType) :
  <!ENTITY % FpML_Strategy "%FpML_Product;, (%FpML_ProductList;)+">

Document Changes

- **Section 5.3 FpML_Product** updated for new defintion
- **Section 5.3 FpML_BulletPayment** entity definition added
- **Section 5.3 FpML_CapFloor** updated definition to include optional productType element.
- **Section 5.3 FpML_Fra** updated definition to include optional productType element.
- **Section 5.3 FpML_Strategy** updated for new definition.
- **Section 5.3 FpML_Swap** updated definition to include optional productType element.
- **Section 5.3 FpML_Swaption** updated definition to include optional productType element.
- **Section 7.1 bulletPayment** changed type to BulletPayment

2. Removal of the product element

DTD Changes

- Added new entity FpML_ProductList to represent the full set of FpML products:
  <!ENTITY % FpML_ProductList "(bulletPayment | capFloor | fra | swap | swaption | strategy)">
- Changed FpML_Trade to use FpML_ProductList rather than the element product:
  <!ENTITY % FpML_Trade "tradeHeader , %FpML_ProductList; , party+ , otherPartyPayment">*
- Changed FpML_Strategy to contain multiple FpML_ProductLists rather than multiple product elements:
  <!ENTITY % FpML_Strategy "%FpML_Product;, (%FpML_ProductList;)+">
- Removed the product element which is now redundant.

Document Changes

- **Section 5.3 FpML_ProductList** entity definition added.
- **Section 5.3 FpML_Trade** definition updated.
- **Section 7.1 product** definition removed.

3. Change FpML_Fee to extend FpML_Payment

DTD Changes

- Added a new entity FpML_Payment:
  <!ENTITY % FpML_Payment "payerPartyReference , receiverPartyReference , paymentAmount , paymentDate? , adjustedPaymentDate?">
- Changed FpML_Fee to extend FpML_Payment:
  <!ENTITY % FpML_Fee "%FpML_Payment; , paymentType?">
- Change element premium to be of type FpML_Payment:
  <!ELEMENT premium (%FpML_Payment;)>
  <!ATTLIST premium type NMTOKEN #FIXED 'Payment'
  id ID #IMPLIED >

Document Changes

- **Section 5.3 FpML_Payment** entity definition added.
- **Section 5.3 FpML_Fee** entity definition updated as now extends FpML_Payment
- **Section 5.3 FpML_Swaption** changed element premium to be of type FpML_Payment
- **Section 7.1 premium** changed premium definition to be of type FpML_Payment

4. Incorporate errata list

1. **feeAmount and feeRate (Section 5.3 FpML_ExerciseFee and Section 7.1)** changed from type 'double' to type decimal
2. **Section 5.3 FpML_MandatoryEarlyTerminationProvision** diagram updated.
3. **Section 5.3 FpML_OptionalEarlyTerminationProvision** diagram updated.
4. **Section 5.3 FpML_Trade** diagram updated.
5. **Section 5.3 FpML_CancelableProvisionAdjustedDates** spelling of cancelable corrected.
6. **Sample XML 9-26** '#' added to hrefs.
7. **Sample XML Example 28** changed to only have one bulletPayment
8. **expirationTime** description updated.

5. Updated Sample XML for changes

All the sample XML have been updated to be valid per the new DTD. This involved the following:

1. Remove product elements
2. Move position of paymentType element wherever FpML_Fee has been used.

6. Other Changes to the text

1. **Section 3.3.1 The Trade Component**
   - Updated trade diagram.
   - Update the product bullet point to reflect the removal of the product element
   - Updated otherPartyPayment diagram. paymentType is now the last element.
2. **Section 9.2 Coding Schemes**
   - Removed Strategy Type from the list of excluded schemes on page 246
3. **paymentDaysOffset** words of clarification added.
4. **Product Architecture Overview** has been re-organised. It has been split into Product Architecture and Interest Rate Derivative Architecture.
5. **Section 3.2 DTD Structure** has been added.