

# XML Schema Documentation

## Table of Contents

- [Schema Document Properties](#)
- [Global Declarations](#)
  - [Element: `dataDocument`](#)
- [Legend](#)
- [Glossary](#)

[top](#)

## Schema Document Properties

<b>Target Namespace</b>	<a href="http://www.fpml.org/FpML-5/confirmation">http://www.fpml.org/FpML-5/confirmation</a>
<b>Version</b>	\$Revision: 2686 \$
<b>Element and Attribute Namespaces</b>	<ul style="list-style-type: none"> <li>• Global element and attribute declarations belong to this schema's target namespace.</li> <li>• By default, local element declarations belong to this schema's target namespace.</li> <li>• By default, local attribute declarations have no namespace.</li> </ul>
<b>Schema Composition</b>	<ul style="list-style-type: none"> <li>• This schema includes components from the following schema document(s):           <ul style="list-style-type: none"> <li>◦ <a href="#">fpml-ird-5-0.xsd</a></li> <li>◦ <a href="#">fpml-eqd-5-0.xsd</a></li> <li>◦ <a href="#">fpml-return-swaps-5-0.xsd</a></li> <li>◦ <a href="#">fpml-cd-5-0.xsd</a></li> <li>◦ <a href="#">fpml-bond-option-5-0.xsd</a></li> <li>◦ <a href="#">fpml-correlation-swaps-5-0.xsd</a></li> <li>◦ <a href="#">fpml-dividend-swaps-5-0.xsd</a></li> <li>◦ <a href="#">fpml-variance-swaps-5-0.xsd</a></li> <li>◦ <a href="#">fpml-com-5-0.xsd</a></li> <li>◦ <a href="#">fpml-confirmation-processes-5-0.xsd</a></li> </ul> </li> </ul>
<b>Documentation</b>	<p>products</p> <p>business process messaging</p> <p>reporting and settlement</p>

## Declared Namespaces

Prefix	Namespace
Default namespace	<a href="http://www.fpml.org/FpML-5/confirmation">http://www.fpml.org/FpML-5/confirmation</a>
xml	http://www.w3.org/XML/1998/namespace
dsig	http://www.w3.org/2000/09/xmldsig#
fpml-annotation	http://www.fpml.org/annotation
xsd	http://www.w3.org/2001/XMLSchema
ecore	http://www.eclipse.org/emf/2002/Ecore

## Schema Component Representation

```
<xsd:schema targetNamespace="http://www.fpml.org/FpML-5/confirmation"
```

```

documentRoot="FpML" nsPrefix="conf" package="org.fpml.confirmation" version="$Revision: 2686
$" elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xsd:include schemaLocation="fpml-ird-5-0.xsd"/>
  <xsd:include schemaLocation="fpml-eqd-5-0.xsd"/>
  <xsd:include schemaLocation="fpml-return-swaps-5-0.xsd"/>
  <xsd:include schemaLocation="fpml-cd-5-0.xsd"/>
  <xsd:include schemaLocation="fpml-bond-option-5-0.xsd"/>
  <xsd:include schemaLocation="fpml-correlation-swaps-5-0.xsd"/>
  <xsd:include schemaLocation="fpml-dividend-swaps-5-0.xsd"/>
  <xsd:include schemaLocation="fpml-variance-swaps-5-0.xsd"/>
  <xsd:include schemaLocation="fpml-com-5-0.xsd"/>
  <xsd:include schemaLocation="fpml-confirmation-processes-5-0.xsd"/>
  ...
</xsd:schema>

```

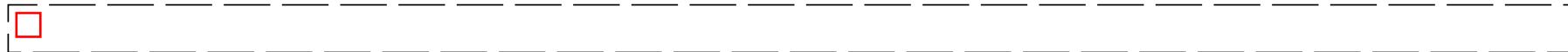
[top](#)

## Global Declarations

### Element: **dataDocument**

<b>Name</b>	dataDocument
<b>Type</b>	<a href="#">DataDocument</a>
<b>Nillable</b>	no
<b>Abstract</b>	no
<b>Documentation</b>	A document containing trade and/or portfolio and/or party data without expressing any processing intention.

#### Logical Diagram



#### XML Instance Representation

```

<dataDocument
  fpmlVersion=" xsd:token (value comes from list: {'5-0'}) [1]
  'Indicate which version of the FpML Schema an FpML message adheres to.'
  "
  expectedBuild=" xsd:positiveInteger [0..1]
  'This optional attribute can be supplied by a message creator in an FpML instance to
  specify which build number of the schema was used to define the message when it was generated.'
  "
  actualBuild="8 [0..1]
  'The specific build number of this schema version. This attribute is not included in
  an instance document. Instead, it is supplied by the XML parser when the document is
  validated against the FpML schema and indicates the build number of the schema file. Every
  time FpML publishes a change to the schema, validation rules, or examples within a version
  (e.g., version 4.2) the actual build number is incremented. If no changes have been
  made between releases within a version (i.e. from Trial Recommendation to Recommendation)
  the actual build number stays the same.'

```

```

    ">
      <validation> Validation </validation> [0..*]
    Start Choice [1]
      <trade> Trade </trade> [0..*]
      'The root element in an FpML trade document.'

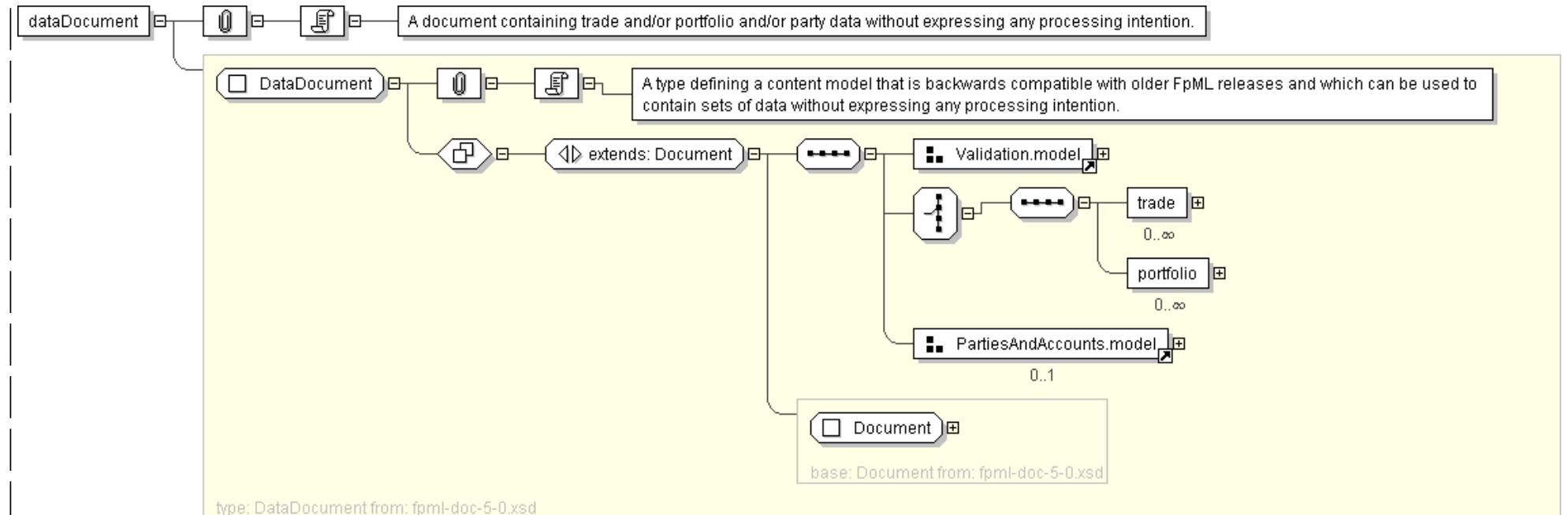
      <portfolio> Portfolio </portfolio> [0..*]
      'An arbitrary grouping of trade references (and possibly other portfolios).'

    End Choice
  Start Group: PartiesAndAccounts.model [0..1]
    <party> Party </party> [1..*]
    'A legal entity or a subdivision of a legal entity.', 'Parties can perform multiple roles in
    a trade lifecycle. For example, the principal parties obligated to make payments from time
    to time during the term of the trade, but may include other parties involved in, or
    incidental to, the trade, such as parties acting in the role of novation transferor/
    transferee, broker, calculation agent, etc. In FpML roles are defined in multiple places
    within a document.'

    <account> Account </account> [0..*]
    'Optional account information used to precisely define the origination and destination
    of financial instruments.'

  End Group: PartiesAndAccounts.model
</dataDocument>

```

**Diagram****Schema Component Representation**

```
<xsd:element name="dataDocument" type=" DataDocument " />
```

[top](#)

## Legend

### Complex Type:

Schema Component Type

### AusAddress

Schema Component Name

**Super-types:** [Address](#) < AusAddress (by extension)

**Sub-types:** • [QLDAddress](#) (by restriction)

If this schema component is a type definition, its type hierarchy is shown in a gray-bordered box.

Name	AusAddress
Abstract	no

The table above displays the properties of this schema component.

### XML Instance Representation

```
<.... country="Australia" >
<unitNo> string </unitNo> [0..1]
<houseNo> string </houseNo> [1]
<street> string </street> [1]
Start Choice [1]
<city> string </city> [1]
<town> string </town> [1]
End Choice
<state> AusStates </state> [1]
<postcode> string <><pattern = [1-9][0-9]{3}></postcode> [1]
</...>
```

The XML Instance Representation table above shows the schema component's content as an XML instance.

- The minimum and maximum occurrence of elements and attributes are provided in square brackets, e.g. [0..1].
- Model group information are shown in gray, e.g. Start Choice ... End Choice.
- For type derivations, the elements and attributes that have been added to or changed from the base type's content are shown in **bold**.
- If an element/attribute has a fixed value, the fixed value is shown in green, e.g. country="Australia".
- Otherwise, the type of the element/attribute is displayed.
  - If the element/attribute's type is in the schema, a link is provided to it.
  - For local simple type definitions, the constraints are displayed in angle brackets, e.g. <><pattern = [1-9][0-9]{3}></>.

### Schema Component Representation

```
<complexType name="AusAddress">
<complexContent>
<extension base=" Address ">
<sequence>
<element name="state" type=" AusStates " />
```

```

<element name="postcode">
  <simpleType>
    <restriction base=" string ">
      <pattern value="[1-9][0-9]{3}" />
    </restriction>
  </simpleType>
</element>
</sequence>
<attribute name="country" type=" string " fixed="Australia" />
</extension>
</complexContent>
</complexType>

```

The Schema Component Representation table above displays the underlying XML representation of the schema component. (Annotations are not shown.)

[top](#)

## Glossary

**Abstract** (Applies to complex type definitions and element declarations). An abstract element or complex type cannot be used to validate an element instance. If there is a reference to an abstract element, only element declarations that can substitute the abstract element can be used to validate the instance. For references to abstract type definitions, only derived types can be used.

**All Model Group** Child elements can be provided *in any order* in instances. See: <http://www.w3.org/TR/xmlschema-1/#element-all>.

**Choice Model Group** *Only one* from the list of child elements and model groups can be provided in instances. See: <http://www.w3.org/TR/xmlschema-1/#element-choice>.

**Collapse Whitespace Policy** Replace tab, line feed, and carriage return characters with space character (Unicode character 32). Then, collapse contiguous sequences of space characters into single space character, and remove leading and trailing space characters.

**Disallowed Substitutions** (Applies to element declarations). If *substitution* is specified, then *substitution group* members cannot be used in place of the given element declaration to validate element instances. If *derivation methods*, e.g. extension, restriction, are specified, then the given element declaration will not validate element instances that have types derived from the element declaration's type using the specified derivation methods. Normally, element instances can override their declaration's type by specifying an *xsi:type* attribute.

**Key Constraint** Like [Uniqueness Constraint](#), but additionally requires that the specified value(s) must be provided. See: [http://www.w3.org/TR/xmlschema-1/#clidentity-constraint\\_Definitions](http://www.w3.org/TR/xmlschema-1/#clidentity-constraint_Definitions).

**Key Reference Constraint** Ensures that the specified value(s) must match value(s) from a [Key Constraint](#) or [Uniqueness Constraint](#). See: [http://www.w3.org/TR/xmlschema-1/#clidentity-constraint\\_Definitions](http://www.w3.org/TR/xmlschema-1/#clidentity-constraint_Definitions).

**Model Group** Groups together element content, specifying the order in which the element content can occur and the number of times the group of element content may be repeated. See: [http://www.w3.org/TR/xmlschema-1/#Model\\_Groups](http://www.w3.org/TR/xmlschema-1/#Model_Groups).

**Nillable** (Applies to element declarations). If an element declaration is nillable, instances can use the *xsi:nil* attribute. The *xsi:nil* attribute is the boolean attribute, *nil*, from the <http://www.w3.org/2001/XMLSchema-instance> namespace. If an element instance has an *xsi:nil* attribute set to true, it can be left empty, even though its element declaration may have required content.

**Notation** A notation is used to identify the format of a piece of data. Values of elements and attributes that are of type, NOTATION, must come from the names of declared notations. See: [http://www.w3.org/TR/xmlschema-1/#cNotation\\_Declarations](http://www.w3.org/TR/xmlschema-1/#cNotation_Declarations).

**Preserve Whitespace Policy** Preserve whitespaces exactly as they appear in instances.

**Prohibited Derivations** (Applies to type definitions). Derivation methods that cannot be used to create sub-types from a given type definition.

**Prohibited Substitutions** (Applies to complex type definitions). Prevents sub-types that have been derived using the specified derivation methods from validating element instances in place of the given type definition.

**Replace Whitespace Policy** Replace tab, line feed, and carriage return characters with space character (Unicode character 32).

**Sequence Model Group** Child elements and model groups must be provided *in the specified order* in instances. See: <http://www.w3.org/TR/xmlschema-1/#element-sequence>.

**Substitution Group** Elements that are *members* of a substitution group can be used wherever the *head* element of the substitution group is referenced.

**Substitution Group Exclusions** (Applies to element declarations). Prohibits element declarations from nominating themselves as being able to substitute a given element declaration, if they have types that are derived from the original element's type using the specified derivation methods.

**Target Namespace** The target namespace identifies the namespace that components in this schema belongs to. If no target namespace is provided, then the schema components do not belong to any namespace.

**Uniqueness Constraint** Ensures uniqueness of an element/attribute value, or a combination of values, within a specified scope. See: [http://www.w3.org/TR/xmlschema-1/#clidentity-constraint\\_Definitions](http://www.w3.org/TR/xmlschema-1/#clidentity-constraint_Definitions).

[top](#)

---

Generated by [oXygen XML Editor](#) using a modified version of [xs3p](#) that adds schema diagrams and chunking support.

# XML Schema Documentation

## Table of Contents

- [Schema Document Properties](#)
- [Global Declarations](#)
  - [Element: basket](#)
  - [Element: bond](#)
  - [Element: cash](#)
  - [Element: commodity](#)
  - [Element: convertibleBond](#)
  - [Element: curveInstrument](#)
  - [Element: deposit](#)
  - [Element: equity](#)
  - [Element: exchangeTradedFund](#)
  - [Element: future](#)
  - [Element: fx](#)
  - [Element: index](#)
  - [Element: loan](#)
  - [Element: mortgage](#)
  - [Element: mutualFund](#)
  - [Element: rateIndex](#)
  - [Element: simpleCreditDefaultSwap](#)
  - [Element: simpleFra](#)
  - [Element: simpleIrsSwap](#)
  - [Element: underlyingAsset](#)
- [Global Definitions](#)
  - [Complex Type: ActualPrice](#)
  - [Complex Type: AnyAssetReference](#)
  - [Complex Type: Asset](#)
  - [Complex Type: AssetMeasureType](#)
  - [Complex Type: AssetPool](#)
  - [Complex Type: AssetReference](#)
  - [Complex Type: BasicQuotation](#)
  - [Complex Type: Basket](#)
  - [Complex Type: BasketConstituent](#)
  - [Complex Type: BasketId](#)
  - [Complex Type: BasketName](#)
  - [Complex Type: Bond](#)
  - [Complex Type: Cash](#)
  - [Complex Type: Commission](#)
  - [Complex Type: Commodity](#)
  - [Complex Type: CommodityBase](#)
  - [Complex Type: CommodityBusinessCalendar](#)
  - [Complex Type: CommodityBusinessCalendarTime](#)
  - [Complex Type: CommodityDetails](#)
  - [Complex Type: ConstituentWeight](#)
  - [Complex Type: ConvertibleBond](#)
  - [Complex Type: CouponType](#)
  - [Complex Type: CurveInstrument](#)
  - [Complex Type: Deposit](#)
  - [Complex Type: DividendPayout](#)
  - [Complex Type: EquityAsset](#)
  - [Complex Type: ExchangeTraded](#)
  - [Complex Type: ExchangeTradedCalculatedPrice](#)
  - [Complex Type: ExchangeTradedContract](#)
  - [Complex Type: ExchangeTradedFund](#)
  - [Complex Type: FacilityType](#)
  - [Complex Type: Future](#)
  - [Complex Type: FutureId](#)
  - [Complex Type: FxConversion](#)
  - [Complex Type: FxRateAsset](#)
  - [Complex Type: IdentifiedAsset](#)
  - [Complex Type: Index](#)
  - [Complex Type: Lien](#)

- [Complex Type: Loan](#)
- [Complex Type: Mortgage](#)
- [Complex Type: MortgageSector](#)
- [Complex Type: MutualFund](#)
- [Complex Type: PendingPayment](#)
- [Complex Type: Price](#)
- [Complex Type: PriceQuoteUnits](#)
- [Complex Type: QuantityUnit](#)
- [Complex Type: QuotationCharacteristics](#)
- [Complex Type: QuoteTiming](#)
- [Complex Type: RateIndex](#)
- [Complex Type: ReportingCurrencyType](#)
- [Complex Type: SimpleCreditDefaultSwap](#)
- [Complex Type: SimpleFra](#)
- [Complex Type: SimpleRSwap](#)
- [Complex Type: SingleUnderlyer](#)
- [Complex Type: TimeZone](#)
- [Complex Type: Underlyer](#)
- [Complex Type: UnderlyingAsset](#)
- [Complex Type: UnderlyingAssetTranche](#)
- [Model Group: BasketIdentifier.model](#)
- [Model Group: BondCalculation.model](#)
- [Model Group: BondChoice.model](#)
- [Model Group: CommodityProduct.model](#)
- [Model Group: CommodityReferencePriceFramework.model](#)
- [Model Group: CreditEntity.model](#)
- [Model Group: EquityPrice.model](#)
- [Model Group: ExchangeIdentifier.model](#)
- [Model Group: FixedIncomeSecurityContent.model](#)
- [Model Group: Quotation.model](#)
- [Model Group: QuotationCharacteristics.model](#)
- [Model Group: QuoteLocation.model](#)

- [Legend](#)

- [Glossary](#)

[top](#)

## Schema Document Properties

<b>Target Namespace</b>	<a href="http://www.fpml.org/FpML-5/confirmation">http://www.fpml.org/FpML-5/confirmation</a>
<b>Version</b>	\$Revision: 2587 \$
<b>Element and Attribute Namespaces</b>	<ul style="list-style-type: none"> <li>• Global element and attribute declarations belong to this schema's target namespace.</li> <li>• By default, local element declarations belong to this schema's target namespace.</li> <li>• By default, local attribute declarations have no namespace.</li> </ul>
<b>Schema Composition</b>	<ul style="list-style-type: none"> <li>• This schema includes components from the following schema document(s):           <ul style="list-style-type: none"> <li>◦ <a href="#">fpml-shared-5-0.xsd</a></li> </ul> </li> </ul>

## Declared Namespaces

Prefix	Namespace
Default namespace	<a href="http://www.fpml.org/FpML-5/confirmation">http://www.fpml.org/FpML-5/confirmation</a>
xml	<a href="http://www.w3.org/XML/1998/namespace">http://www.w3.org/XML/1998/namespace</a>
fpml-annotation	<a href="http://www.fpml.org/annotation">http://www.fpml.org/annotation</a>
xsd	<a href="http://www.w3.org/2001/XMLSchema">http://www.w3.org/2001/XMLSchema</a>
ecore	<a href="http://www.eclipse.org/emf/2002/Ecore">http://www.eclipse.org/emf/2002/Ecore</a>

## Schema Component Representation

```

<xsd:schema targetNamespace="http://www.fpml.org/FpML-5/confirmation"
  documentRoot="FpML" nsPrefix="conf" package="org.fpml.confirmation" version="$Revision: 2587 "
  $" elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xsd:include schemaLocation="fpml-shared-5-0.xsd" />
  ...
</xsd:schema>

```

[top](#)

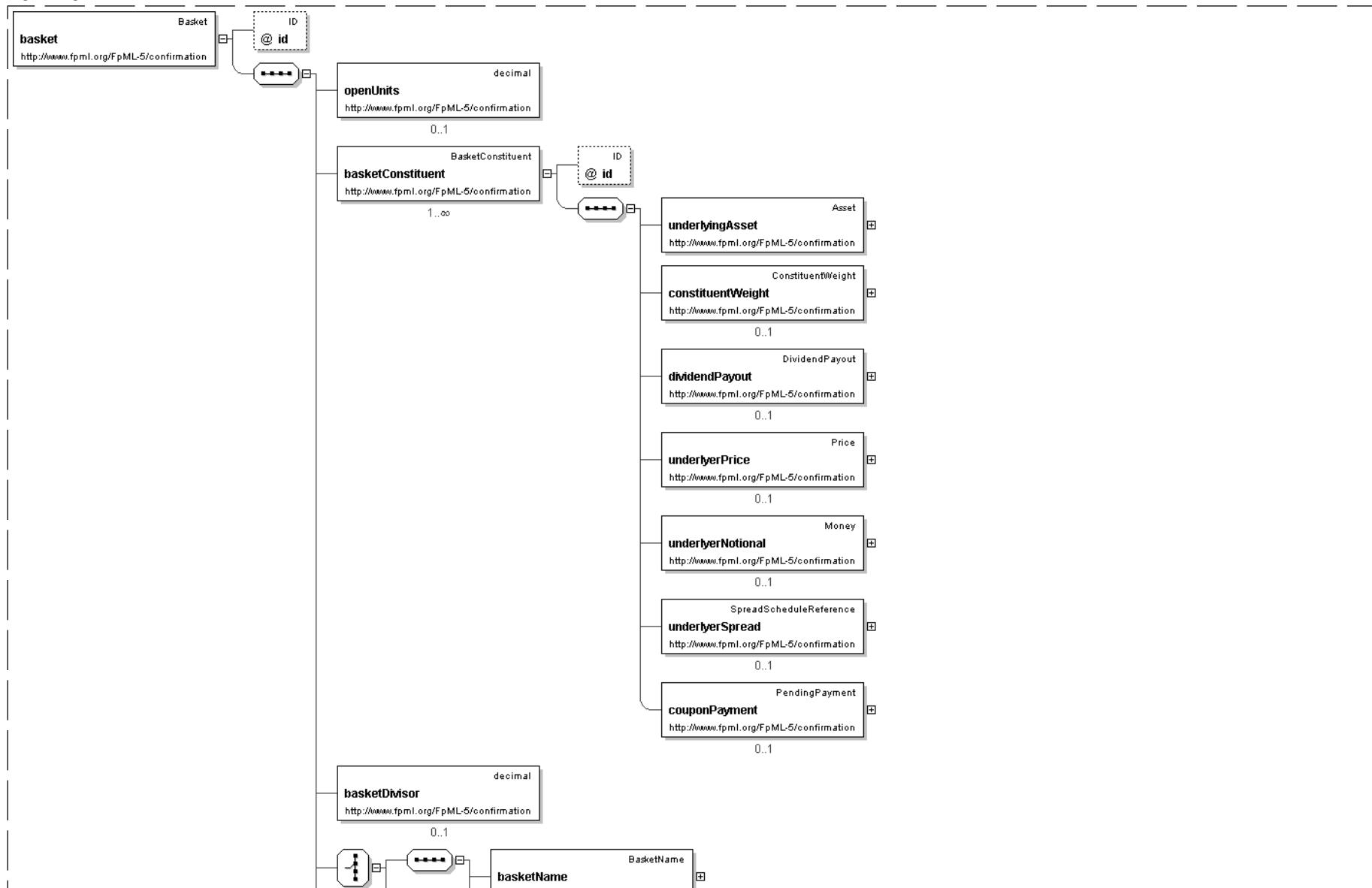
## Global Declarations

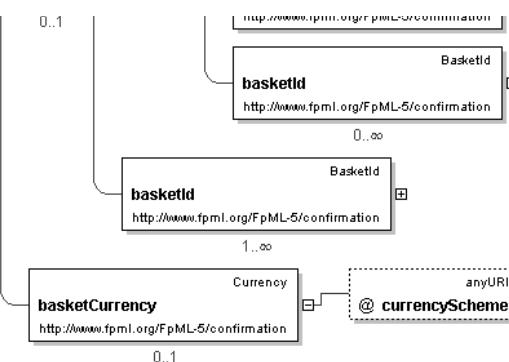
### Element: basket

- This element can be used wherever the following element is referenced:
  - [underlyingAsset](#)

Name	basket
Type	<a href="#">Basket</a>
Nillable	no
Abstract	no
Documentation	Defines the underlying asset when it is a basket.

#### Logical Diagram



**XML Instance Representation**

```

<basket
  id=" xsd:ID [0..1]">
  <openUnits> xsd:decimal </openUnits> [0..1]
  'The number of units (index or securities) that constitute the underlyer of the swap. In
  the case of a basket swap, this element is used to reference both the number of basket
  units, and the number of each asset components of the basket when these are expressed
  in absolute terms.'

  <basketConstituent> BasketConstituent </basketConstituent> [1..*]
  'Describes each of the components of the basket.'

  <basketDivisor> xsd:decimal </basketDivisor> [0..1]
  'Specifies the basket divisor amount. This value is normally used to adjust the
  constituent weight for pricing or to adjust for dividends, or other corporate actions.'

Start Group: BasketIdentifier.model [0..1]
'Reuses the group that specifies a name and an identifier for a given basket.'

Start Choice [1]
  <basketName> BasketName </basketName> [1]
  'The name of the basket expressed as a free format string. FpML does not define usage rules
  for this element.'

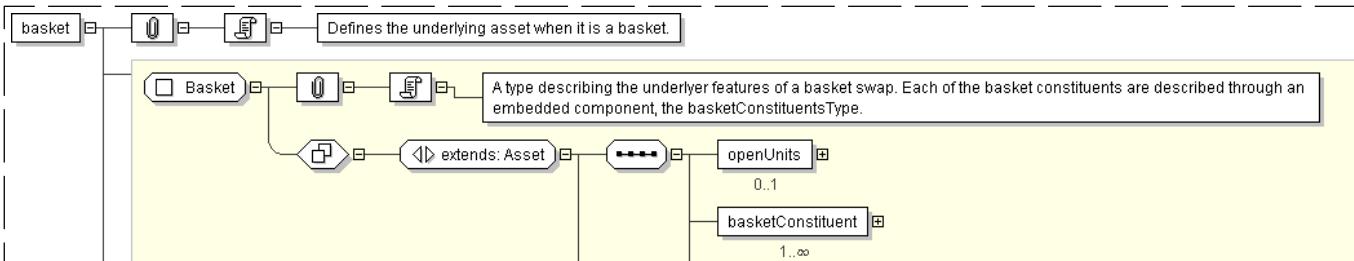
  <basketId> BasketId </basketId> [0..*]
  'A CDS basket identifier'

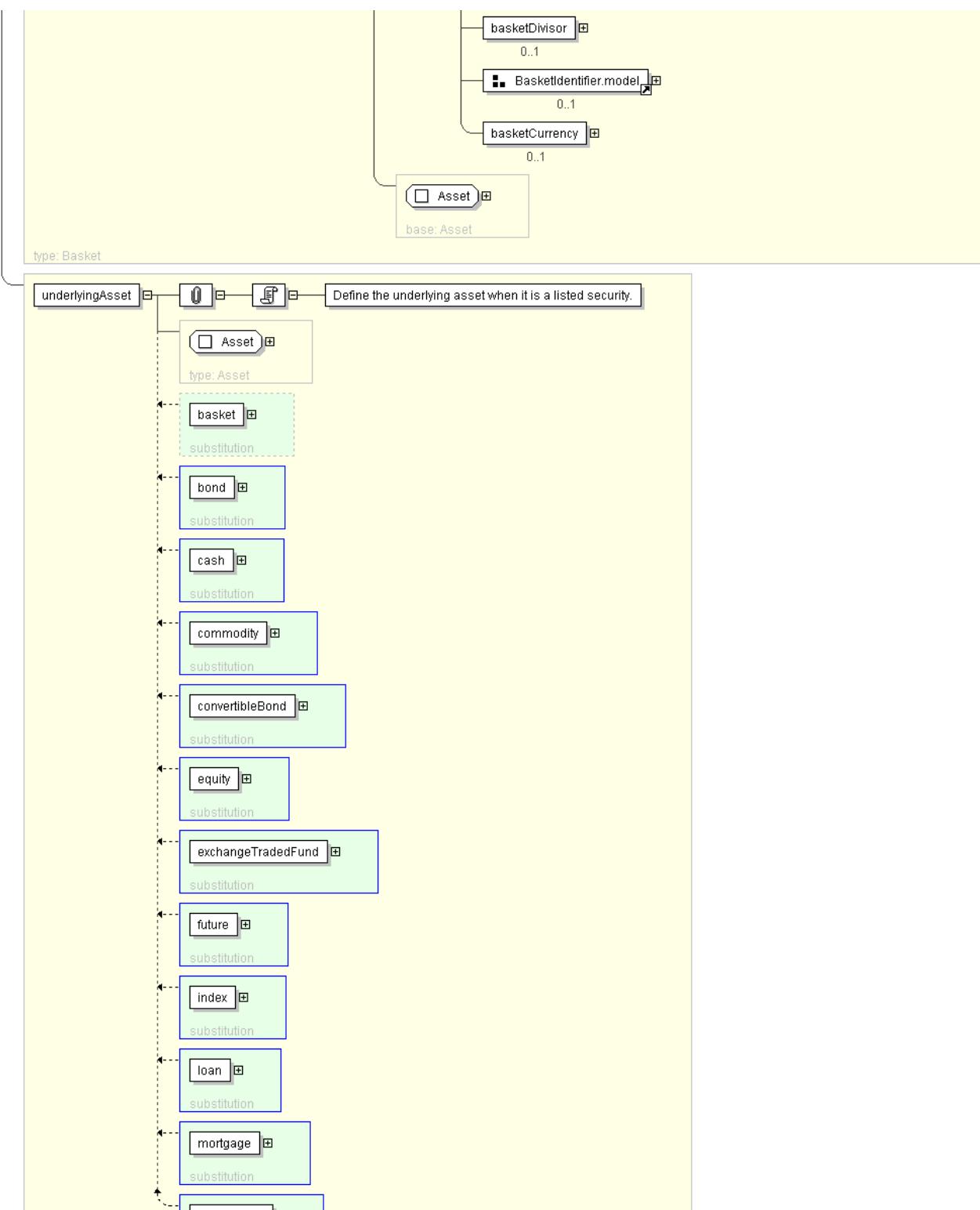
  <basketId> BasketId </basketId> [1..*]
  'A CDS basket identifier'

End Choice
End Group: BasketIdentifier.model
  <basketCurrency> Currency </basketCurrency> [0..1]
  'Specifies the currency for this basket.'

</basket>

```

**Diagram**



**Schema Component Representation**

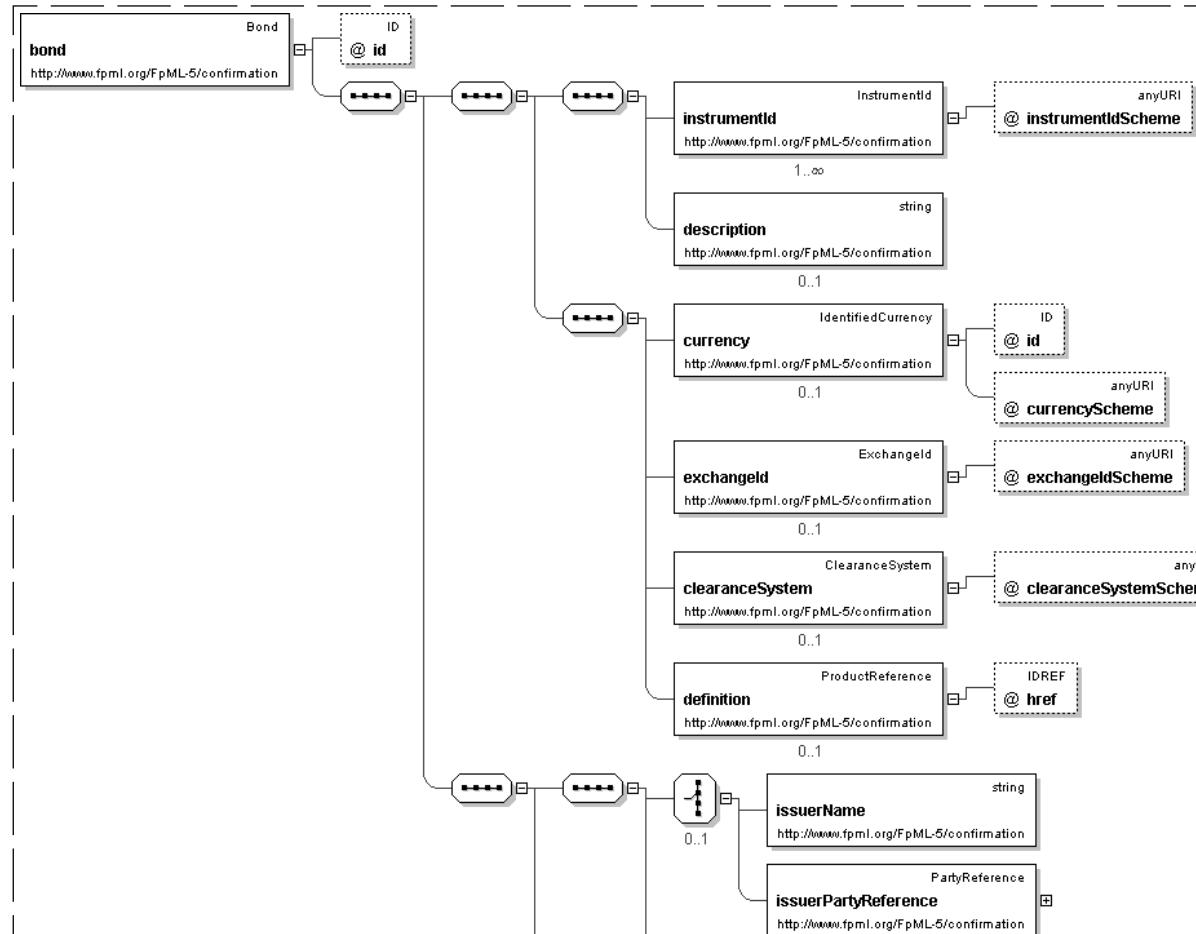
```
<xsd:element name="basket" type="Basket" substitutionGroup="underlyingAsset"/>
```

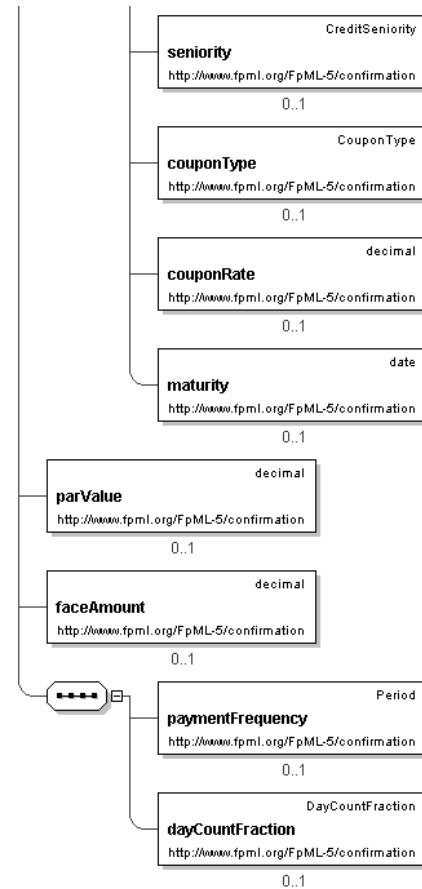
top

**Element: bond**

- This element can be used wherever the following element is referenced:
  - underlyingAsset

Name	bond
Used by (from the same schema document)	Model Group <a href="#">BondChoice.model</a>
Type	Bond
Nullable	no
Abstract	no
Documentation	Identifies the underlying asset when it is a series or a class of bonds.

**Logical Diagram**

**XML Instance Representation**

```

<bond
  id=" xsd:ID [0..1]">
  <instrumentId> InstrumentId </instrumentId> [1..*]
  'Identification of the underlying asset, using public and/or private identifiers.'

  <description> xsd:string </description> [0..1]
  'Long name of the underlying asset.'

  <currency> IdentifiedCurrency </currency> [0..1]
  'Trading currency of the underlyer when transacted as a cash instrument.'

  <exchangeId> ExchangeId </exchangeId> [0..1]
  'Identification of the exchange on which this asset is transacted for the purposes
  of calculating a contractual payoff. The term \"Exchange\" is assumed to have the meaning
  as defined in the ISDA 2002 Equity Derivatives Definitions.'

  <clearanceSystem> ClearanceSystem </clearanceSystem> [0..1]
  'Identification of the clearance system associated with the transaction exchange.'

  <definition> ProductReference </definition> [0..1]
  'An optional reference to a full FpML product that defines the simple product in
  greater detail. In case of inconsistency between the terms of the simple product and those
  of the detailed definition, the values in the simple product override those in the
  detailed definition.'

```

Start Choice [0..1]

'Specifies the issuer name of a fixed income security or convertible bond. This name can either be explicitly stated, or specified as an href into another element of the document, such as the obligor.'

```
<issuerName> xsd:string </issuerName> [1]
<issuerPartyReference> PartyReference </issuerPartyReference> [1]
```

End Choice

&lt;seniority&gt; CreditSeniority &lt;/seniority&gt; [0..1]

'The repayment precedence of a debt instrument.'

&lt;couponType&gt; CouponType &lt;/couponType&gt; [0..1]

'Specifies if the bond has a variable coupon, step-up/down coupon or a zero-coupon.'

&lt;couponRate&gt; xsd:decimal &lt;/couponRate&gt; [0..1]

'Specifies the coupon rate (expressed in percentage) of a fixed income security or convertible bond.'

&lt;maturity&gt; xsd:date &lt;/maturity&gt; [0..1]

'The date when the principal amount of a security becomes due and payable.'

&lt;parValue&gt; xsd:decimal &lt;/parValue&gt; [0..1]

'Specifies the nominal amount of a fixed income security or convertible bond.'

&lt;faceAmount&gt; xsd:decimal &lt;/faceAmount&gt; [0..1]

'Specifies the total amount of the issue. Corresponds to the par value multiplied by the number of issued security.'

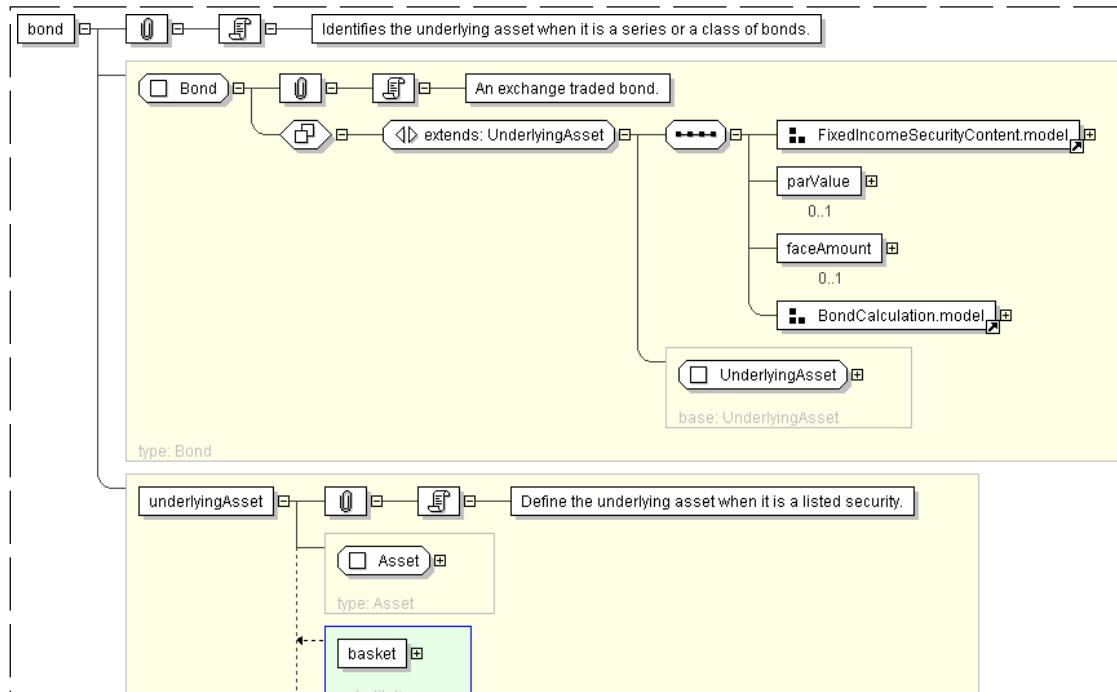
&lt;paymentFrequency&gt; Period &lt;/paymentFrequency&gt; [0..1]

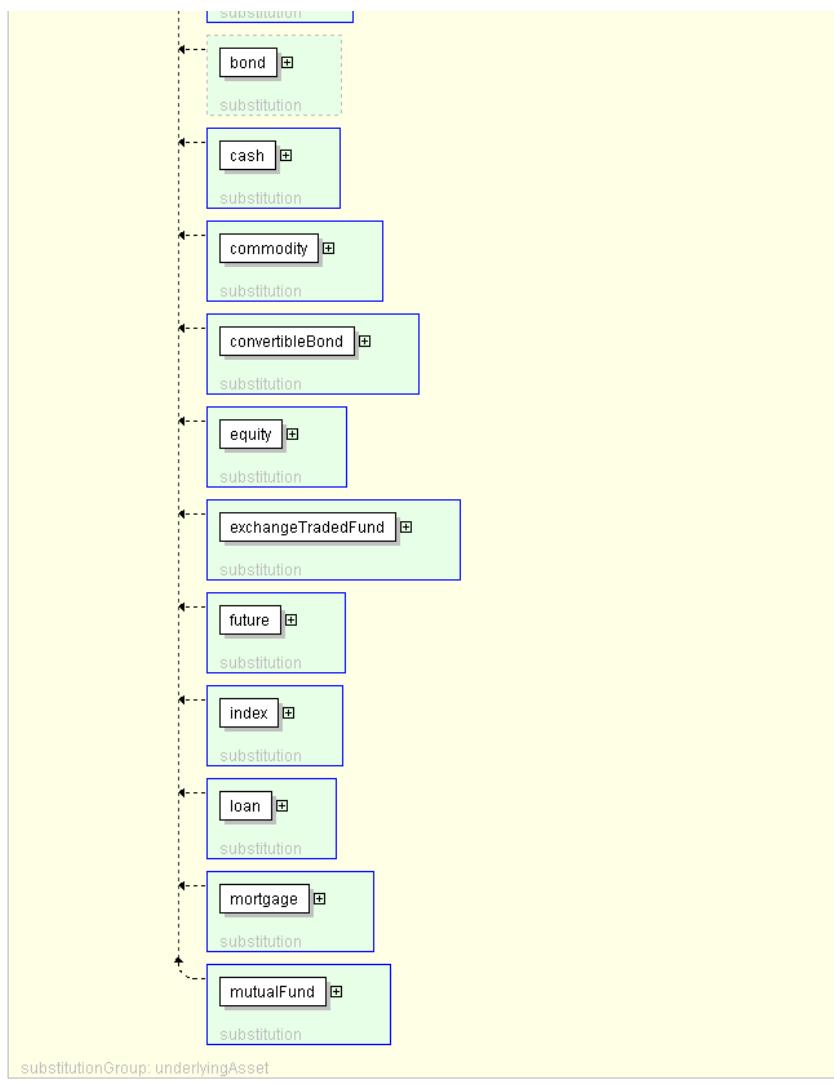
'Specifies the frequency at which the bond pays, e.g. 6M.'

&lt;dayCountFraction&gt; DayCountFraction &lt;/dayCountFraction&gt; [0..1]

'The day count basis for the bond.'

&lt;/bond&gt;

**Diagram**

**Schema Component Representation**

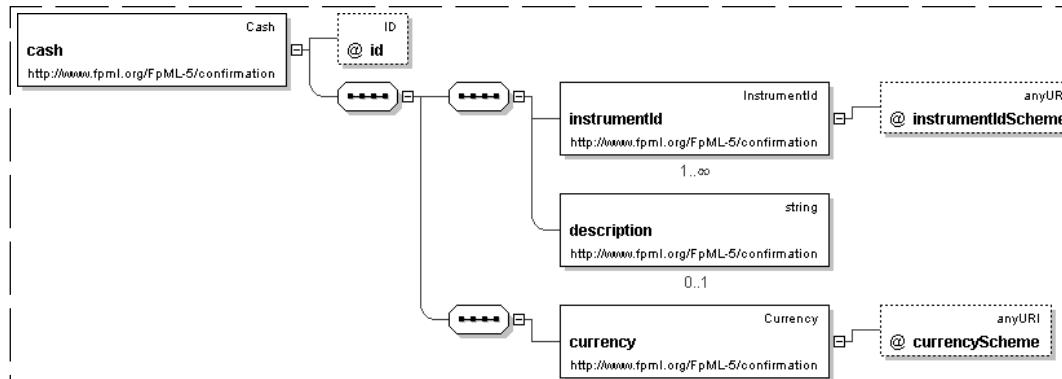
```
<xsd:element name="bond" type="#Bond" substitutionGroup="underlyingAsset" />
```

top

**Element: cash**

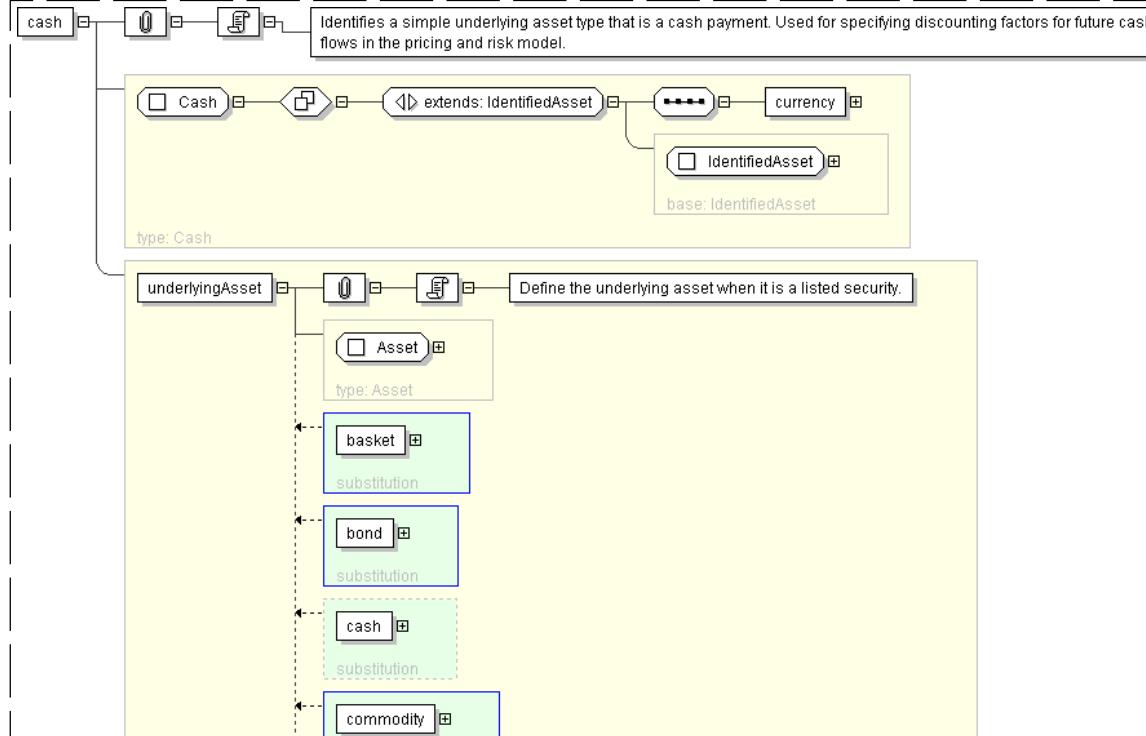
- This element can be used wherever the following element is referenced:
  - underlyingAsset

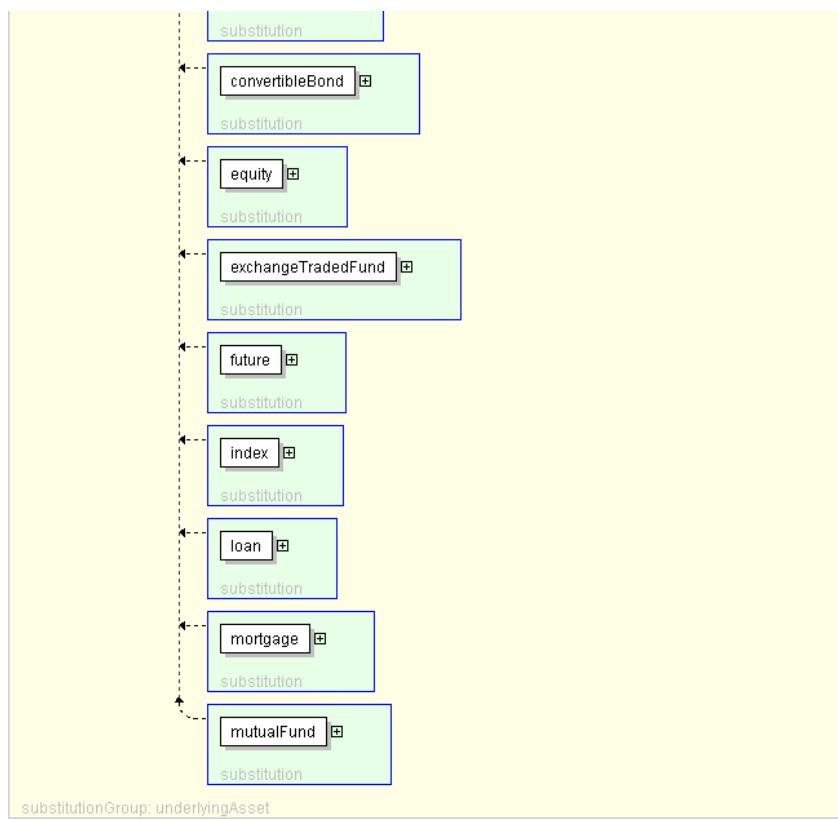
<b>Name</b>	cash
<b>Type</b>	Cash
<b>Nillable</b>	no
<b>Abstract</b>	no
<b>Documentation</b>	Identifies a simple underlying asset type that is a cash payment. Used for specifying discounting factors for future cash flows in the pricing and risk model.

**Logical Diagram****XML Instance Representation**

```

<cash
  id=" xsd:ID [0..1]">
    <instrumentId> InstrumentId </instrumentId> [1..*]
    'Identification of the underlying asset, using public and/or private identifiers.'
    <description> xsd:string </description> [0..1]
    'Long name of the underlying asset.'
    <currency> Currency </currency> [1]
    'The currency in which an amount is denominated.'
  </cash>
  
```

**Diagram**

**Schema Component Representation**

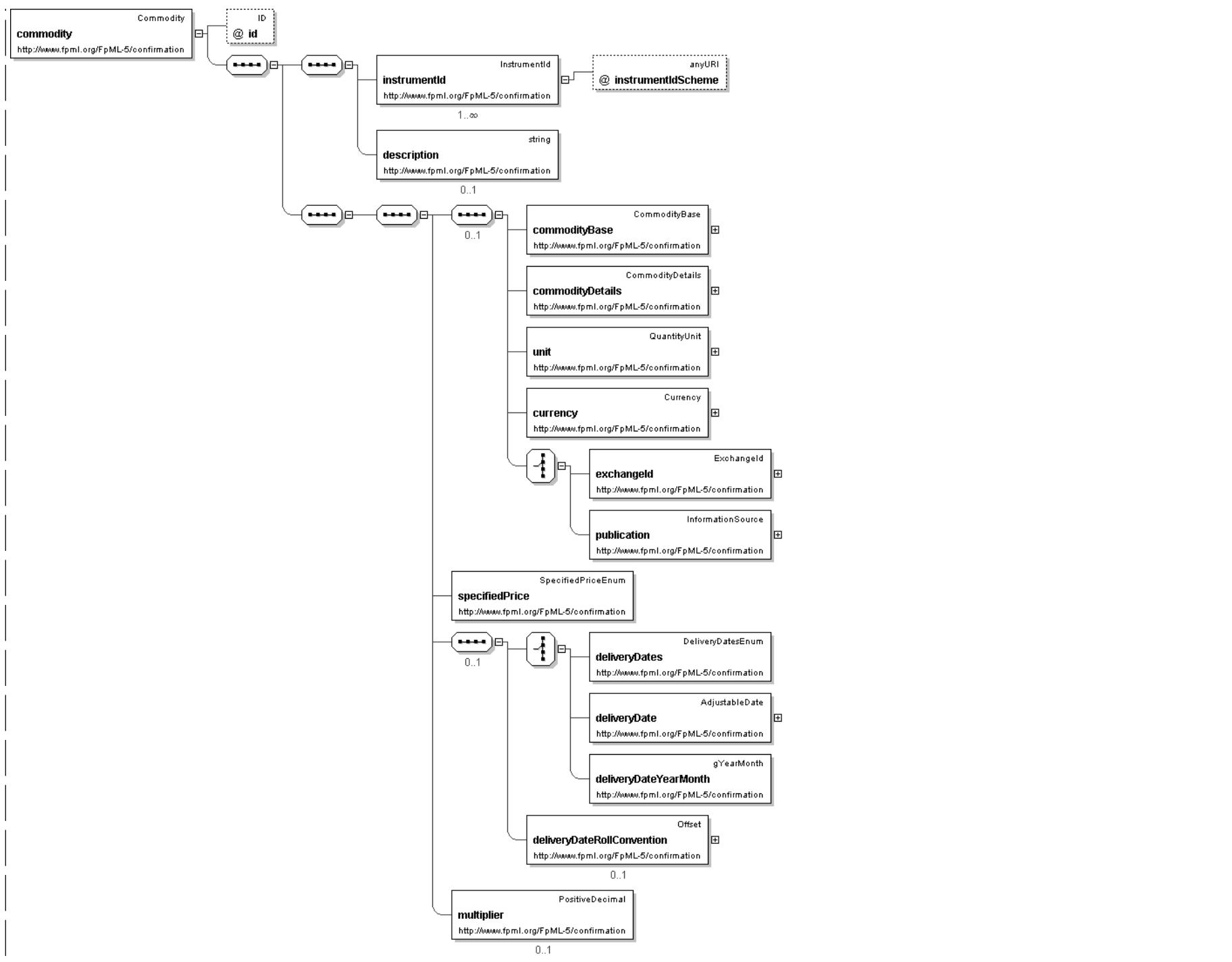
```
<xsd:element name="cash" type="Cash" substitutionGroup="underlyingAsset"/>
```

[top](#)**Element: commodity**

- This element can be used wherever the following element is referenced:
  - underlyingAsset

Name	commodity
Type	Commodity
Nullable	no
Abstract	no
Documentation	Identifies the underlying asset when it is a listed commodity.

**Logical Diagram**

**XML Instance Representation**

```

<commodity
  id=" xsd:ID [0..1]">
  <instrumentId> InstrumentId </instrumentId> [1..*]

```

'Identification of the underlying asset, using public and/or private identifiers.'

<description> xsd:string </description> [0..1]

'Long name of the underlying asset.'

Start Group: CommodityReferencePriceFramework.model [0..1]

<commodityBase> CommodityBase </commodityBase> [1]

'A coding scheme value to identify the base type of the commodity being traded. Where possible, this should follow the naming convention used in the 2005 ISDA Commodity Definitions. For example, \'Oil\''.

<commodityDetails> CommodityDetails </commodityDetails> [1]

'A coding scheme value to identify the commodity being traded more specifically. Where possible, this should follow the naming convention used in the 2005 ISDA Commodity Definitions. For example, \'Brent\''.

<unit> QuantityUnit </unit> [1]

'A coding scheme value to identify the unit in which the undelyer is denominated. Where possible, this should follow the naming convention used in the 2005 ISDA Commodity Definitions.'

<currency> Currency </currency> [1]

'The currency in which the Commodity Reference Price is published.'

Start Choice [1]

<exchangeId> ExchangeId </exchangeId> [1]

'For those commodities being traded with reference to the price of a listed future, the exchange where that future is listed should be specified here.'

<publication> InformationSource </publication> [1]

'For those commodities being traded with reference to a price distributed by a publication, that publication should be specified here.'

End Choice

End Group: CommodityReferencePriceFramework.model

<specifiedPrice> SpecifiedPriceEnum </specifiedPrice> [1]

'The Specified Price is not defined in the Commodity Reference Price and so needs to be stated in the Underlyer definition as it will impact the calculation of the Floating Price.'

Start Sequence [0..1]

Start Choice [1]

<deliveryDates> DeliveryDatesEnum </deliveryDates> [1]

'The Delivery Date is a NearbyMonth, for use when the Commodity Transaction references Futures Contract.'

<deliveryDate> AdjustableDate </deliveryDate> [1]

'The Delivery Date is a fixed, single day.'

<deliveryDateYearMonth> xsd:gYearMonth </deliveryDateYearMonth> [1]

'The Delivery Date is a fixed, single month.'

End Choice

<deliveryDateRollConvention> Offset </deliveryDateRollConvention> [0..1]

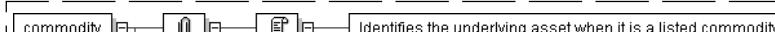
'Specifies, for a Commodity Transaction that references a listed future via the deliveryDates element, the day on which the specified future will roll to the next nearby month when the referenced future expires. If the future will not roll at all - i.e. the price will be taken from the expiring contract, 0 should be specified here. If the future will roll to the next nearby on the last trading day - i.e. the price will be taken from the next nearby on the last trading day, then 1 should be specified and so on.'

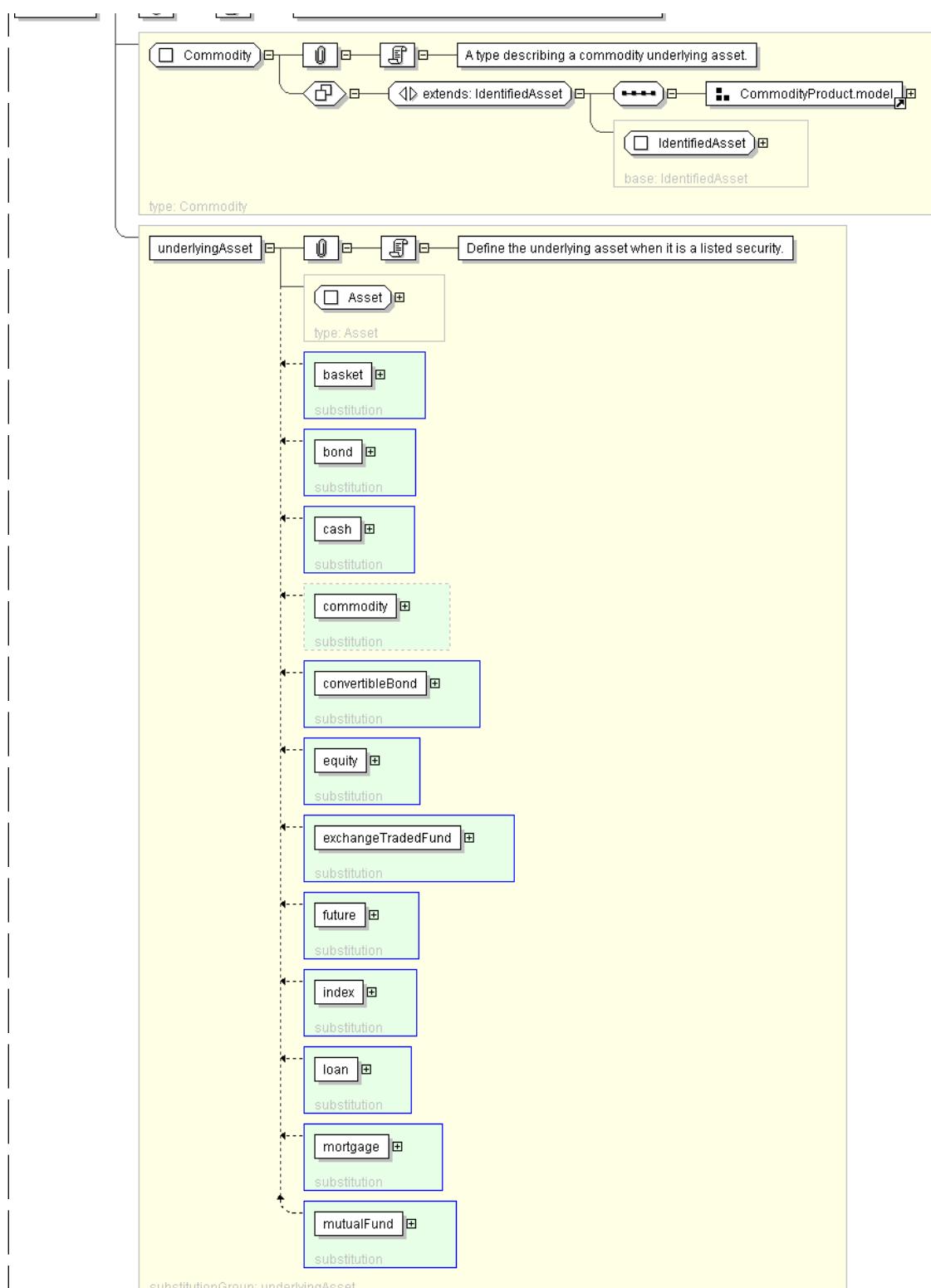
End Sequence

<multiplier> PositiveDecimal </multiplier> [0..1]

'Specifies the multiplier associated with a Transaction.'

</commodity>

**Diagram**



**Schema Component Representation**

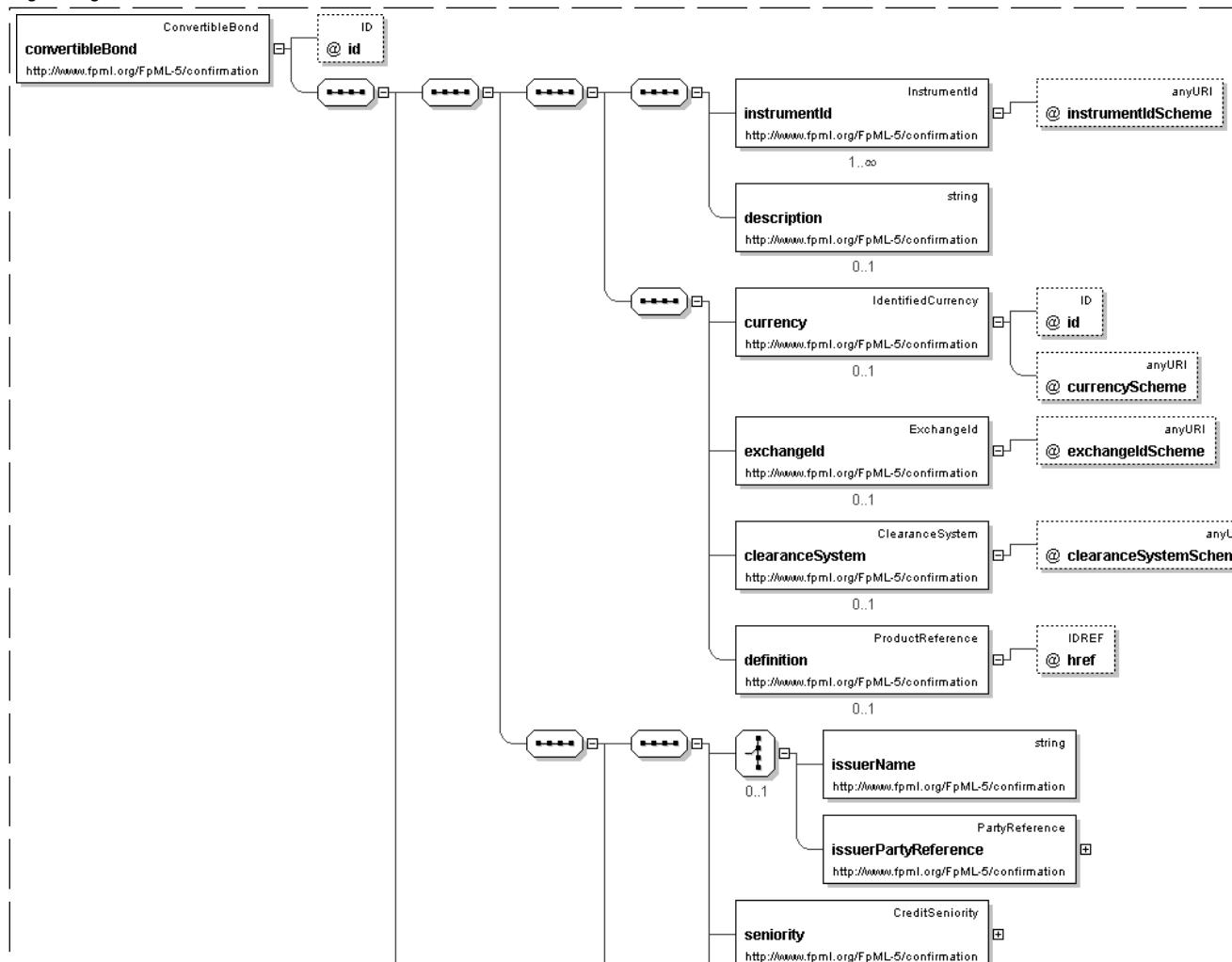
```
<xsd:element name="commodity" type="Commodity" substitutionGroup="underlyingAsset"/>
```

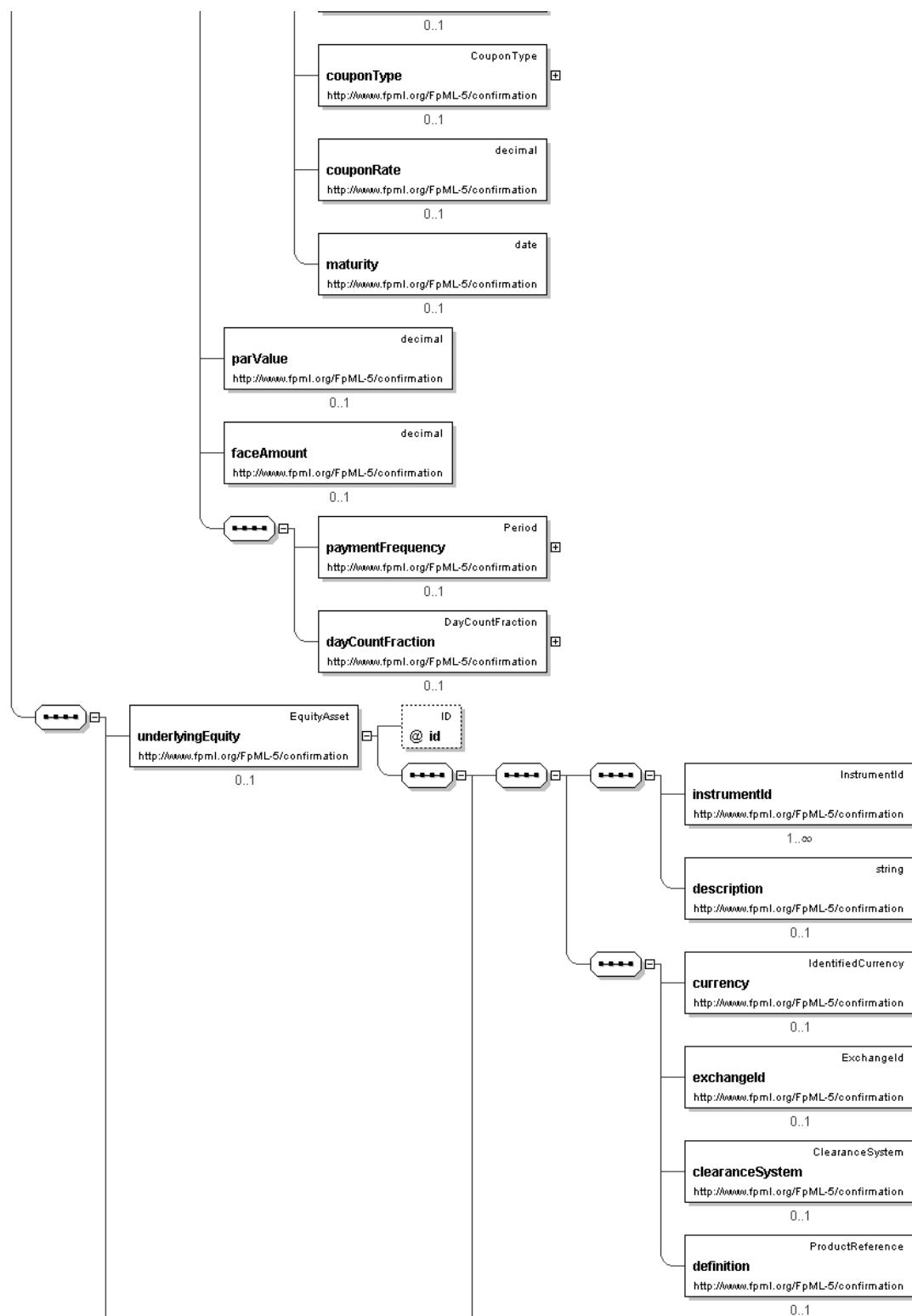
top

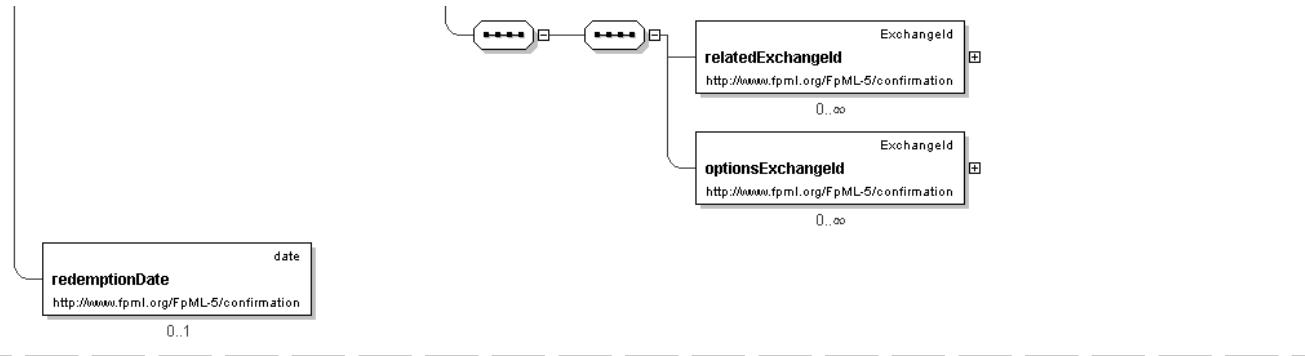
**Element: convertibleBond**

- This element can be used wherever the following element is referenced:
  - underlyingAsset

<b>Name</b>	convertibleBond
<b>Used by (from the same schema document)</b>	Model Group <b>BondChoice.model</b>
<b>Type</b>	ConvertibleBond
<b>Nillable</b>	no
<b>Abstract</b>	no
<b>Documentation</b>	Identifies the underlying asset when it is a convertible bond.

**Logical Diagram**



**XML Instance Representation**

```

<convertibleBond
id=" xsd:ID [0..1]">
  <instrumentId> InstrumentId </instrumentId> [1..*]
  'Identification of the underlying asset, using public and/or private identifiers.'

  <description> xsd:string </description> [0..1]
  'Long name of the underlying asset.'

  <currency> IdentifiedCurrency </currency> [0..1]
  'Trading currency of the underlyer when transacted as a cash instrument.'

  <exchangeId> ExchangeId </exchangeId> [0..1]
  'Identification of the exchange on which this asset is transacted for the purposes
  of calculating a contractual payoff. The term \"Exchange\" is assumed to have the meaning
  as defined in the ISDA 2002 Equity Derivatives Definitions.'

  <clearanceSystem> ClearanceSystem </clearanceSystem> [0..1]
  'Identification of the clearance system associated with the transaction exchange.'

  <definition> ProductReference </definition> [0..1]
  'An optional reference to a full FpML product that defines the simple product in
  greater detail. In case of inconsistency between the terms of the simple product and those
  of the detailed definition, the values in the simple product override those in the
  detailed definition.'

Start Choice [0..1]
  'Specifies the issuer name of a fixed income security or convertible bond. This name can
  either be explicitly stated, or specified as an href into another element of the document,
  such as the obligor.'

```

```

    <issuerName> xsd:string </issuerName> [1]
    <issuerPartyReference> PartyReference </issuerPartyReference> [1]
End Choice

```

```

  <seniority> CreditSeniority </seniority> [0..1]
  'The repayment precedence of a debt instrument.'

```

```

  <couponType> CouponType </couponType> [0..1]
  'Specifies if the bond has a variable coupon, step-up/down coupon or a zero-coupon.'

```

```

  <couponRate> xsd:decimal </couponRate> [0..1]
  'Specifies the coupon rate (expressed in percentage) of a fixed income security or
  convertible bond.'

```

```

  <maturity> xsd:date </maturity> [0..1]
  'The date when the principal amount of a security becomes due and payable.'

```

```

  <parValue> xsd:decimal </parValue> [0..1]
  'Specifies the nominal amount of a fixed income security or convertible bond.'

```

```

<faceAmount> xsd:decimal </faceAmount> [0..1]
'Specifies the total amount of the issue. Corresponds to the par value multiplied by the
number of issued security.'

<paymentFrequency> Period </paymentFrequency> [0..1]
'Specifies the frequency at which the bond pays, e.g. 6M.'

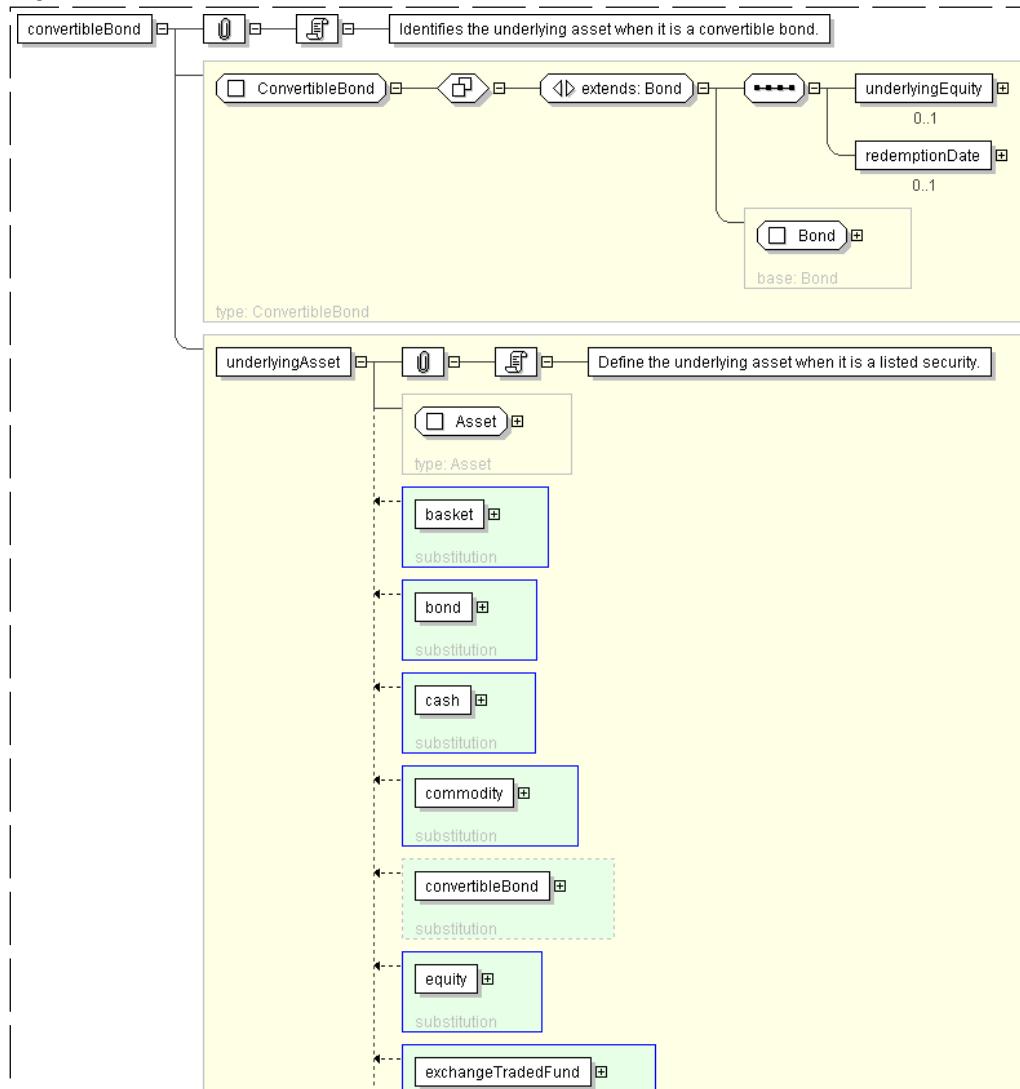
<dayCountFraction> DayCountFraction </dayCountFraction> [0..1]
'The day count basis for the bond.'

<underlyingEquity> EquityAsset </underlyingEquity> [0..1]
'Specifies the equity in which the convertible bond can be converted.'

<redemptionDate> xsd:date </redemptionDate> [0..1]
'Earlier date between the convertible bond put dates and its maturity date.'

```

&lt;/convertibleBond&gt;

**Diagram**

**Schema Component Representation**

```
<xsd:element name="convertibleBond" type=" ConvertibleBond"
  " substitutionGroup="underlyingAsset" />
```

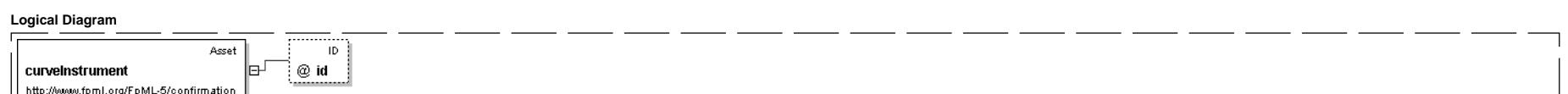
top

**Element: curveInstrument**

- The following elements can be used wherever this element is referenced:

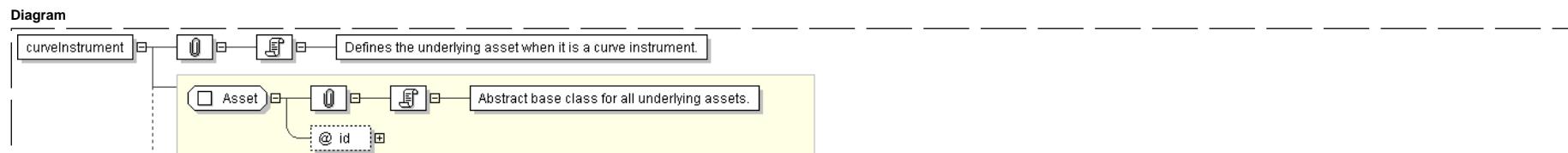
- deposit
- fx
- rateIndex
- simpleCreditDefaultSwap
- simpleFra
- simpleIrSwap

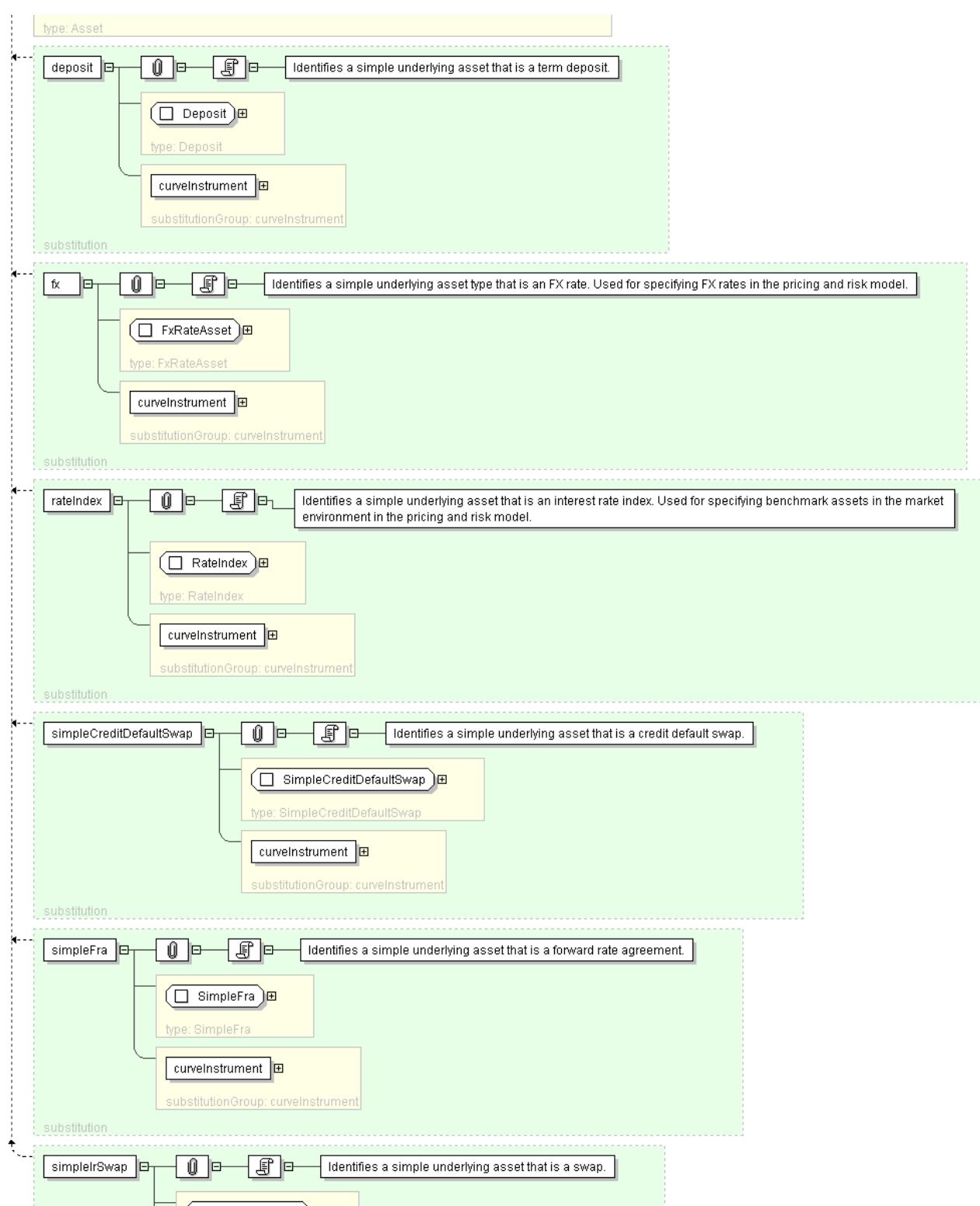
Name	curveInstrument
Type	Asset
Nillable	no
Abstract	yes
Documentation	Defines the underlying asset when it is a curve instrument.



XML Instance Representation

```
<curveInstrument
  id=" xsd:ID [0..1]" />
```





**Schema Component Representation**

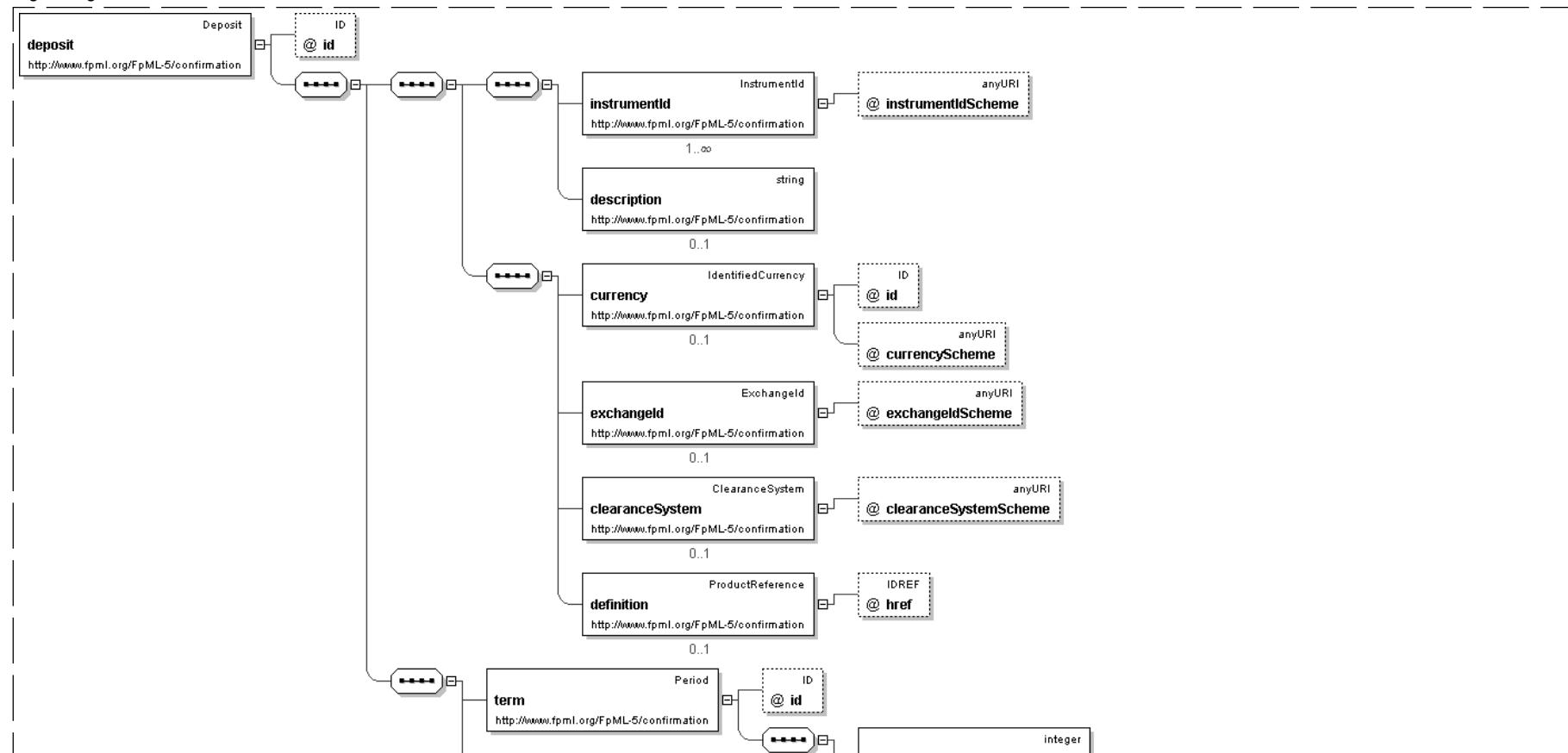
```
<xsd:element name="curveInstrument" type="Asset" abstract="true"/>
```

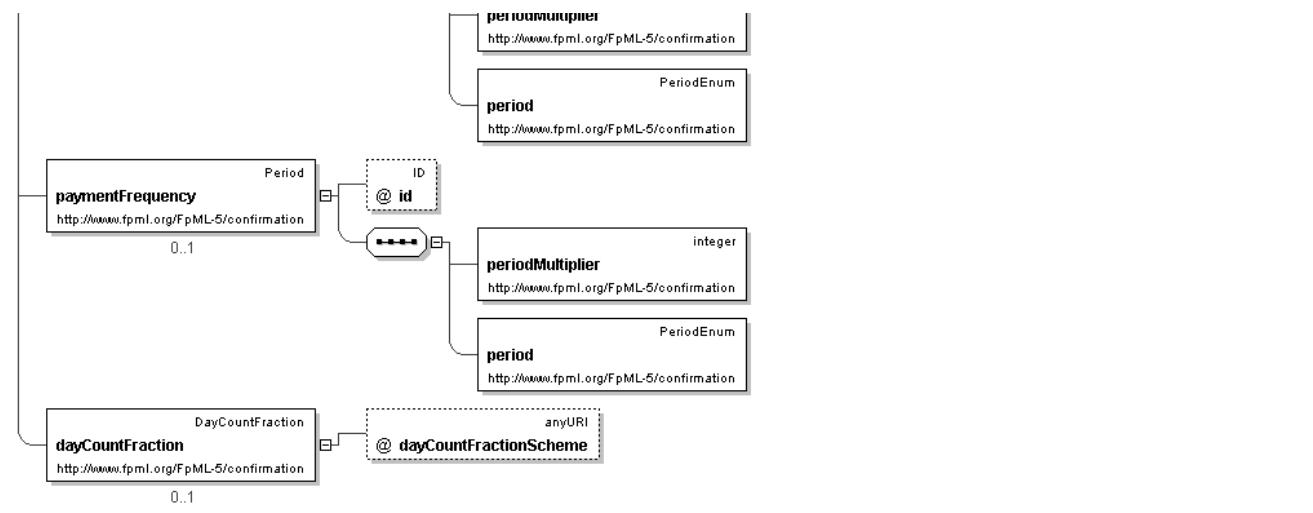
top

**Element: deposit**

- This element can be used wherever the following element is referenced:
  - curveInstrument

<b>Name</b>	deposit
<b>Type</b>	Deposit
<b>Nillable</b>	no
<b>Abstract</b>	no
<b>Documentation</b>	Identifies a simple underlying asset that is a term deposit.

**Logical Diagram**

**XML Instance Representation**

```

<deposit
id=" xsd:ID [0..1]">
  <instrumentId> InstrumentId </instrumentId> [1..*]
  'Identification of the underlying asset, using public and/or private identifiers.'

  <description> xsd:string </description> [0..1]
  'Long name of the underlying asset.'

  <currency> IdentifiedCurrency </currency> [0..1]
  'Trading currency of the underlyer when transacted as a cash instrument.'

  <exchangeId> ExchangeId </exchangeId> [0..1]
  'Identification of the exchange on which this asset is transacted for the purposes
  of calculating a contractual payoff. The term "Exchange" is assumed to have the meaning
  as defined in the ISDA 2002 Equity Derivatives Definitions.'

  <clearanceSystem> ClearanceSystem </clearanceSystem> [0..1]
  'Identification of the clearance system associated with the transaction exchange.'

  <definition> ProductReference </definition> [0..1]
  'An optional reference to a full FpML product that defines the simple product in
  greater detail. In case of inconsistency between the terms of the simple product and those
  of the detailed definition, the values in the simple product override those in the
  detailed definition.'

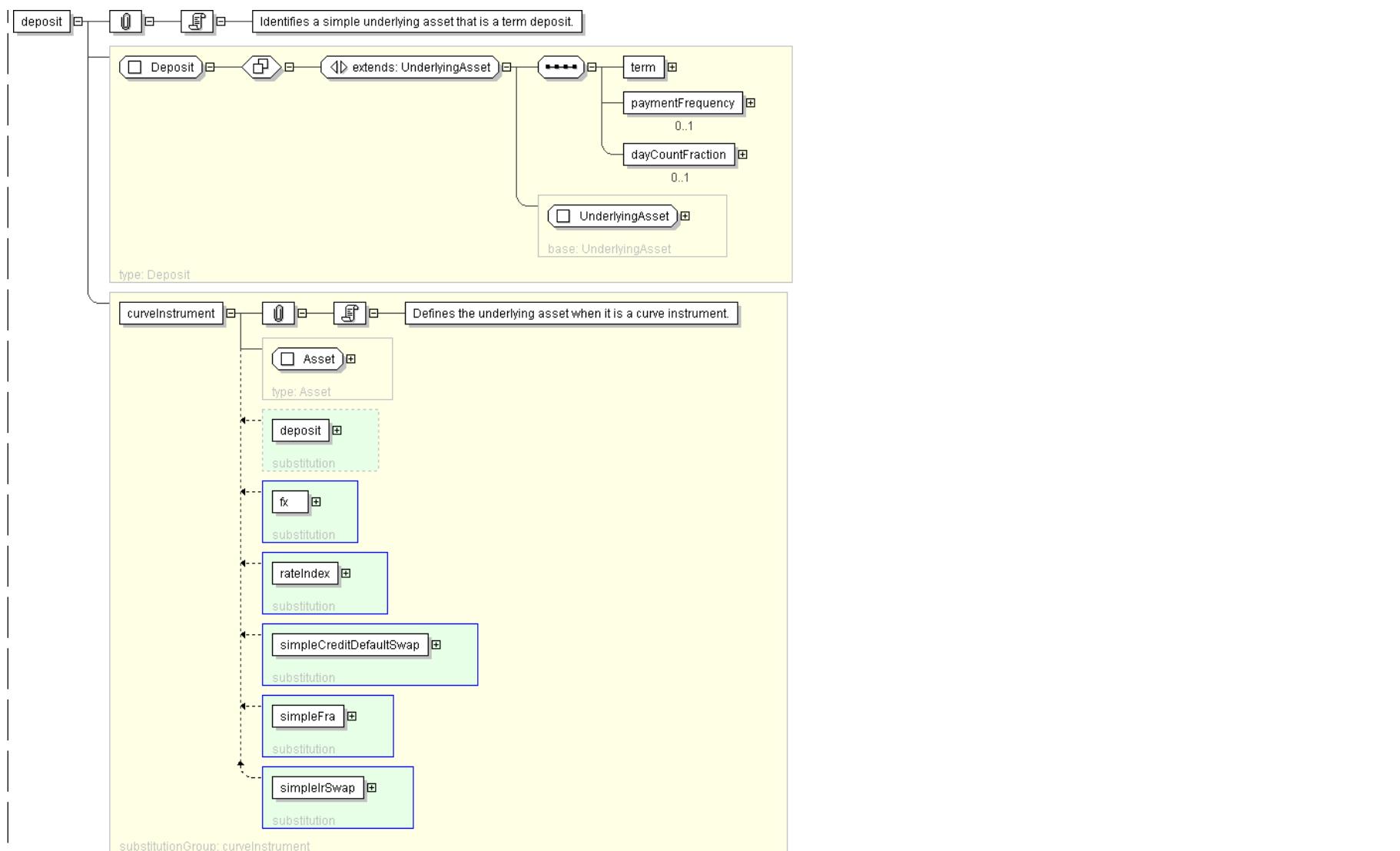
  <term> Period </term> [1]
  'Specifies the term of the deposit, e.g. 5Y.'

  <paymentFrequency> Period </paymentFrequency> [0..1]
  'Specifies the frequency at which the deposit pays, e.g. 6M.'

  <dayCountFraction> DayCountFraction </dayCountFraction> [0..1]
  'The day count basis for the deposit.'

</deposit>
  
```

**Diagram**

**Schema Component Representation**

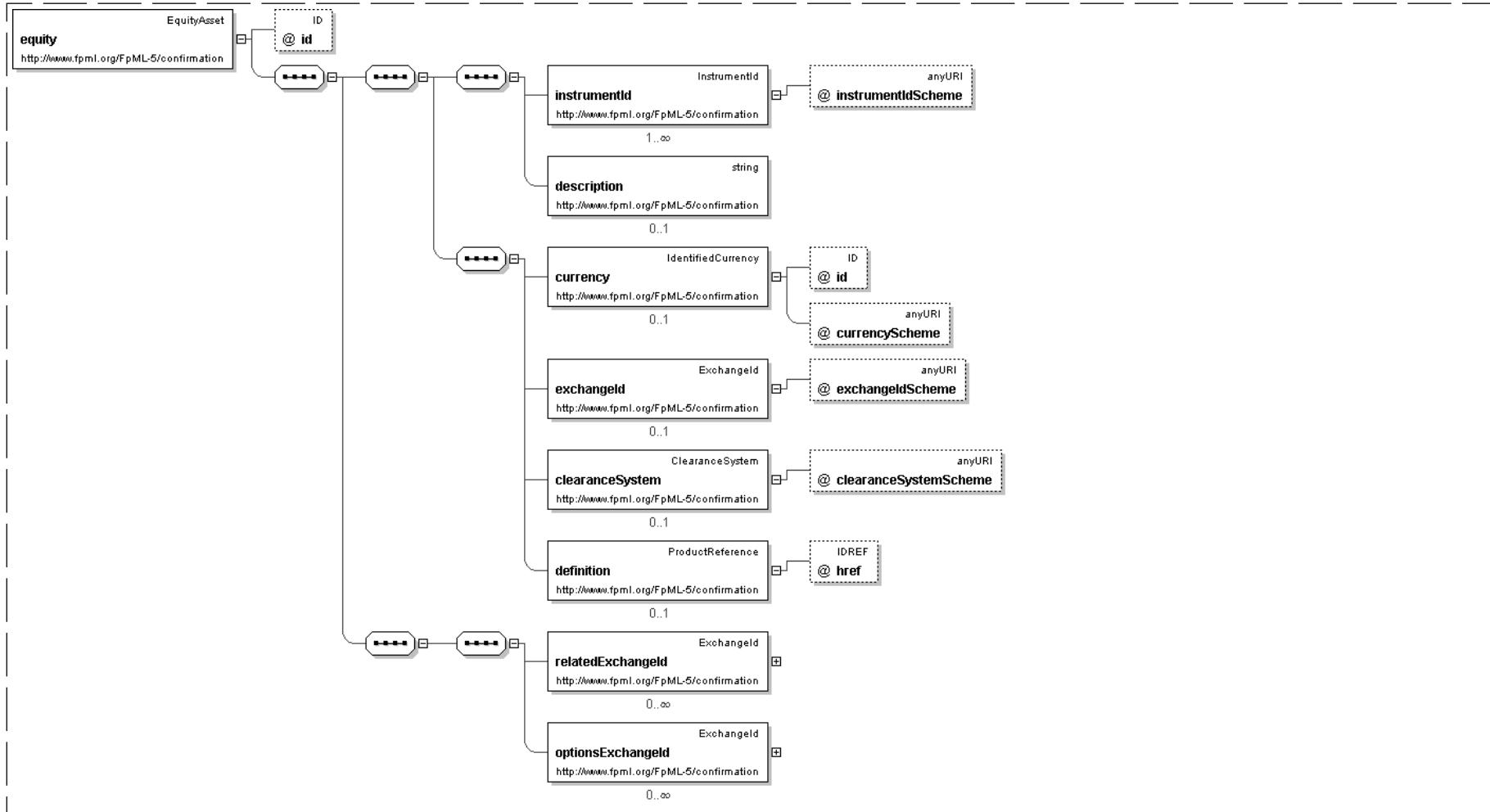
```
<xsd:element name="deposit" type="#Deposit" substitutionGroup="curveInstrument"/>
```

top

**Element: equity**

- This element can be used wherever the following element is referenced:
  - underlyingAsset

Name	equity
Type	<a href="#">EquityAsset</a>
Nillable	no
Abstract	no
Documentation	Identifies the underlying asset when it is a listed equity.

**Logical Diagram****XML Instance Representation**

```

<equity
  id=" xsd:ID [0..1]">
  <instrumentId> InstrumentId </instrumentId> [1..*]
  'Identification of the underlying asset, using public and/or private identifiers.'

  <description> xsd:string </description> [0..1]
  'Long name of the underlying asset.'

  <currency> IdentifiedCurrency </currency> [0..1]
  'Trading currency of the underlyer when transacted as a cash instrument.'

  <exchangeId> ExchangeId </exchangeId> [0..1]
  'Identification of the exchange on which this asset is transacted for the purposes
  of calculating a contractual payoff. The term ("Exchange") is assumed to have the meaning
  as defined in the ISDA 2002 Equity Derivatives Definitions.'

  <clearanceSystem> ClearanceSystem </clearanceSystem> [0..1]
  'Identification of the clearance system associated with the transaction exchange.'

```

```
<definition> ProductReference </definition> [0..1]
```

'An optional reference to a full FpML product that defines the simple product in greater detail. In case of inconsistency between the terms of the simple product and those of the detailed definition, the values in the simple product override those in the detailed definition.'

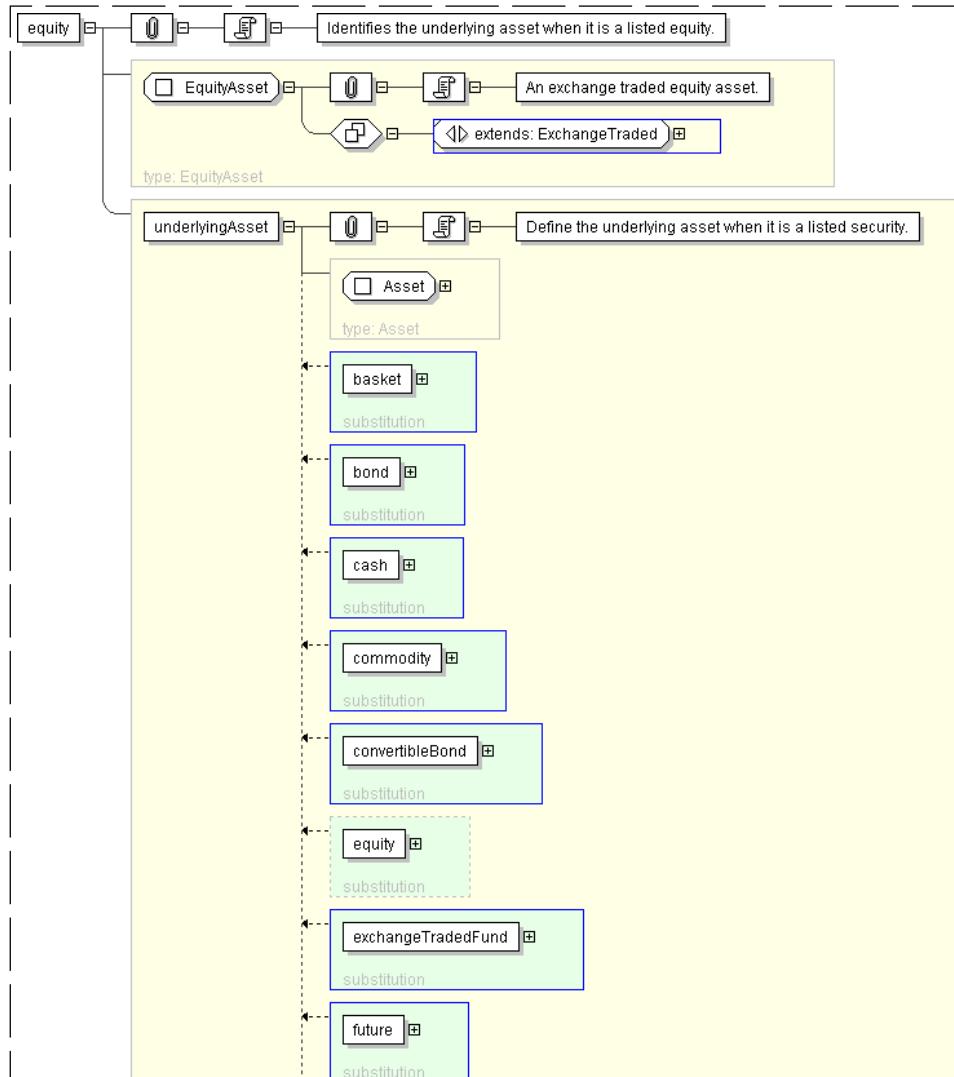
```
<relatedExchangeId> ExchangeId </relatedExchangeId> [0..*]
```

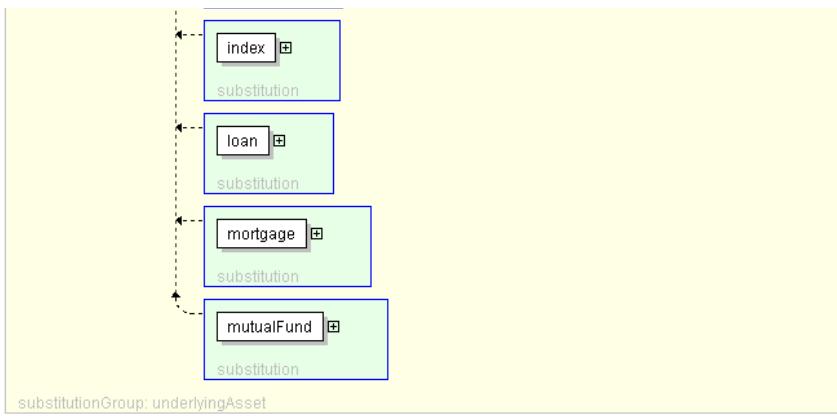
'A short form unique identifier for a related exchange. If the element is not present then the exchange shall be the primary exchange on which listed futures and options on the underlying are listed. The term \"Exchange\" is assumed to have the meaning as defined in the ISDA 2002 Equity Derivatives Definitions.'

```
<optionsExchangeId> ExchangeId </optionsExchangeId> [0..*]
```

'A short form unique identifier for an exchange on which the reference option contract is listed. This is to address the case where the reference exchange for the future is different than the one for the option. The options Exchange is referenced on share options when Merger Elections are selected as Options Exchange Adjustment.'

&lt;/equity&gt;

**Diagram**

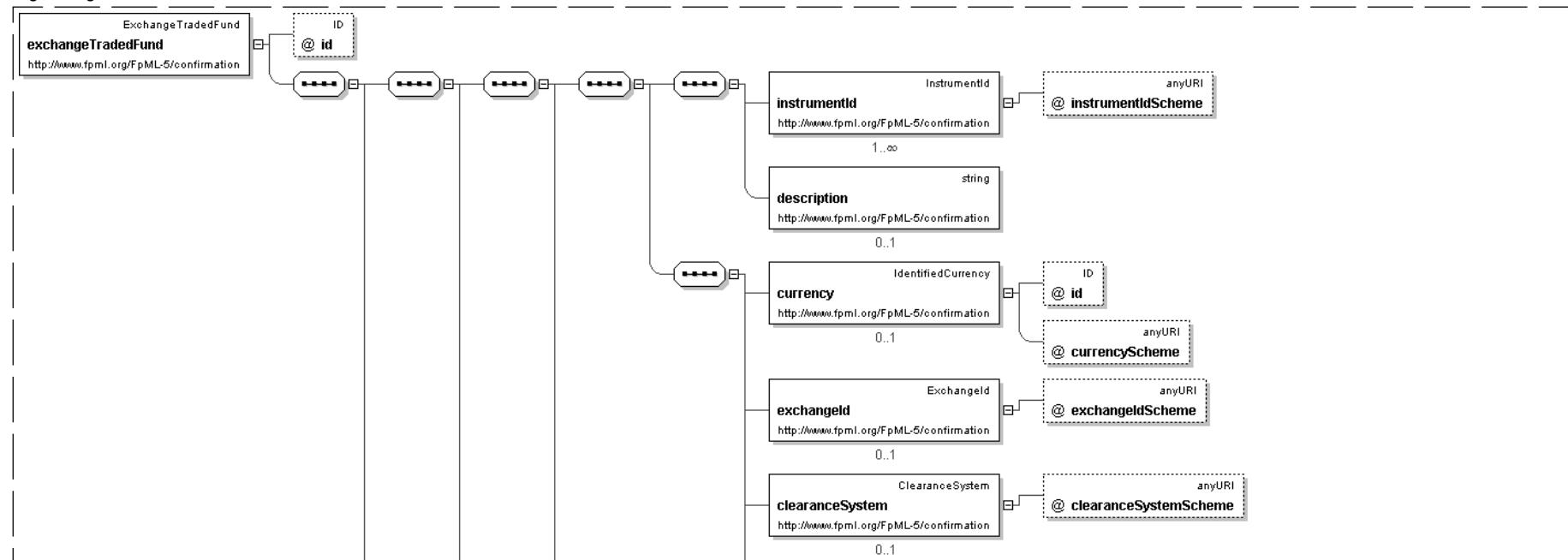
**Schema Component Representation**

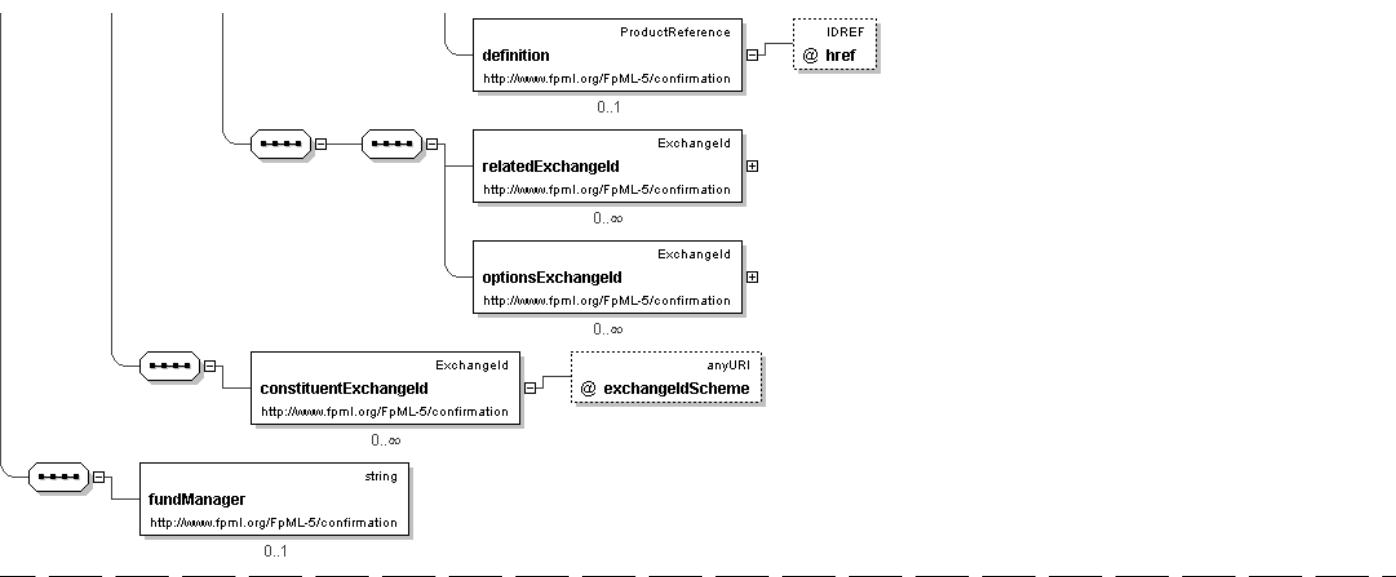
```
<xsd:element name="equity" type="#EquityAsset" substitutionGroup="underlyingAsset"/>
```

[top](#)**Element: exchangeTradedFund**

- This element can be used wherever the following element is referenced:
  - underlyingAsset

<b>Name</b>	exchangeTradedFund
<b>Type</b>	ExchangeTradedFund
<b>Nullable</b>	no
<b>Abstract</b>	no
<b>Documentation</b>	Identifies the underlying asset when it is an exchange-traded fund.

**Logical Diagram**

**XML Instance Representation**

```

<exchangeTradedFund
id=" xsd:ID [0..1]">
    <instrumentId> InstrumentId </instrumentId> [1..*]
    'Identification of the underlying asset, using public and/or private identifiers.'

    <description> xsd:string </description> [0..1]
    'Long name of the underlying asset.'

    <currency> IdentifiedCurrency </currency> [0..1]
    'Trading currency of the underlyer when transacted as a cash instrument.'

    <exchangeId> ExchangeId </exchangeId> [0..1]
    'Identification of the exchange on which this asset is transacted for the purposes
    of calculating a contractual payoff. The term \"Exchange\" is assumed to have the meaning
    as defined in the ISDA 2002 Equity Derivatives Definitions.'

    <clearanceSystem> ClearanceSystem </clearanceSystem> [0..1]
    'Identification of the clearance system associated with the transaction exchange.'

    <definition> ProductReference </definition> [0..1]
    'An optional reference to a full FpML product that defines the simple product in
    greater detail. In case of inconsistency between the terms of the simple product and those
    of the detailed definition, the values in the simple product override those in the
    detailed definition.'

    <relatedExchangeId> ExchangeId </relatedExchangeId> [0..*]
    'A short form unique identifier for a related exchange. If the element is not present then
    the exchange shall be the primary exchange on which listed futures and options on
    the underlying are listed. The term \"Exchange\" is assumed to have the meaning as defined
    in the ISDA 2002 Equity Derivatives Definitions.'

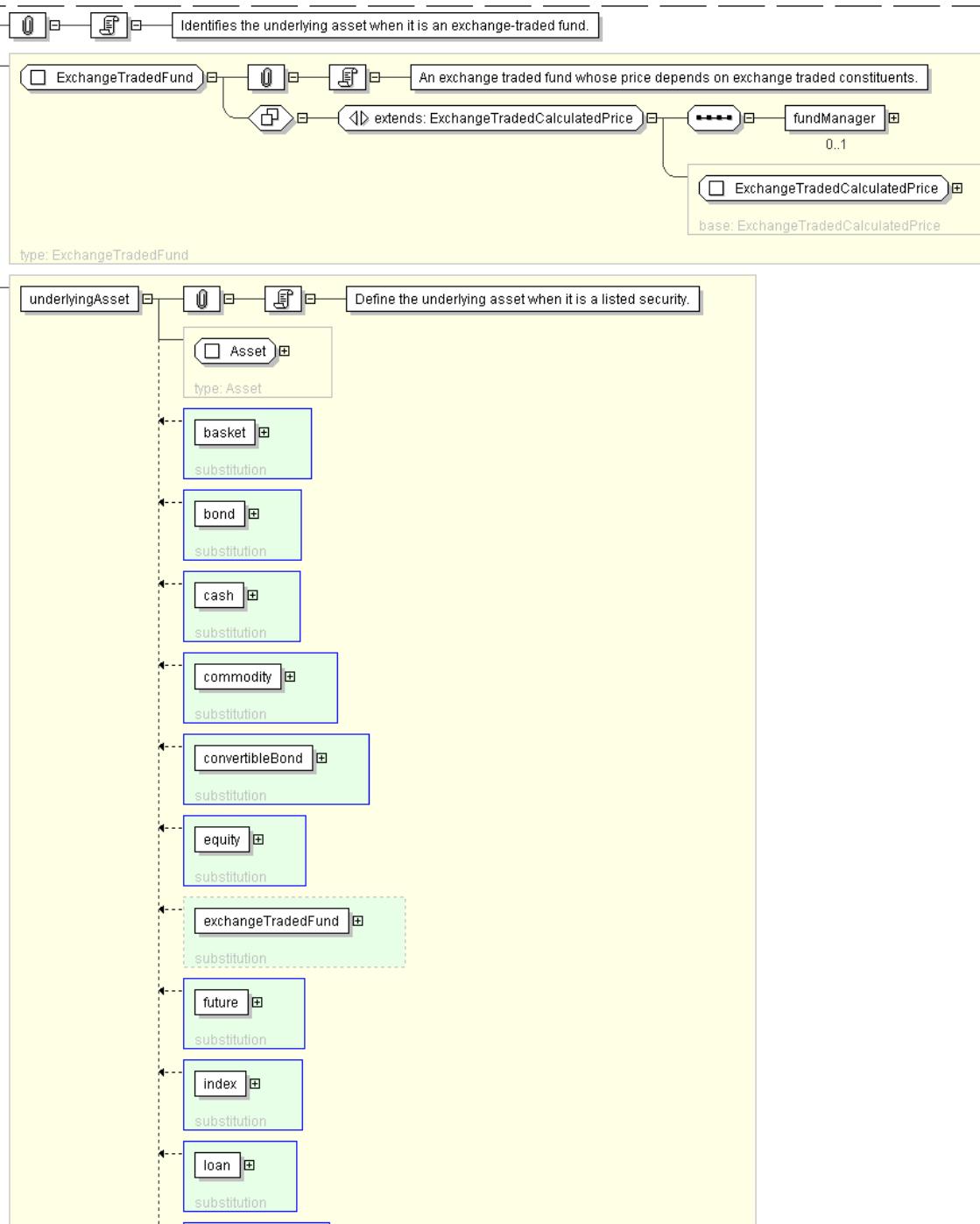
    <optionsExchangeId> ExchangeId </optionsExchangeId> [0..*]
    'A short form unique identifier for an exchange on which the reference option contract
    is listed. This is to address the case where the reference exchange for the future is
    different than the one for the option. The Options Exchange is referenced on share options
    when Merger Elections are selected as Options Exchange Adjustment.'

    <constituentExchangeId> ExchangeId </constituentExchangeId> [0..*]
    'Identification of all the exchanges where constituents are traded. The term \"Exchange\"
    is assumed to have the meaning as defined in the ISDA 2002 Equity Derivatives Definitions.'

```

```
<fundManager> xsd:string </fundManager> [0..1]
'Specifies the fund manager that is in charge of the fund.'
```

&lt;/exchangeTradedFund&gt;

**Diagram**

**Schema Component Representation**

```

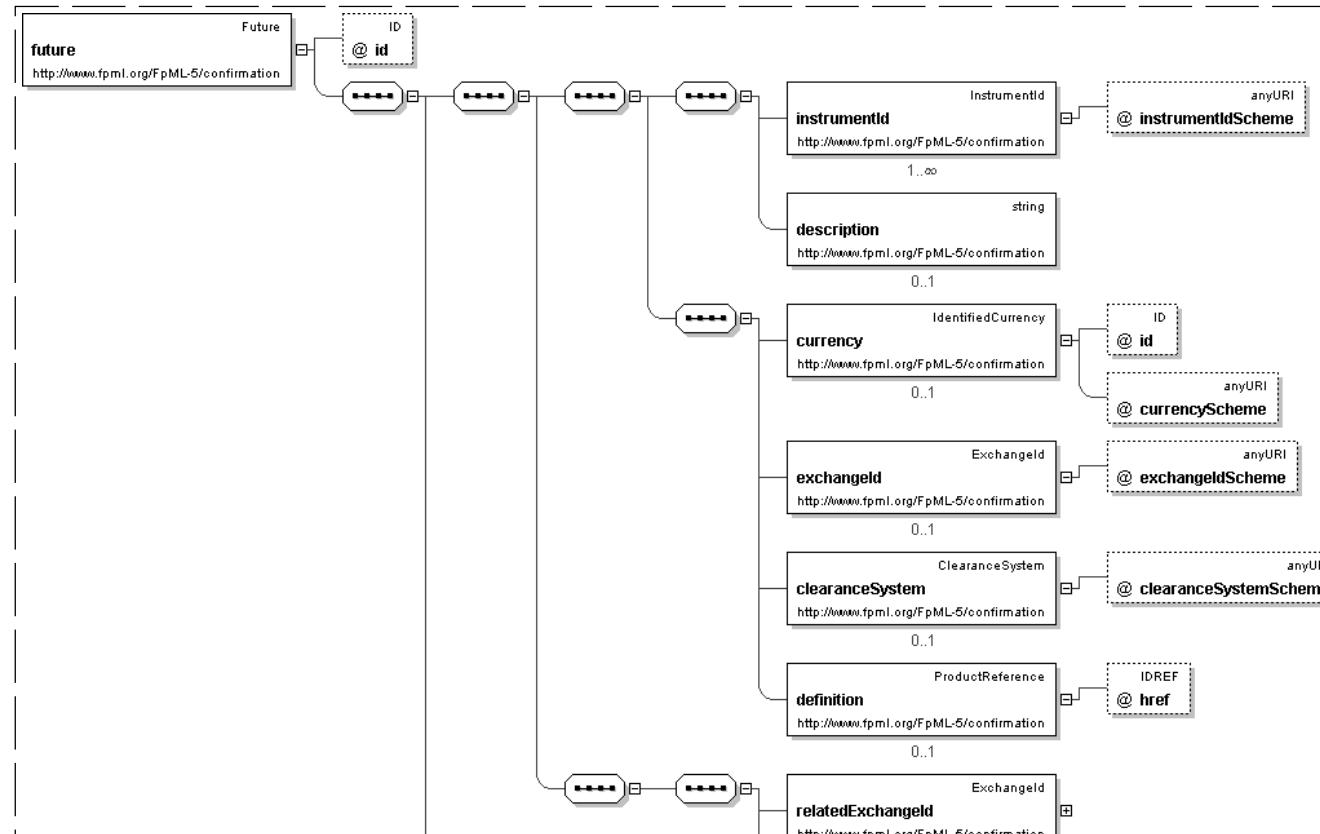
<xsd:element name="exchangeTradedFund" type=" ExchangeTradedFund"
  " substitutionGroup="underlyingAsset"/>
  
```

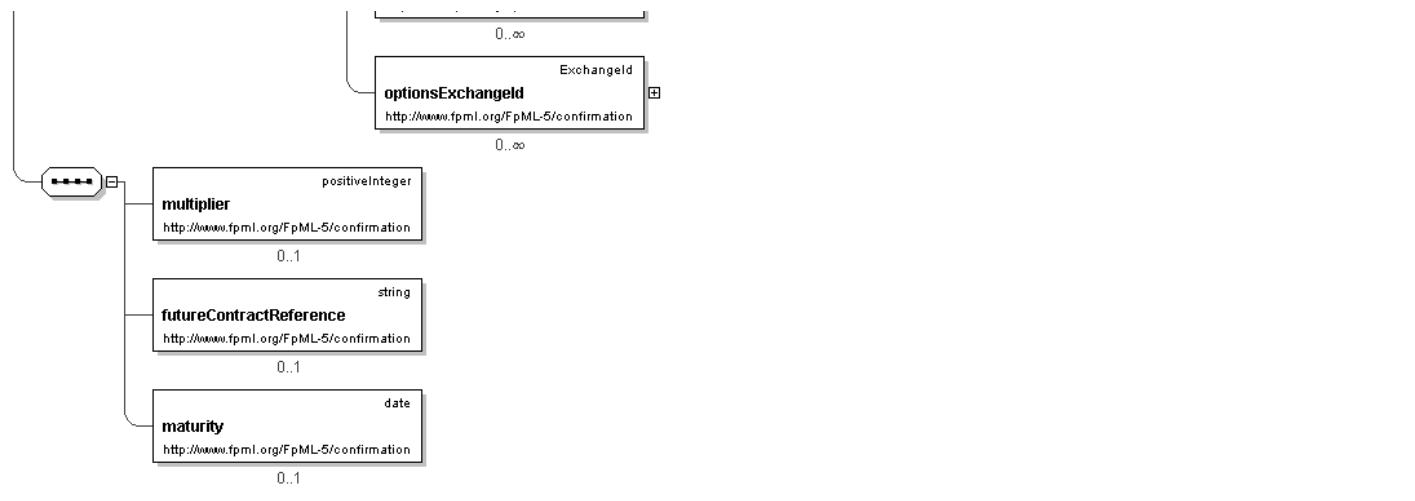
top

**Element: future**

- This element can be used wherever the following element is referenced:
  - underlyingAsset

Name	future
Type	Future
Nillable	no
Abstract	no
Documentation	Identifies the underlying asset when it is a listed future contract.

**Logical Diagram**

**XML Instance Representation**

```

<future
id=" xsd:ID [0..1]">
  <instrumentId> InstrumentId </instrumentId> [1..*]
  'Identification of the underlying asset, using public and/or private identifiers.'

  <description> xsd:string </description> [0..1]
  'Long name of the underlying asset.'

  <currency> IdentifiedCurrency </currency> [0..1]
  'Trading currency of the underlyer when transacted as a cash instrument.'

  <exchangeId> ExchangeId </exchangeId> [0..1]
  'Identification of the exchange on which this asset is transacted for the purposes
  of calculating a contractual payoff. The term \"Exchange\" is assumed to have the meaning
  as defined in the ISDA 2002 Equity Derivatives Definitions.'

  <clearanceSystem> ClearanceSystem </clearanceSystem> [0..1]
  'Identification of the clearance system associated with the transaction exchange.'

  <definition> ProductReference </definition> [0..1]
  'An optional reference to a full FpML product that defines the simple product in
  greater detail. In case of inconsistency between the terms of the simple product and those
  of the detailed definition, the values in the simple product override those in the
  detailed definition.'

  <relatedExchangeId> ExchangeId </relatedExchangeId> [0..*]
  'A short form unique identifier for a related exchange. If the element is not present then
  the exchange shall be the primary exchange on which listed futures and options on
  the underlying are listed. The term \"Exchange\" is assumed to have the meaning as defined
  in the ISDA 2002 Equity Derivatives Definitions.'

  <optionsExchangeId> ExchangeId </optionsExchangeId> [0..*]
  'A short form unique identifier for an exchange on which the reference option contract
  is listed. This is to address the case where the reference exchange for the future is
  different than the one for the option. The options Exchange is referenced on share options
  when Merger Elections are selected as Options Exchange Adjustment.'

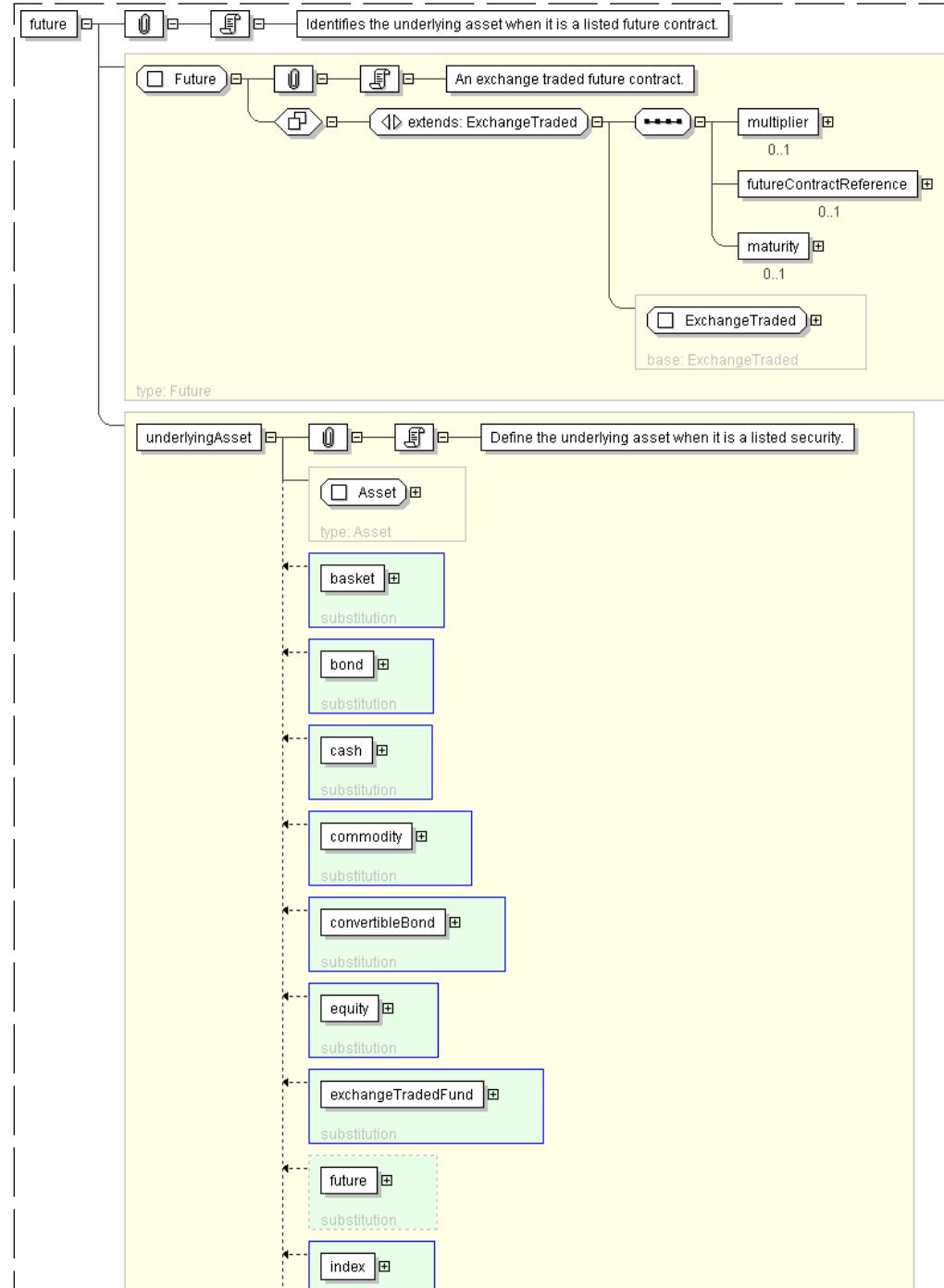
  <multiplier> xsd:positiveInteger </multiplier> [0..1]
  'Specifies the contract multiplier that can be associated with the number of units.'

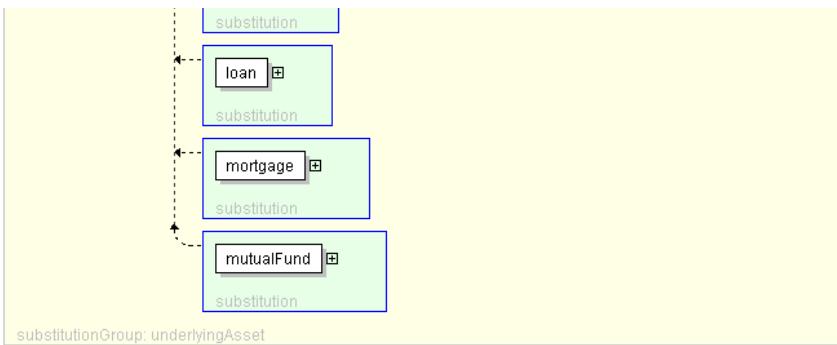
  <futureContractReference> xsd:string </futureContractReference> [0..1]
  'Specifies the future contract that can be referenced, besides the equity or index
  reference defined as part of the UnderlyerAsset type.'

```

```
<maturity> xsd:date </maturity> [0..1]
  'The date when the future contract expires.'
```

&lt;/future&gt;

**Diagram**

**Schema Component Representation**

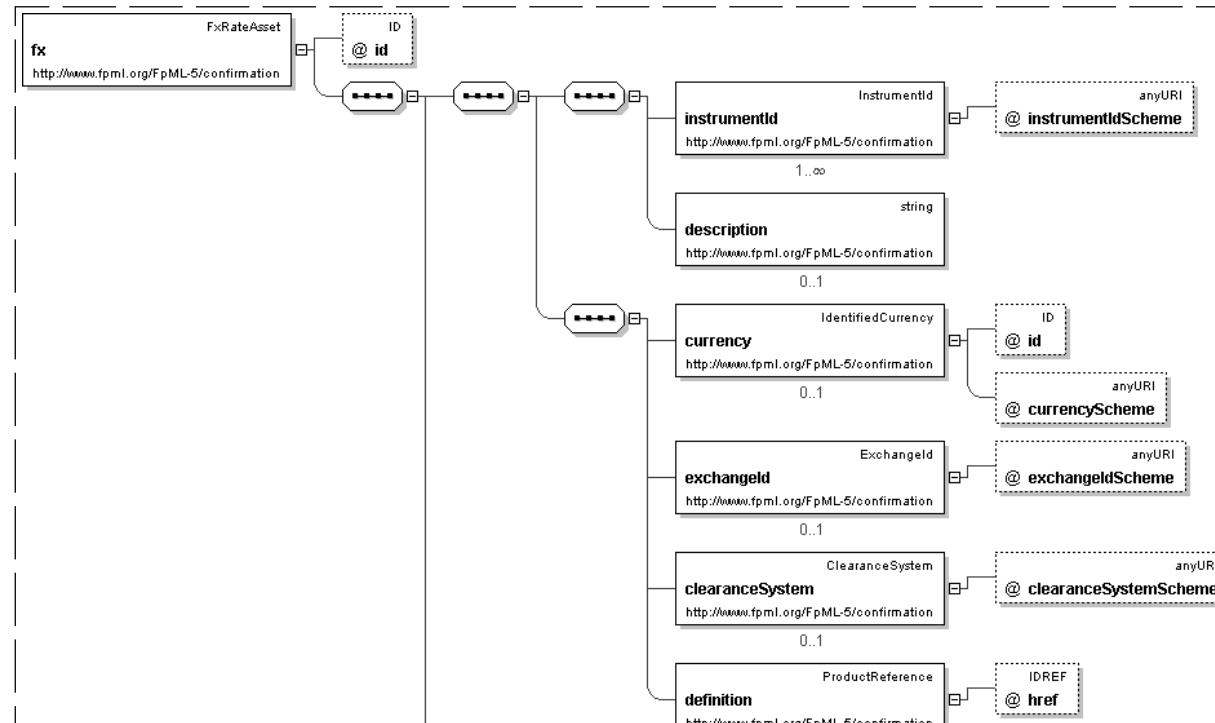
```
<xsd:element name="future" type=" Future " substitutionGroup="underlyingAsset"/>
```

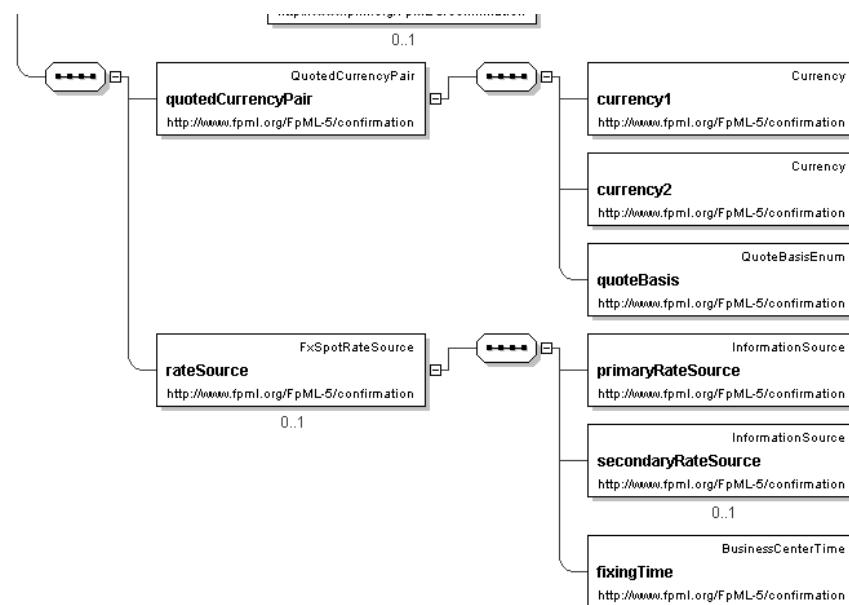
top

**Element: fx**

- This element can be used wherever the following element is referenced:
  - curveInstrument

<b>Name</b>	fx
<b>Type</b>	FxRateAsset
<b>Nillable</b>	no
<b>Abstract</b>	no
<b>Documentation</b>	Identifies a simple underlying asset type that is an FX rate. Used for specifying FX rates in the pricing and risk model.

**Logical Diagram**

**XML Instance Representation**

```

<fx
  id=" xsd:ID [0..1]">
  <instrumentId> InstrumentId </instrumentId> [1..*]
  'Identification of the underlying asset, using public and/or private identifiers.'

  <description> xsd:string </description> [0..1]
  'Long name of the underlying asset.'

  <currency> IdentifiedCurrency </currency> [0..1]
  'Trading currency of the underlyer when transacted as a cash instrument.'

  <exchangeId> ExchangeId </exchangeId> [0..1]
  'Identification of the exchange on which this asset is transacted for the purposes
  of calculating a contractual payoff. The term \"Exchange\" is assumed to have the meaning
  as defined in the ISDA 2002 Equity Derivatives Definitions.'

  <clearanceSystem> ClearanceSystem </clearanceSystem> [0..1]
  'Identification of the clearance system associated with the transaction exchange.'

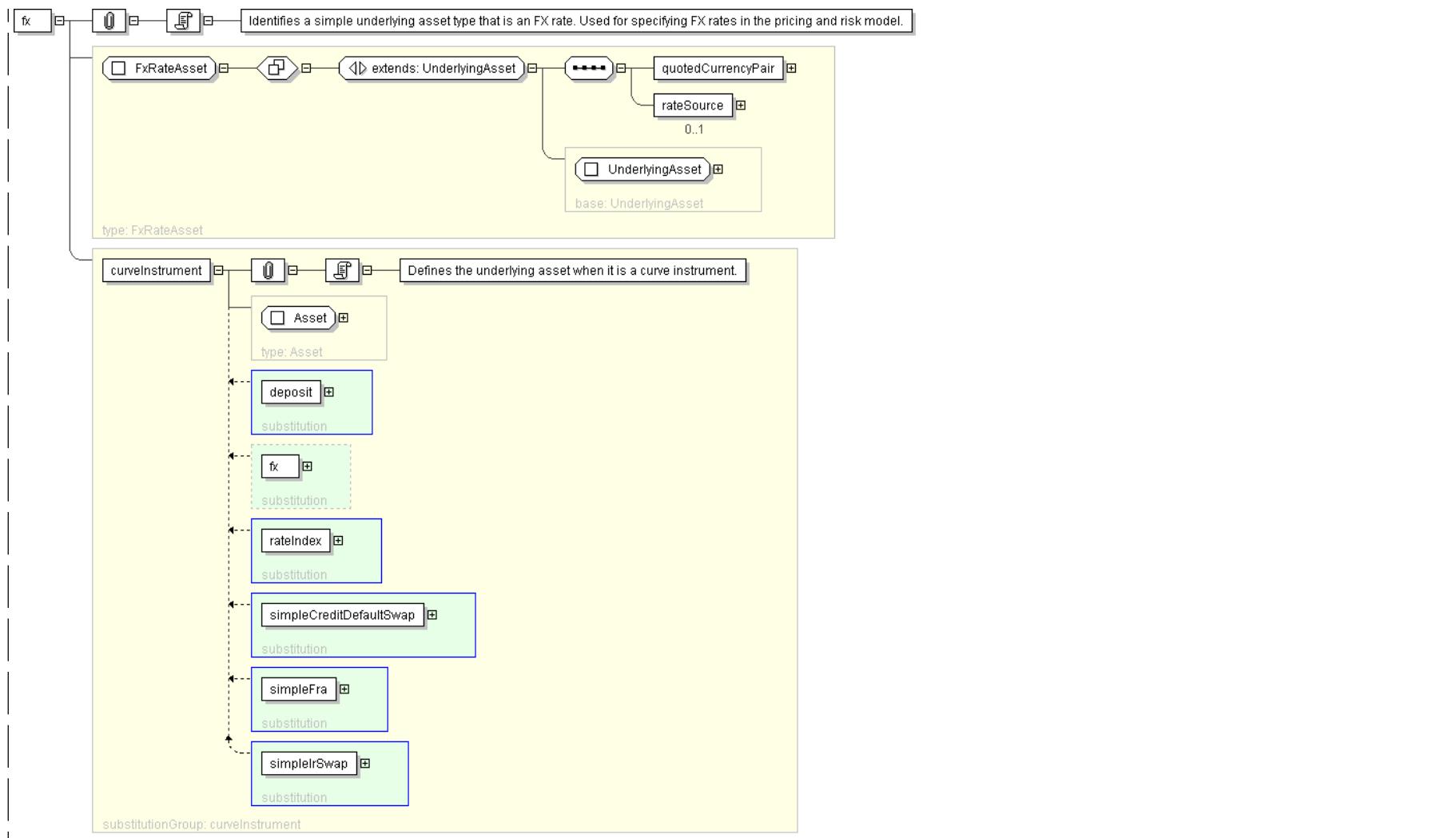
  <definition> ProductReference </definition> [0..1]
  'An optional reference to a full FpML product that defines the simple product in
  greater detail. In case of inconsistency between the terms of the simple product and those
  of the detailed definition, the values in the simple product override those in the
  detailed definition.'

  <quotedCurrencyPair> QuotedCurrencyPair </quotedCurrencyPair> [1]
  'Defines the two currencies for an FX trade and the quotation relationship between the
  two currencies.'

  <rateSource> FxSpotRateSource </rateSource> [0..1]
  'Defines the source of the FX rate.'

</fx>
  
```

**Diagram**

**Schema Component Representation**

```
<xsd:element name="fx" type="FxRateAsset" substitutionGroup="curveInstrument"/>
```

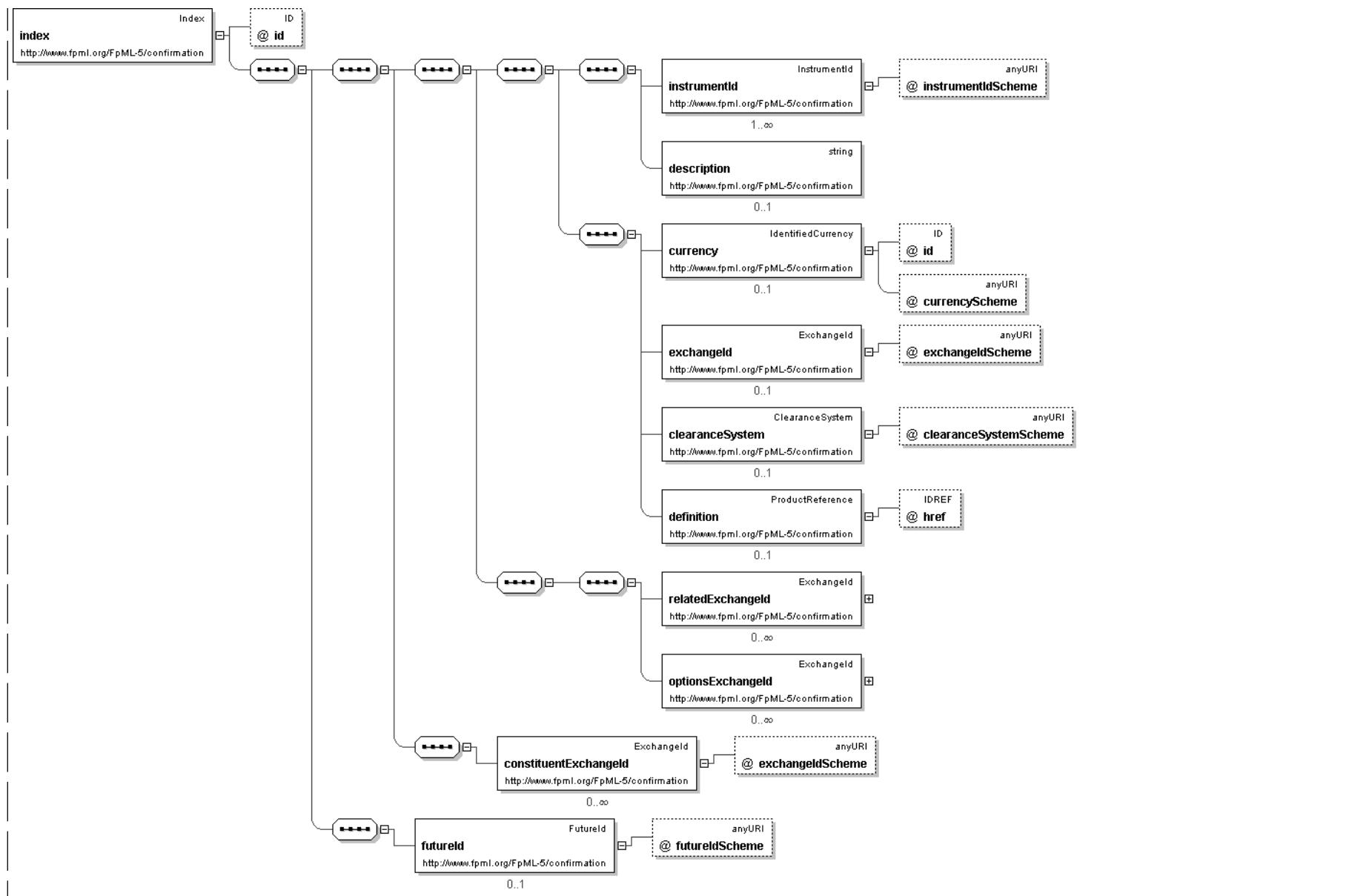
top

**Element: index**

- This element can be used wherever the following element is referenced:
  - underlyingAsset

<b>Name</b>	index
<b>Type</b>	<a href="#">Index</a>
<b>Nullable</b>	no
<b>Abstract</b>	no
<b>Documentation</b>	Identifies the underlying asset when it is a financial index.

**Logical Diagram**

**XML Instance Representation**

```

<index
  id=" xsd:ID [0..1]">
  <instrumentId> InstrumentId </instrumentId> [1..*]
  'Identification of the underlying asset, using public and/or private identifiers.'

  <description> xsd:string </description> [0..1]
  'Long name of the underlying asset.'

  <currency> IdentifiedCurrency </currency> [0..1]
  'Trading currency of the underlyer when transacted as a cash instrument.'

  <exchangeId> ExchangeId </exchangeId> [0..1]
  
```

'Identification of the exchange on which this asset is transacted for the purposes of calculating a contractual payoff. The term \"Exchange\" is assumed to have the meaning as defined in the ISDA 2002 Equity Derivatives Definitions.'

<clearanceSystem> ClearanceSystem </clearanceSystem> [0..1]

'Identification of the clearance system associated with the transaction exchange.'

<definition> ProductReference </definition> [0..1]

'An optional reference to a full FpML product that defines the simple product in greater detail. In case of inconsistency between the terms of the simple product and those of the detailed definition, the values in the simple product override those in the detailed definition.'

<relatedExchangeId> ExchangeId </relatedExchangeId> [0..\*]

'A short form unique identifier for a related exchange. If the element is not present then the exchange shall be the primary exchange on which listed futures and options on the underlying are listed. The term \"Exchange\" is assumed to have the meaning as defined in the ISDA 2002 Equity Derivatives Definitions.'

<optionsExchangeId> ExchangeId </optionsExchangeId> [0..\*]

'A short form unique identifier for an exchange on which the reference option contract is listed. This is to address the case where the reference exchange for the future is different than the one for the option. The options Exchange is referenced on share options when Merger Elections are selected as Options Exchange Adjustment.'

<constituentExchangeId> ExchangeId </constituentExchangeId> [0..\*]

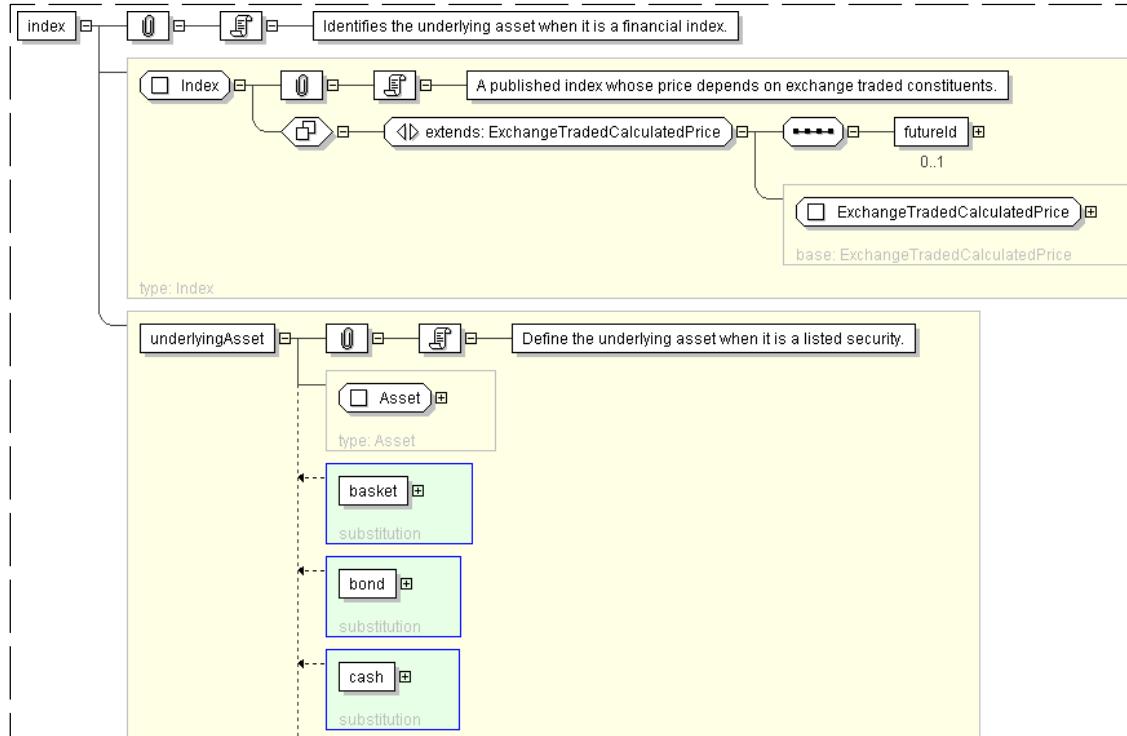
'Identification of all the exchanges where constituents are traded. The term \"Exchange\" is assumed to have the meaning as defined in the ISDA 2002 Equity Derivatives Definitions.'

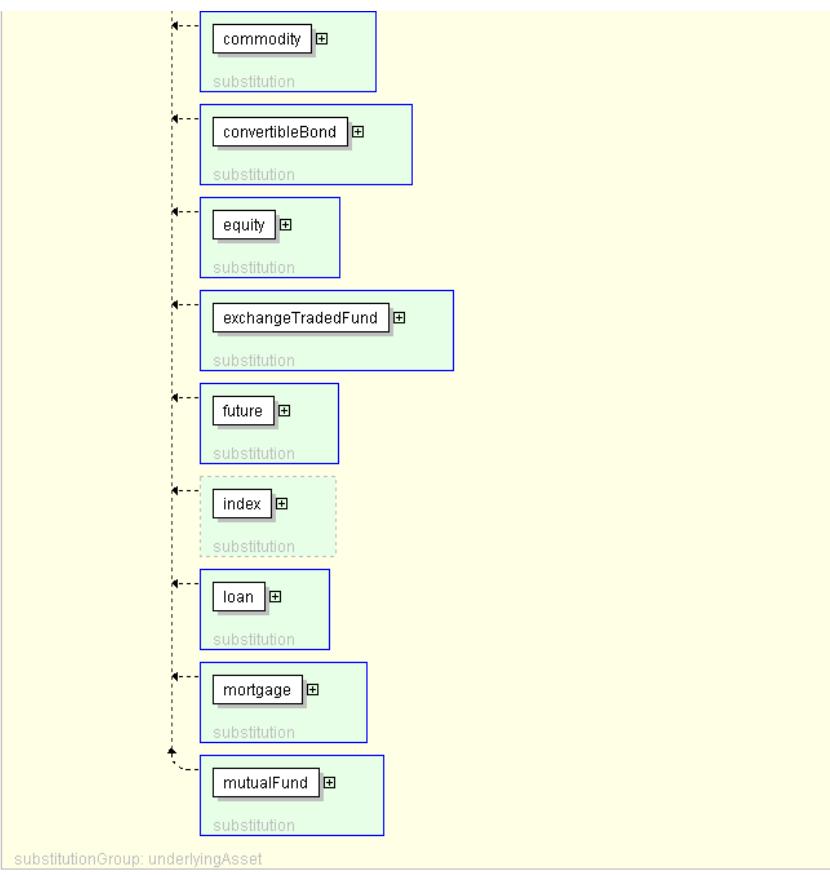
<futureId> FutureId </futureId> [0..1]

'A short form unique identifier for the reference future contract in the case of an index underlyer.'

</index>

#### Diagram



**Schema Component Representation**

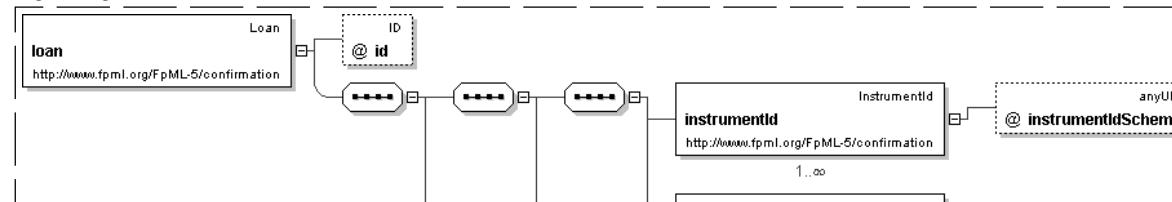
```
<xsd:element name="index" type=" Index " substitutionGroup="underlyingAsset"/>
```

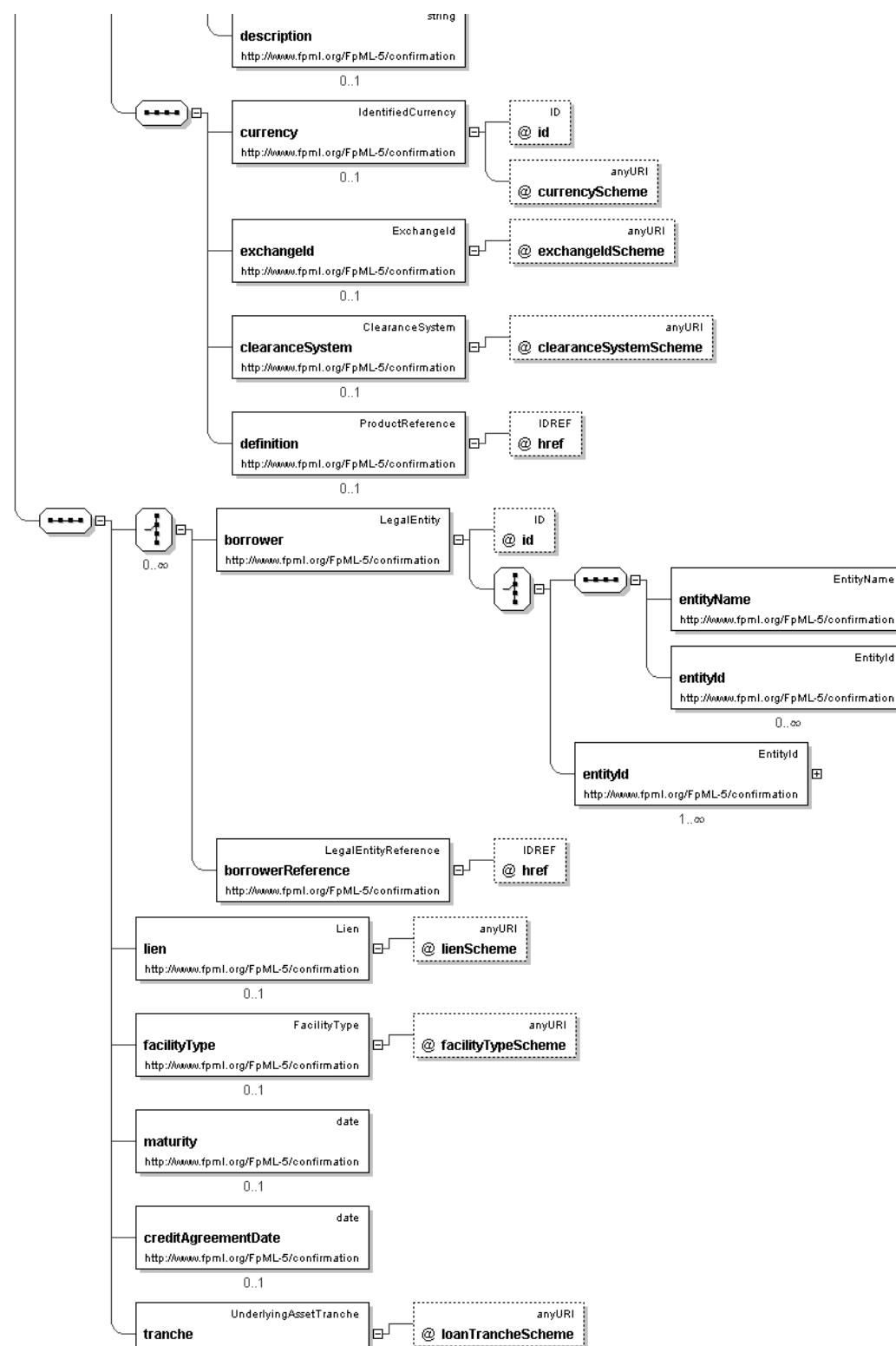
top

**Element: [loan](#)**

- This element can be used wherever the following element is referenced:
  - [underlyingAsset](#)

Name	loan
Type	<a href="#">Loan</a>
Nillable	no
Abstract	no
Documentation	Identifies a simple underlying asset that is a loan.

**Logical Diagram**



**XML Instance Representation**

```

<loan
  id=" xsd:ID [0..1]">
  <instrumentId> InstrumentId </instrumentId> [1..*]
  'Identification of the underlying asset, using public and/or private identifiers.'

  <description> xsd:string </description> [0..1]
  'Long name of the underlying asset.'

  <currency> IdentifiedCurrency </currency> [0..1]
  'Trading currency of the underlyer when transacted as a cash instrument.'

  <exchangeId> ExchangeId </exchangeId> [0..1]
  'Identification of the exchange on which this asset is transacted for the purposes
  of calculating a contractual payoff. The term \"Exchange\" is assumed to have the meaning
  as defined in the ISDA 2002 Equity Derivatives Definitions.'

  <clearanceSystem> ClearanceSystem </clearanceSystem> [0..1]
  'Identification of the clearance system associated with the transaction exchange.'

  <definition> ProductReference </definition> [0..1]
  'An optional reference to a full FpML product that defines the simple product in
  greater detail. In case of inconsistency between the terms of the simple product and those
  of the detailed definition, the values in the simple product override those in the
  detailed definition.'

Start Choice [0..*]
'Specifies the borrower. There can be more than one borrower. It is meant to be used in
the event that there is no Bloomberg Id or the Secured List isn't applicable.'

  <borrower> LegalEntity </borrower> [1]
  <borrowerReference> LegalEntityReference </borrowerReference> [1]
End Choice
  <lien> Lien </lien> [0..1]
  'Specifies the seniority level of the lien.'

  <facilityType> FacilityType </facilityType> [0..1]
  'The type of loan facility (letter of credit, revolving, ...).'

  <maturity> xsd:date </maturity> [0..1]
  'The date when the principal amount of the loan becomes due and payable.'

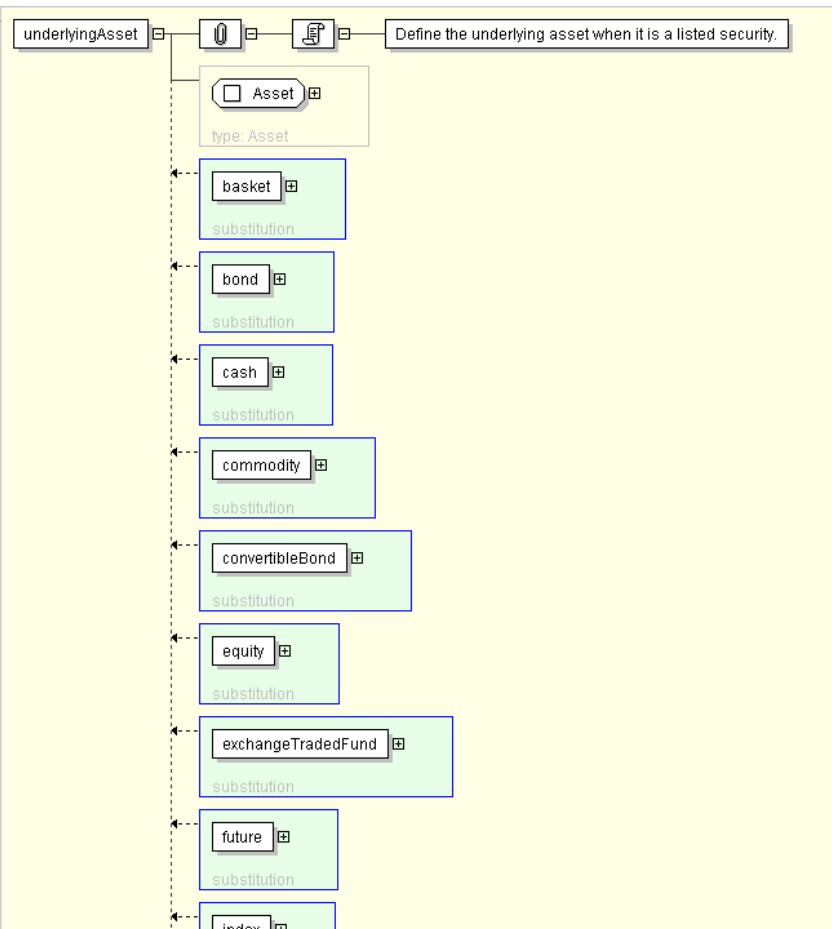
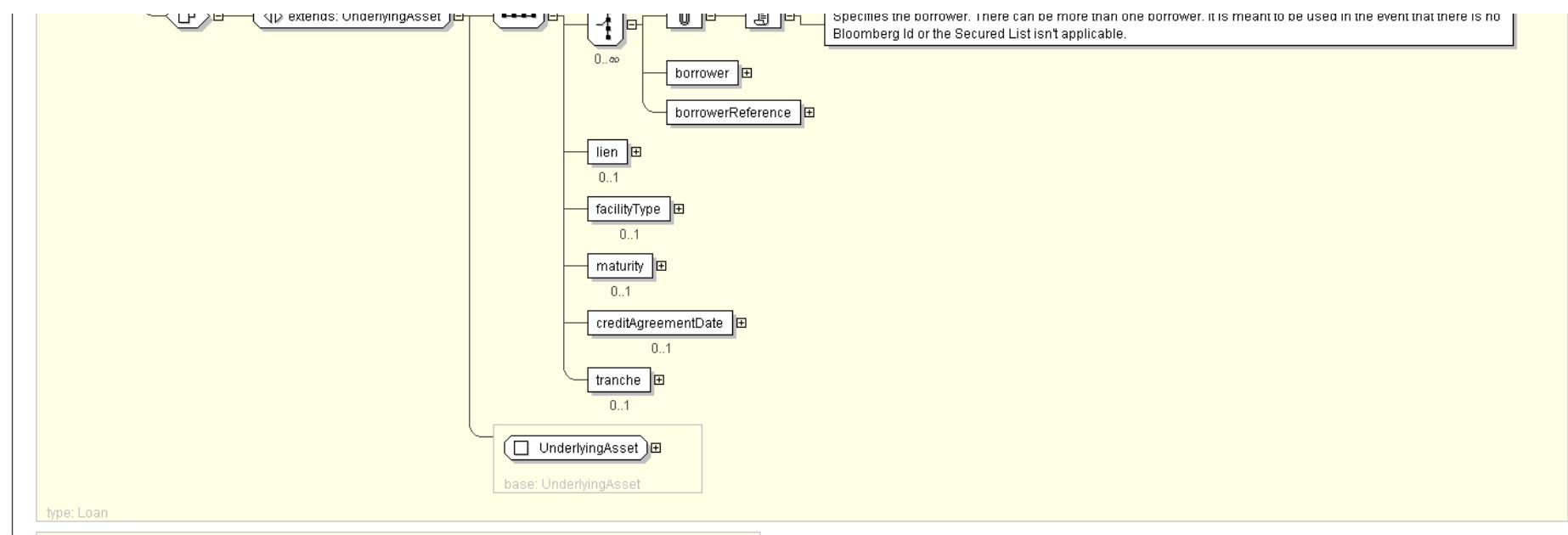
  <creditAgreementDate> xsd:date </creditAgreementDate> [0..1]
  'The credit agreement date is the closing date (the date where the agreement has been
  signed) for the loans in the credit agreement. Funding of the facilities occurs on
  (or sometimes a little after) the Credit Agreement date. This underlyer attribute is used
  to help identify which of the company\'s outstanding loans are being referenced by knowing
  to which credit agreement it belongs. ISDA Standards Terms Supplement term: Date of
  Original Credit Agreement.'

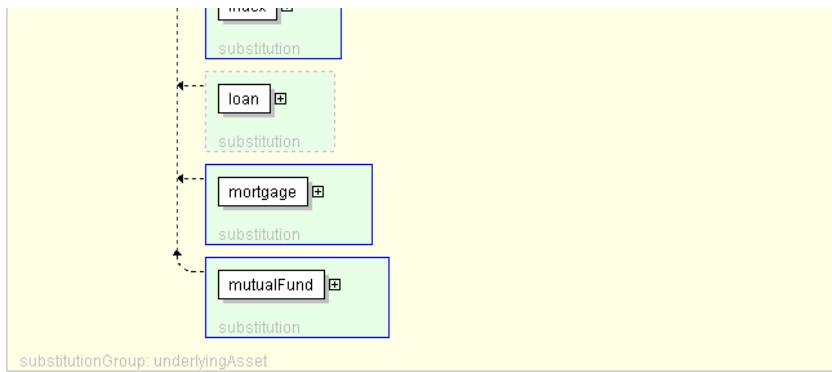
  <tranche> UnderlyingAssetTranche </tranche> [0..1]
  'The loan tranche that is subject to the derivative transaction. It will typically
  be referenced as the Bloomberg tranche number. ISDA Standards Terms Supplement term:
  Bloomberg Tranche Number.'

</loan>

```

**Diagram**



**Schema Component Representation**

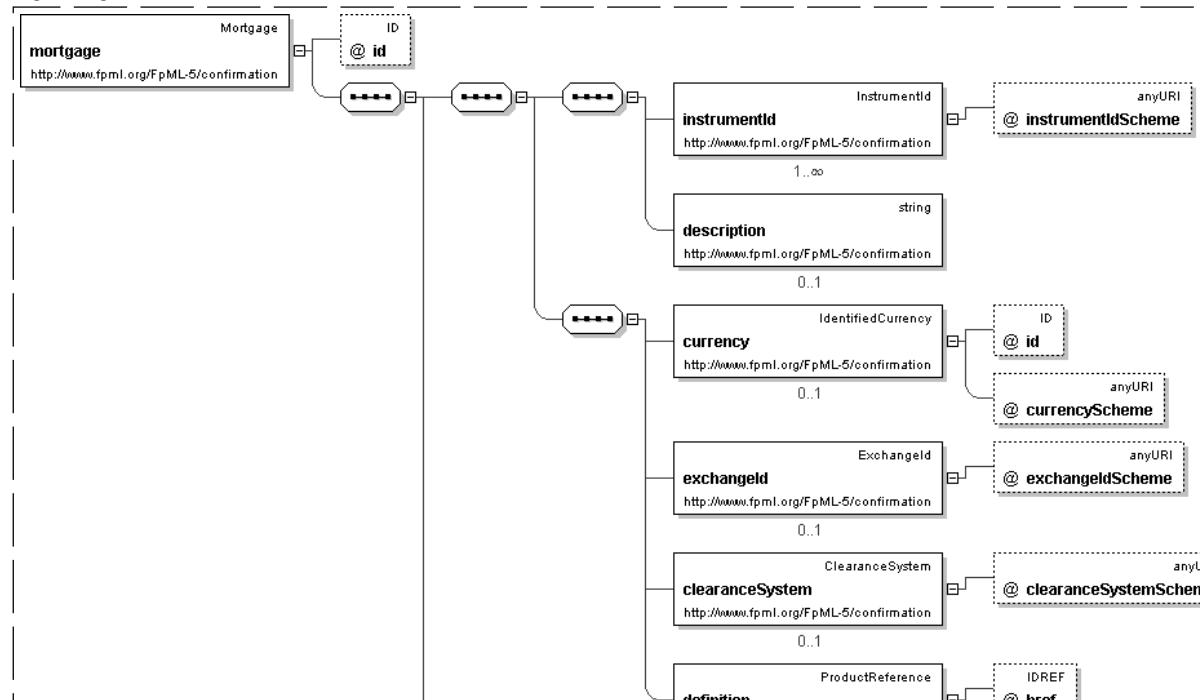
```
<xsd:element name="loan" type="Loan" substitutionGroup="underlyingAsset"/>
```

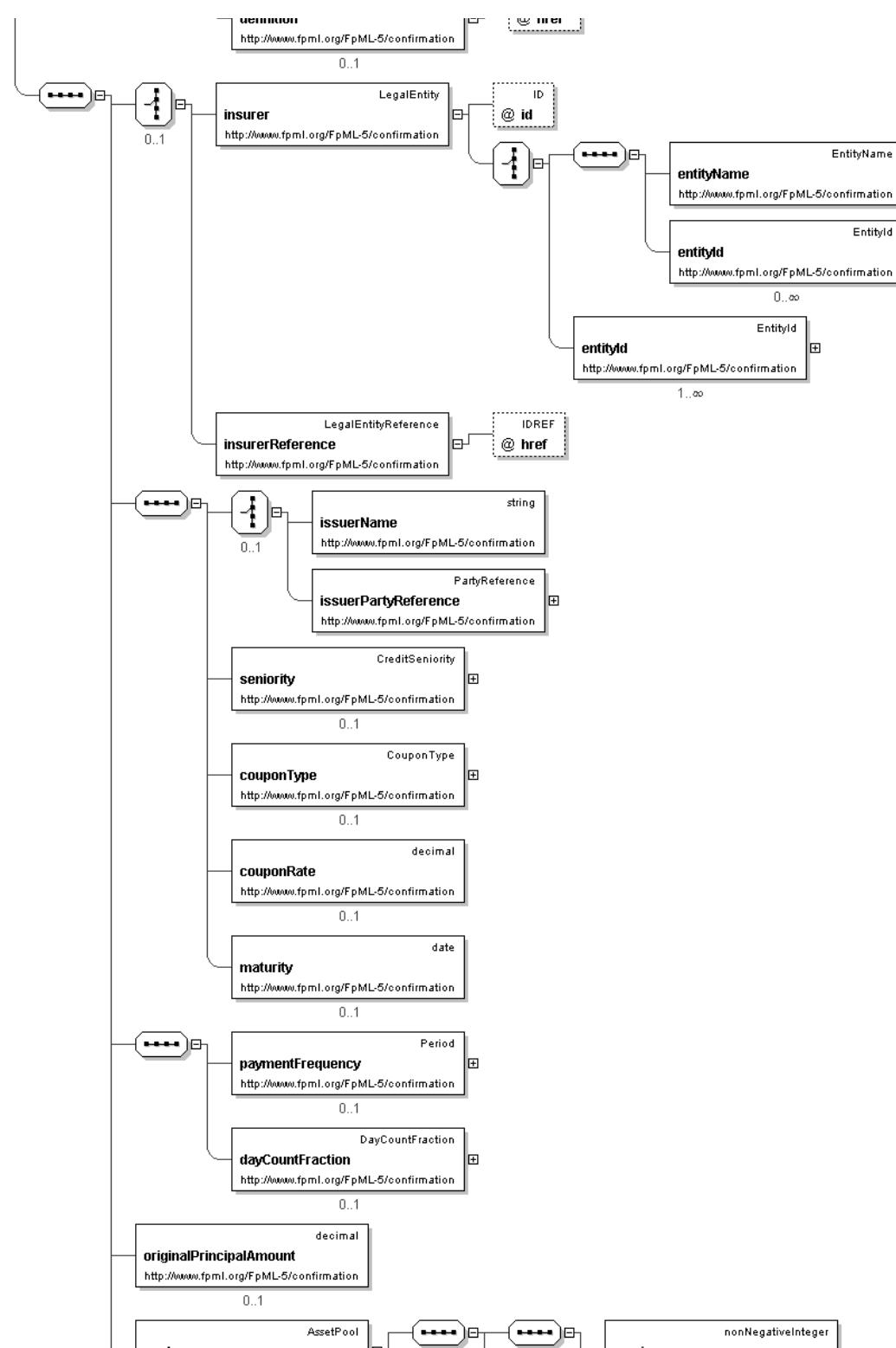
top

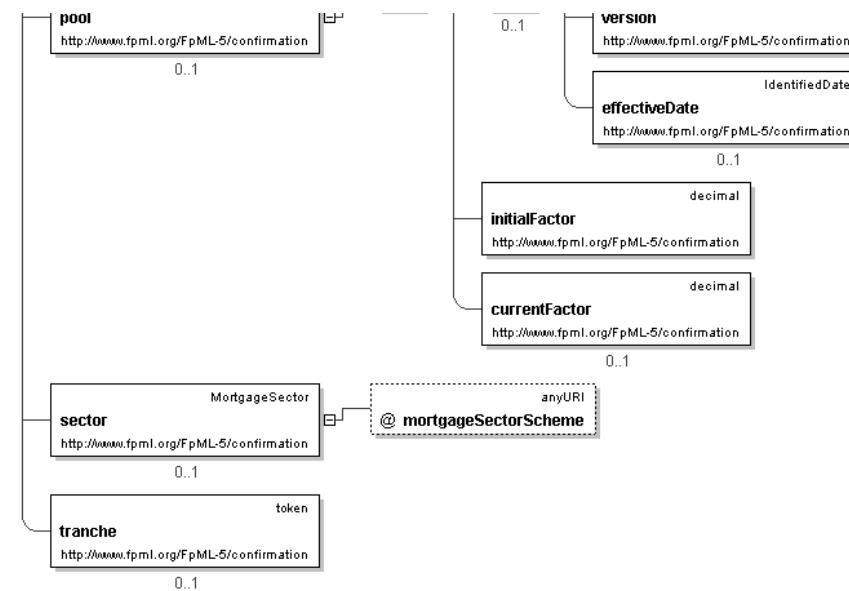
**Element: mortgage**

- This element can be used wherever the following element is referenced:
  - underlyingAsset

<b>Name</b>	mortgage
<b>Type</b>	Mortgage
<b>Nillable</b>	no
<b>Abstract</b>	no
<b>Documentation</b>	Identifies a mortgage backed security.

**Logical Diagram**



**XML Instance Representation**

```

<mortgage id="xsd:ID [0..1]">
  <instrumentId> InstrumentId </instrumentId> [1..*]
  'Identification of the underlying asset, using public and/or private identifiers.'

  <description> xsd:string </description> [0..1]
  'Long name of the underlying asset.'

  <currency> IdentifiedCurrency </currency> [0..1]
  'Trading currency of the underlyer when transacted as a cash instrument.'

  <exchangeId> ExchangeId </exchangeId> [0..1]
  'Identification of the exchange on which this asset is transacted for the purposes of calculating a contractual payoff. The term \"Exchange\" is assumed to have the meaning as defined in the ISDA 2002 Equity Derivatives Definitions.'

  <clearanceSystem> ClearanceSystem </clearanceSystem> [0..1]
  'Identification of the clearance system associated with the transaction exchange.'

  <definition> ProductReference </definition> [0..1]
  'An optional reference to a full FpML product that defines the simple product in greater detail. In case of inconsistency between the terms of the simple product and those of the detailed definition, the values in the simple product override those in the detailed definition.'

Start Choice [0..1]
'Applicable to the case of default swaps on MBS terms. For specifying the insurer name, when applicable (when the element is not present, it signifies that the insurer is Not Applicable)'

  <insurer> LegalEntity </insurer> [1]
  <insurerReference> LegalEntityReference </insurerReference> [1]
End Choice
Start Choice [0..1]
'Specifies the issuer name of a fixed income security or convertible bond. This name can either be explicitly stated, or specified as an href into another element of the document, such as the obligor.'
  
```

```

<issuerName> xsd:string </issuerName> [1]
<issuerPartyReference> PartyReference </issuerPartyReference> [1]
End Choice
<seniority> CreditSeniority </seniority> [0..1]
'The repayment precedence of a debt instrument.'

<couponType> CouponType </couponType> [0..1]
'Specifies if the bond has a variable coupon, step-up/down coupon or a zero-coupon.'

<couponRate> xsd:decimal </couponRate> [0..1]
'Specifies the coupon rate (expressed in percentage) of a fixed income security or convertible bond.'

<maturity> xsd:date </maturity> [0..1]
'The date when the principal amount of a security becomes due and payable.'

<paymentFrequency> Period </paymentFrequency> [0..1]
'Specifies the frequency at which the bond pays, e.g. 6M.'

<dayCountFraction> DayCountFraction </dayCountFraction> [0..1]
'The day count basis for the bond.'

<originalPrincipalAmount> xsd:decimal </originalPrincipalAmount> [0..1]
'The initial issued amount of the mortgage obligation.'

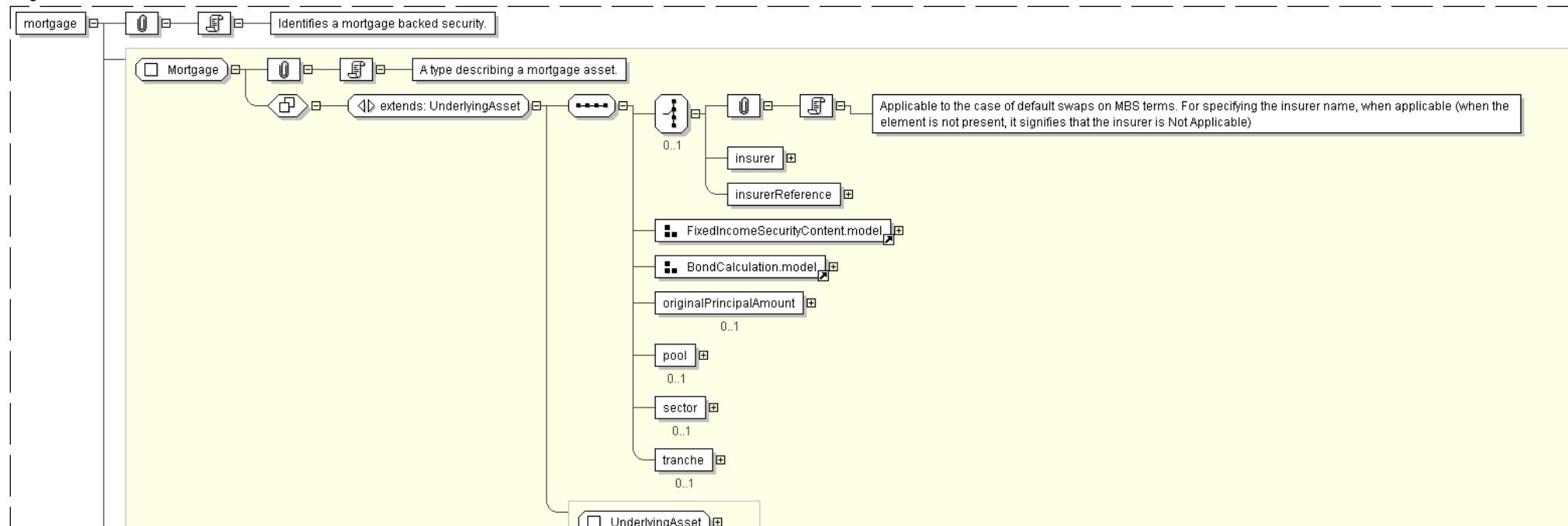
<pool> AssetPool </pool> [0..1]
'The mortgage pool that is underneath the mortgage obligation.'

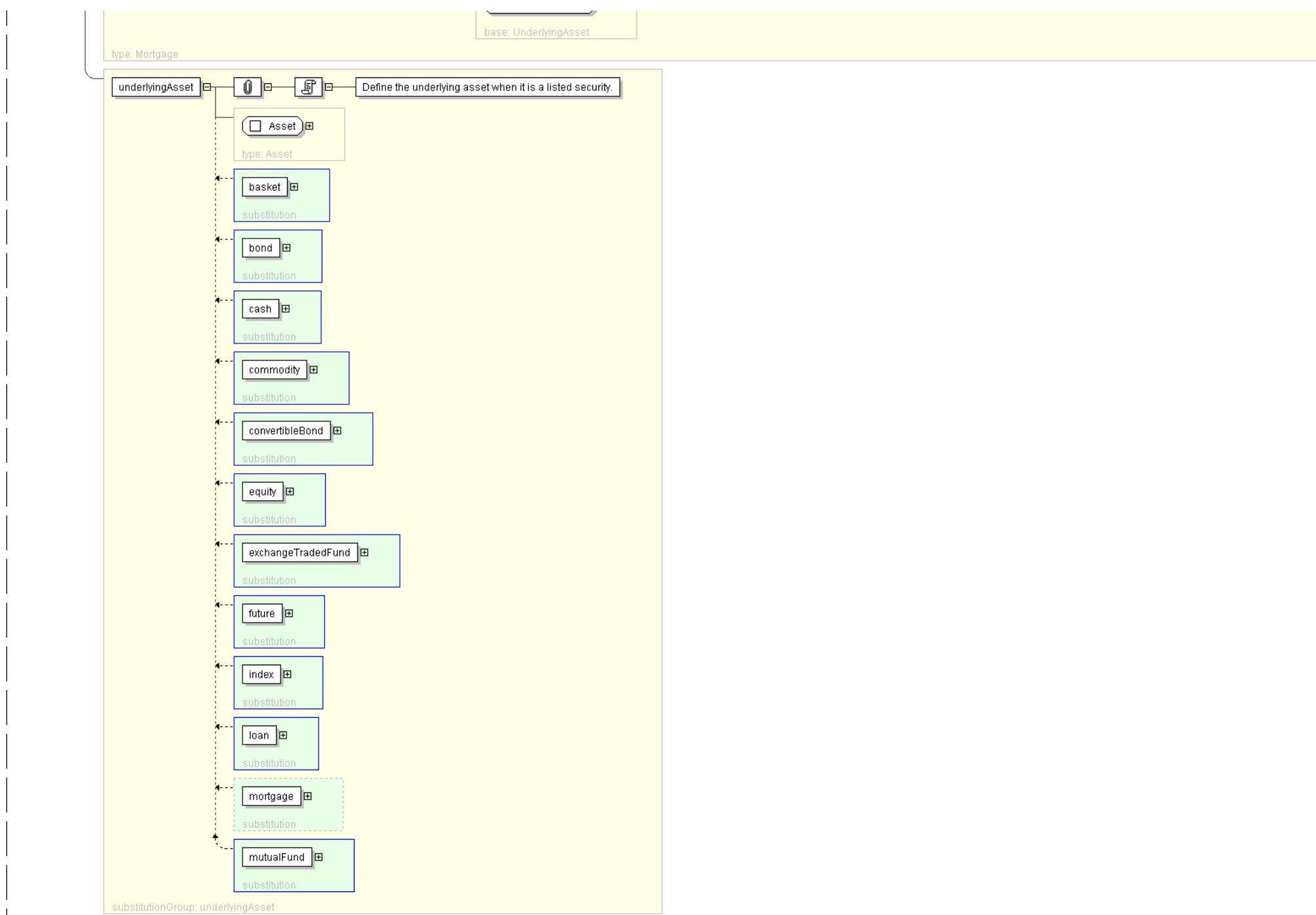
<sector> MortgageSector </sector> [0..1]
'The sector classification of the mortgage obligation.'

<tranche> xsd:token </tranche> [0..1]
'The mortgage obligation tranche that is subject to the derivative transaction.'

</mortgage>

```

**Diagram**

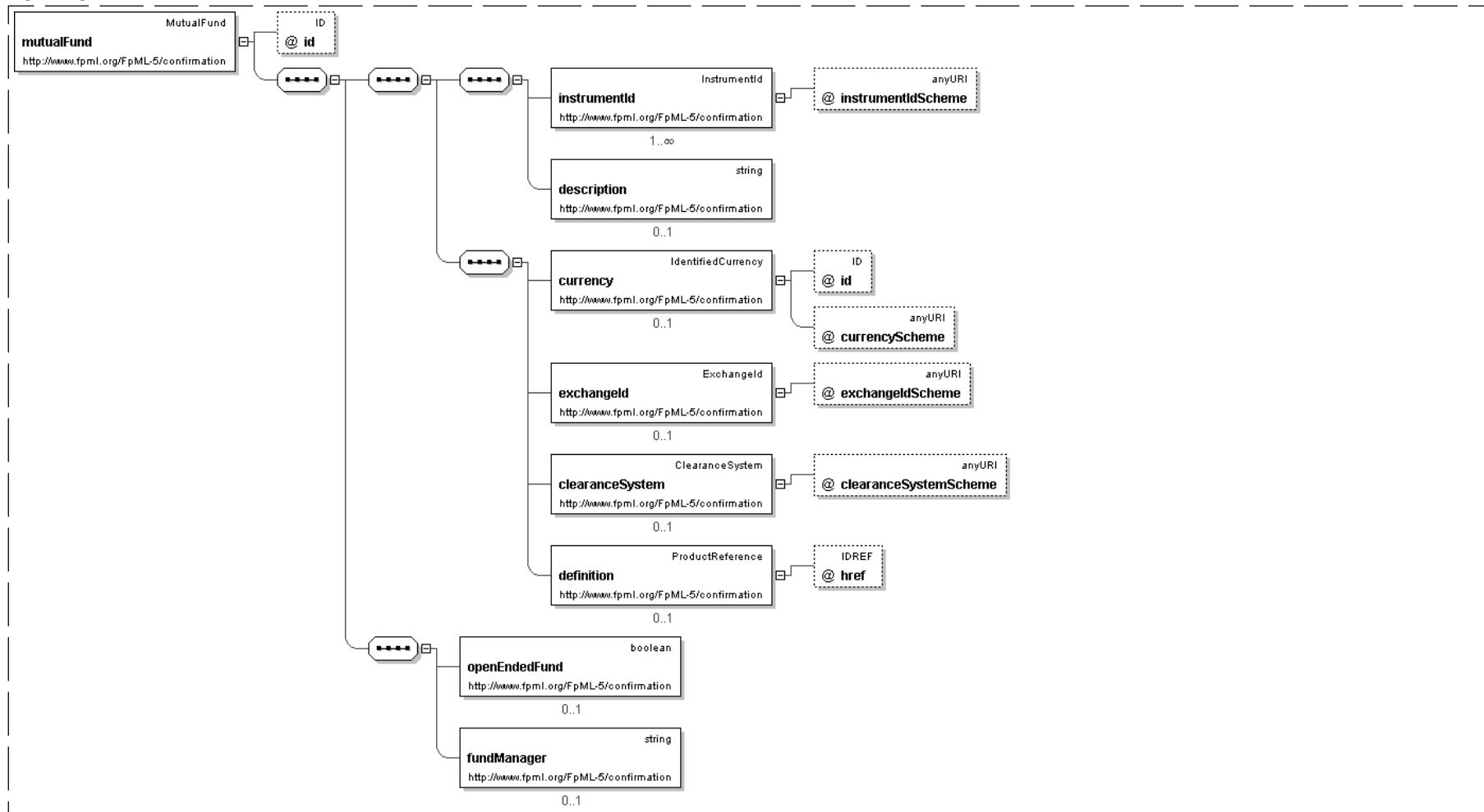
**Schema Component Representation**

```
<xsd:element name="mortgage" type="#Mortgage" substitutionGroup="underlyingAsset"/>
```

**Element: mutualFund**

- This element can be used wherever the following element is referenced:
  - [underlyingAsset](#)

Name	mutualFund
Type	<a href="#">MutualFund</a>
Nillable	no
Abstract	no
Documentation	Identifies the class of unit issued by a fund.

**Logical Diagram****XML Instance Representation**

```

<mutualFund
  id="xsd:ID [0..1]">
  <instrumentId> InstrumentId </instrumentId> [1..*]
  'Identification of the underlying asset, using public and/or private identifiers.'

  <description> xsd:string </description> [0..1]
  'Long name of the underlying asset.'
  
```

```

<currency> IdentifiedCurrency </currency> [0..1]
'Trading currency of the underlyer when transacted as a cash instrument.'

<exchangeId> ExchangeId </exchangeId> [0..1]
'Identification of the exchange on which this asset is transacted for the purposes
of calculating a contractual payoff. The term \"Exchange\" is assumed to have the meaning
as defined in the ISDA 2002 Equity Derivatives Definitions.'

<clearanceSystem> ClearanceSystem </clearanceSystem> [0..1]
'Identification of the clearance system associated with the transaction exchange.'

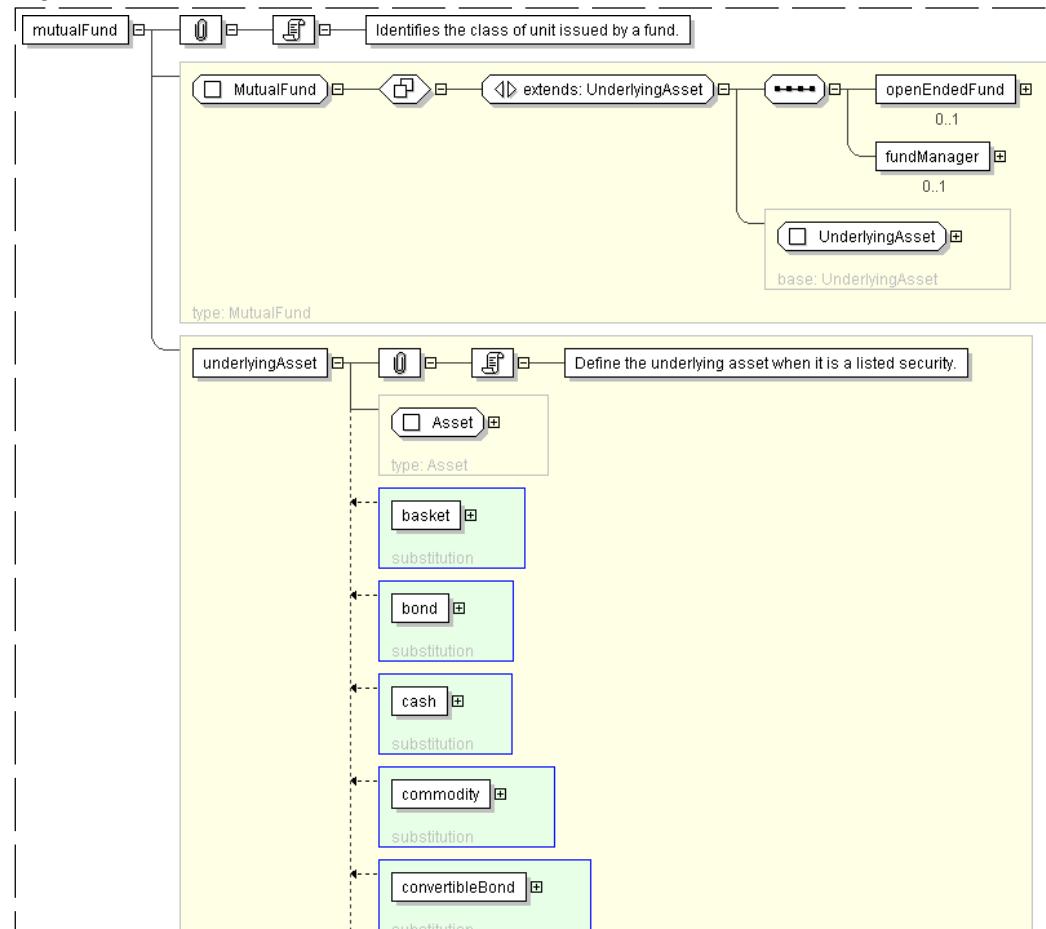
<definition> ProductReference </definition> [0..1]
'An optional reference to a full FpML product that defines the simple product in
greater detail. In case of inconsistency between the terms of the simple product and those
of the detailed definition, the values in the simple product override those in the
detailed definition.'

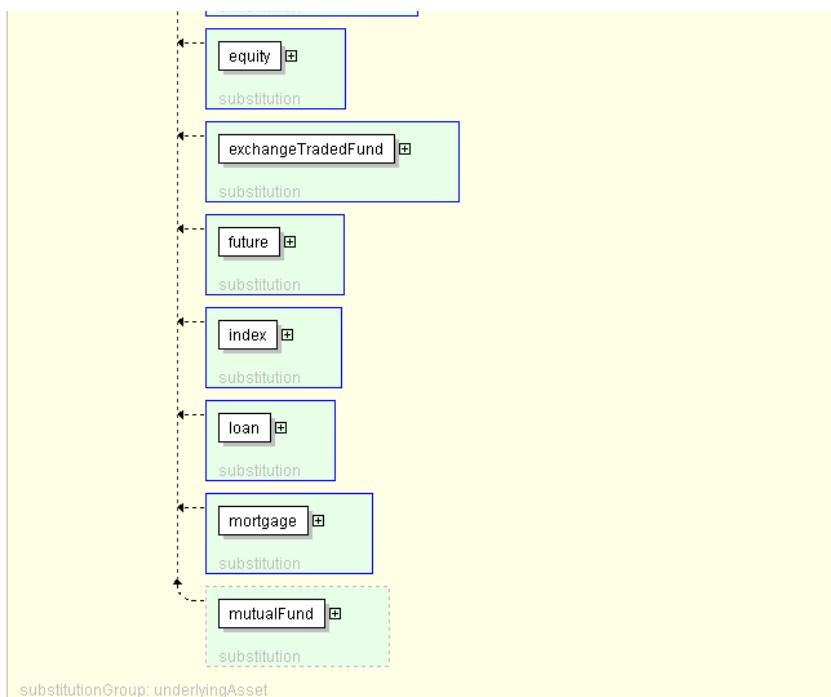
<openEndedFund> xsd:boolean </openEndedFund> [0..1]
'Boolean indicator to specify whether the mutual fund is an open-ended mutual fund.'

<fundManager> xsd:string </fundManager> [0..1]
'Specifies the fund manager that is in charge of the fund.'

</mutualFund>

```

**Diagram**

**Schema Component Representation**

```
<xsd:element name="mutualFund" type=" MutualFund " substitutionGroup="underlyingAsset" />
```

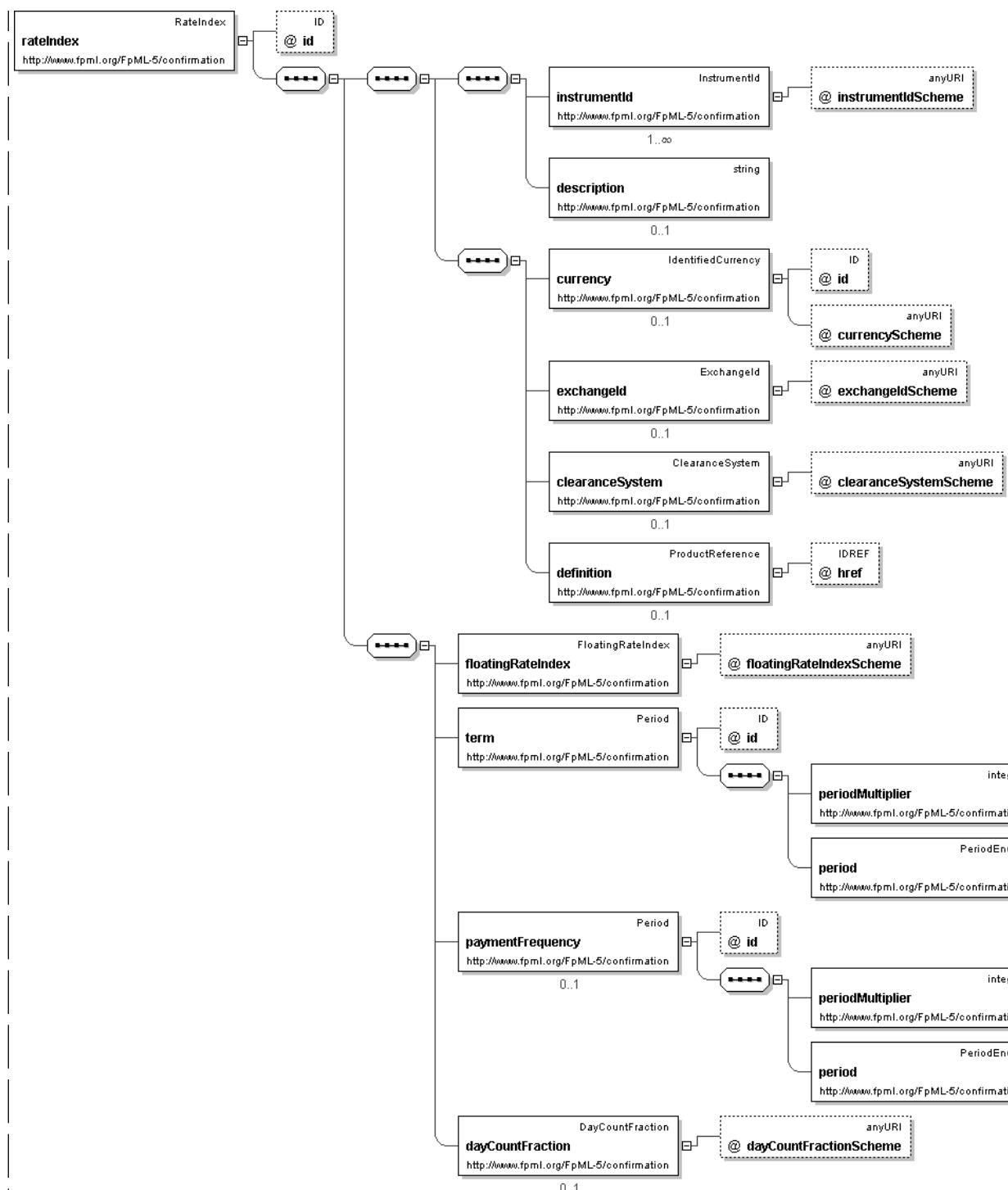
top

**Element: rateIndex**

- This element can be used wherever the following element is referenced:
  - curveInstrument

Name	rateIndex
Type	RateIndex
Nillable	no
Abstract	no
Documentation	Identifies a simple underlying asset that is an interest rate index. Used for specifying benchmark assets in the market environment in the pricing and risk model.

**Logical Diagram**

**XML Instance Representation**

&lt;rateIndex

```

id=" xsd:ID [0..1]">
<instrumentId> InstrumentId </instrumentId> [1..*]
'Identification of the underlying asset, using public and/or private identifiers.'

<description> xsd:string </description> [0..1]
'Long name of the underlying asset.'

<currency> IdentifiedCurrency </currency> [0..1]
'Trading currency of the underlyer when transacted as a cash instrument.'

<exchangeId> ExchangeId </exchangeId> [0..1]
'Identification of the exchange on which this asset is transacted for the purposes
of calculating a contractual payoff. The term "Exchange" is assumed to have the meaning
as defined in the ISDA 2002 Equity Derivatives Definitions.'

<clearanceSystem> ClearanceSystem </clearanceSystem> [0..1]
'Identification of the clearance system associated with the transaction exchange.'

<definition> ProductReference </definition> [0..1]
'An optional reference to a full FpML product that defines the simple product in
greater detail. In case of inconsistency between the terms of the simple product and those
of the detailed definition, the values in the simple product override those in the
detailed definition.'

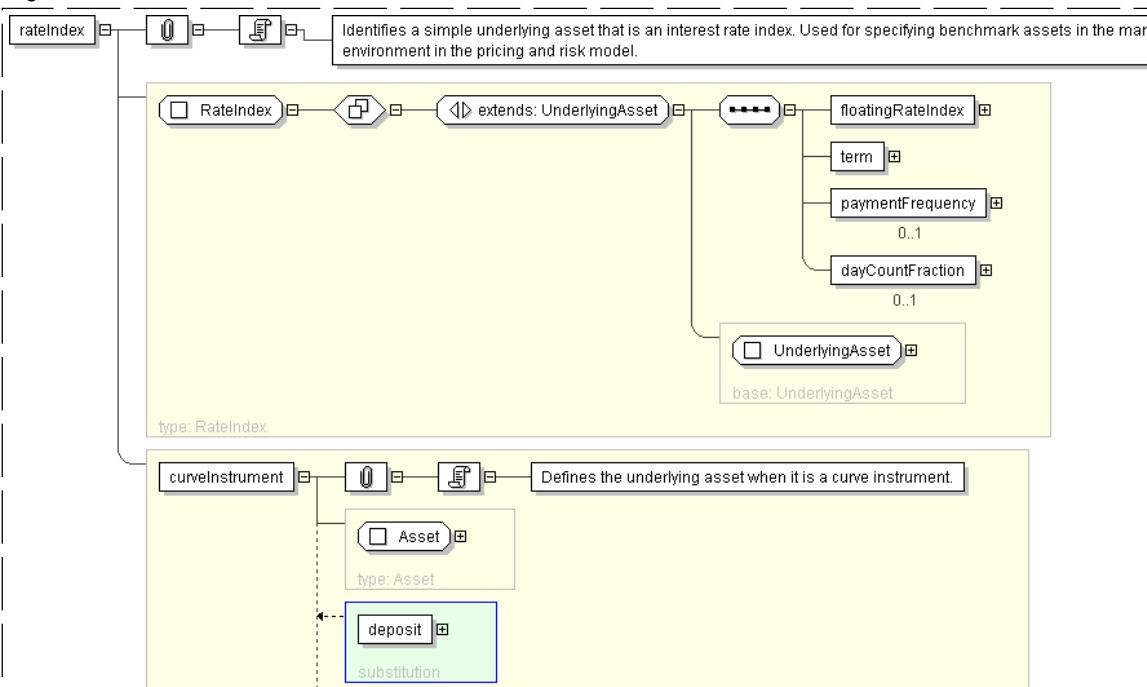
<floatingRateIndex> FloatingRateIndex </floatingRateIndex> [1]
<term> Period </term> [1]
'Specifies the term of the simple swap, e.g. 5Y.'

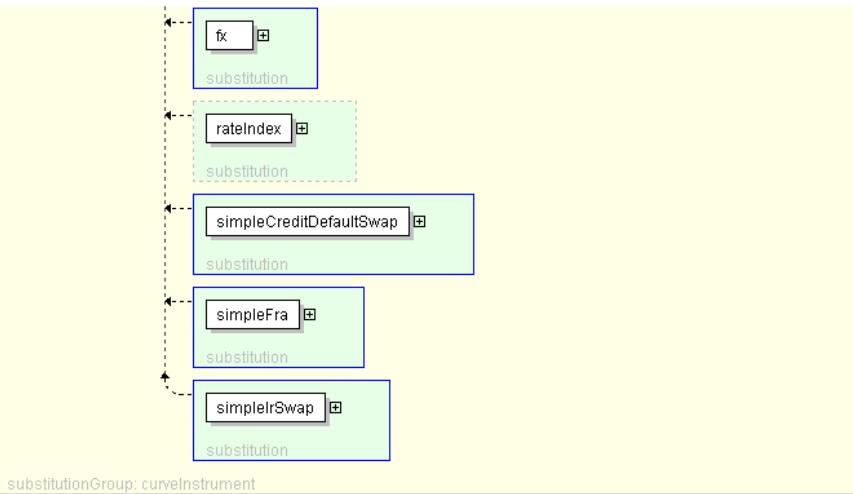
<paymentFrequency> Period </paymentFrequency> [0..1]
'Specifies the frequency at which the index pays, e.g. 6M.'

<dayCountFraction> DayCountFraction </dayCountFraction> [0..1]
'The day count basis for the index.'

</rateIndex>

```

**Diagram**



## Schema Component Representation

```
<xsd:element name="rateIndex" type="RateIndex" substitutionGroup="curveInstrument"
```

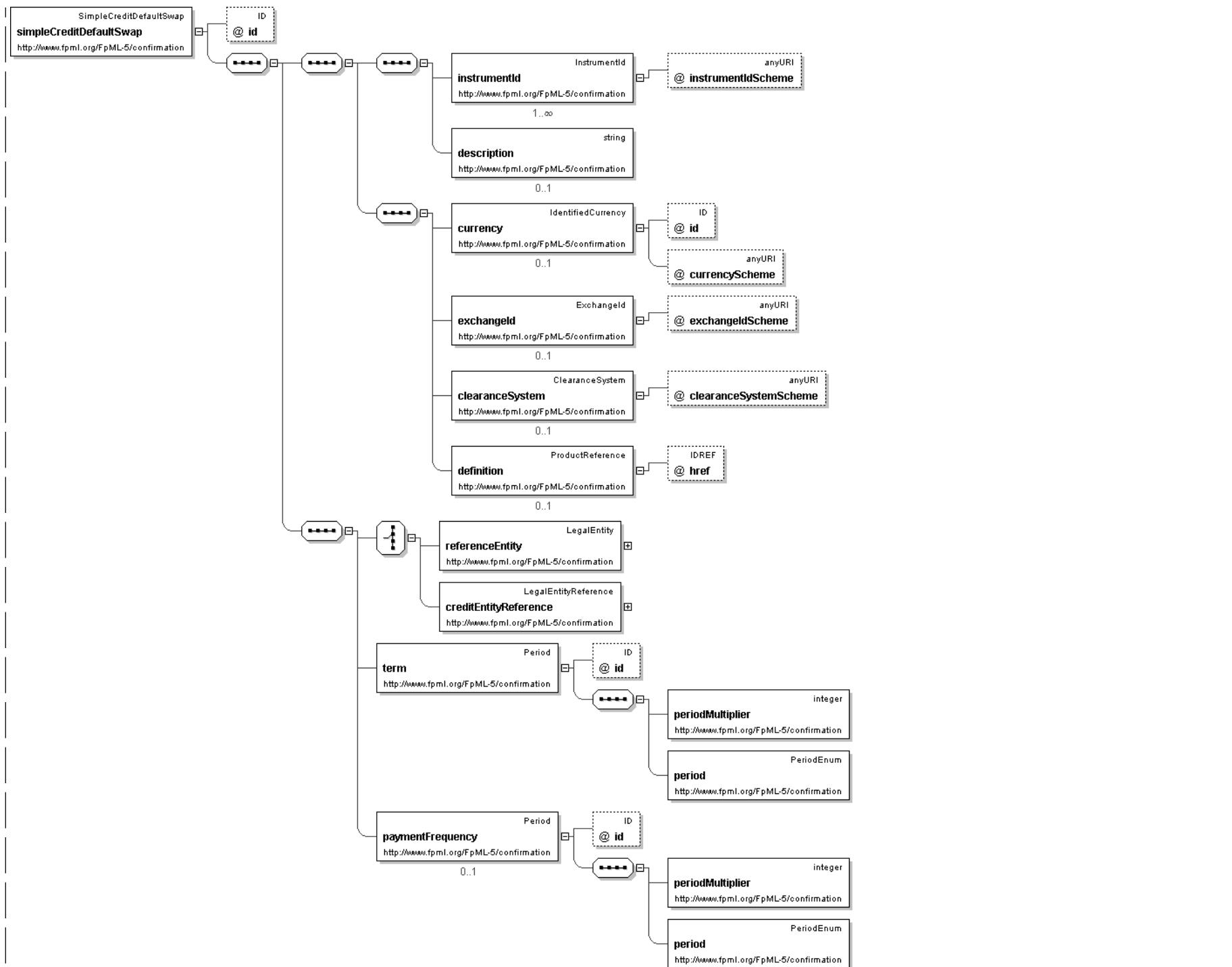
top

## Element: simpleCreditDefaultSwap

- This element can be used wherever the following element is referenced
    - [curveInstrument](#)

<b>Name</b>	simpleCreditDefaultSwap
<b>Type</b>	<a href="#">SimpleCreditDefaultSwap</a>
<b>Nillable</b>	no
<b>Abstract</b>	no
<b>Documentation</b>	Identifies a simple underlying asset that is a credit default swap.

## Logical Diagram

**XML Instance Representation**

```
<simpleCreditDefaultSwap
  id=" xsd:ID [0..1]">
```

```

<instrumentId> InstrumentId </instrumentId> [1..*]
  'Identification of the underlying asset, using public and/or private identifiers.'

<description> xsd:string </description> [0..1]
  'Long name of the underlying asset.'

<currency> IdentifiedCurrency </currency> [0..1]
  'Trading currency of the underlyer when transacted as a cash instrument.'

<exchangeId> ExchangeId </exchangeId> [0..1]
  'Identification of the exchange on which this asset is transacted for the purposes
  of calculating a contractual payoff. The term "Exchange" is assumed to have the meaning
  as defined in the ISDA 2002 Equity Derivatives Definitions.'

<clearanceSystem> ClearanceSystem </clearanceSystem> [0..1]
  'Identification of the clearance system associated with the transaction exchange.'

<definition> ProductReference </definition> [0..1]
  'An optional reference to a full FpML product that defines the simple product in
  greater detail. In case of inconsistency between the terms of the simple product and those
  of the detailed definition, the values in the simple product override those in the
  detailed definition.'

Start Choice [1]
  <referenceEntity> LegalEntity </referenceEntity> [1]
    'The entity for which this is defined.'

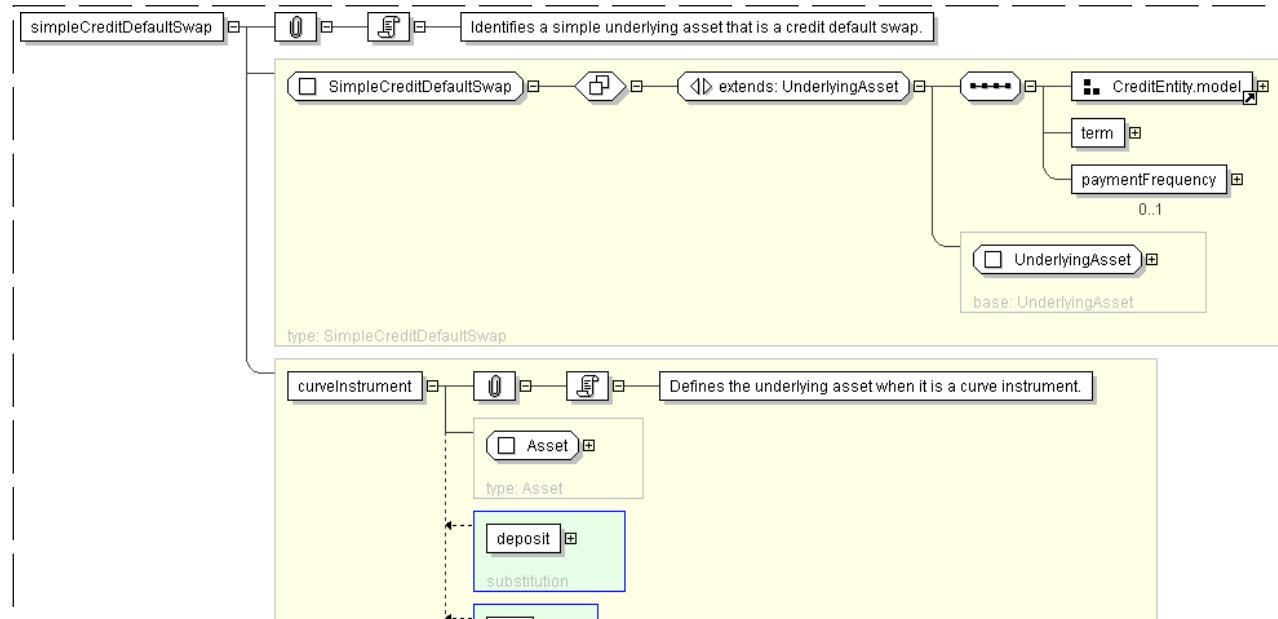
  <creditEntityReference> LegalEntityReference </creditEntityReference> [1]
    'An XML reference a credit entity defined elsewhere in the document.'

End Choice
  <term> Period </term> [1]
    'Specifies the term of the simple CD swap, e.g. 5Y.'

  <paymentFrequency> Period </paymentFrequency> [0..1]
    'Specifies the frequency at which the swap pays, e.g. 6M.'

</simpleCreditDefaultSwap>

```

**Diagram**

**Schema Component Representation**

```
<xsd:element name="simpleCreditDefaultSwap" type=" SimpleCreditDefaultSwap
  " substitutionGroup="curveInstrument" />
```

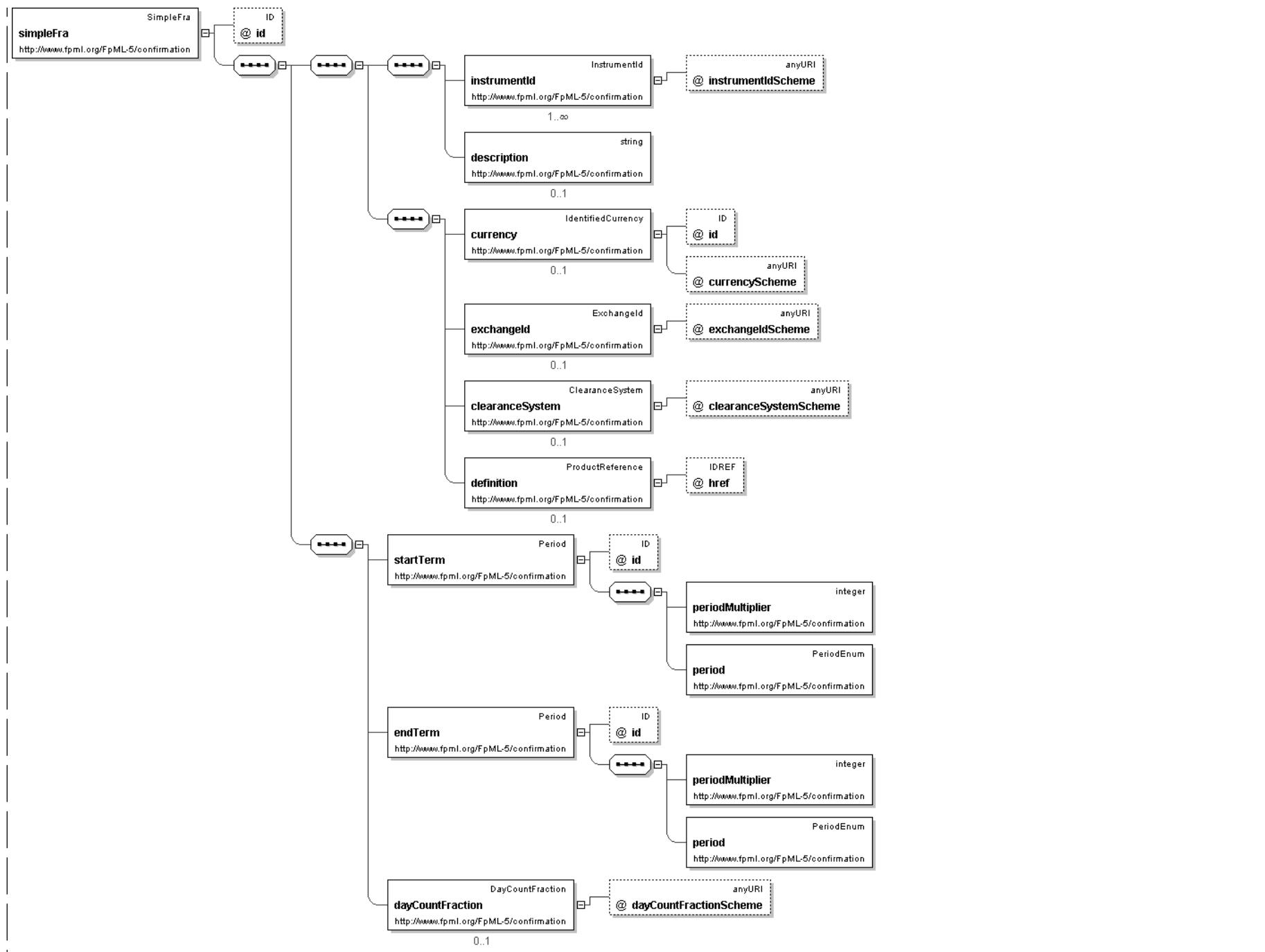
top

**Element: simpleFra**

- This element can be used wherever the following element is referenced:
  - [curveInstrument](#)

Name	simpleFra
Type	<a href="#">SimpleFra</a>
Nullable	no
Abstract	no
Documentation	Identifies a simple underlying asset that is a forward rate agreement.

**Logical Diagram**

**XML Instance Representation**

```
<simpleFra
  id=" xsd:ID [0..1]">
    <instrumentId> InstrumentId </instrumentId> [1..*]
    'Identification of the underlying asset, using public and/or private identifiers.'
```

```

<description> xsd:string </description> [0..1]
'Long name of the underlying asset.'

<currency> IdentifiedCurrency </currency> [0..1]
'Trading currency of the underlyer when transacted as a cash instrument.'

<exchangeId> ExchangeId </exchangeId> [0..1]
'Identification of the exchange on which this asset is transacted for the purposes
of calculating a contractual payoff. The term "Exchange" is assumed to have the meaning
as defined in the ISDA 2002 Equity Derivatives Definitions.'

<clearanceSystem> ClearanceSystem </clearanceSystem> [0..1]
'Identification of the clearance system associated with the transaction exchange.'

<definition> ProductReference </definition> [0..1]
'An optional reference to a full FpML product that defines the simple product in
greater detail. In case of inconsistency between the terms of the simple product and those
of the detailed definition, the values in the simple product override those in the
detailed definition.'

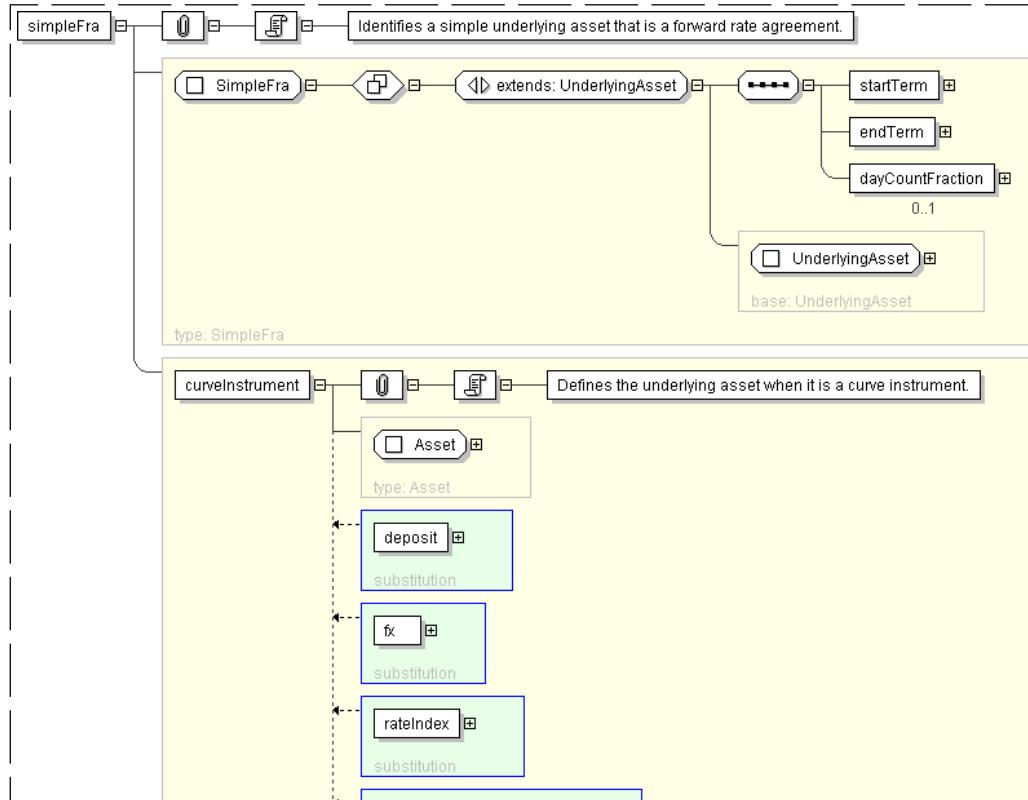
<startTerm> Period </startTerm> [1]
'Specifies the start term of the simple fra, e.g. 3M.'

<endTerm> Period </endTerm> [1]
'Specifies the end term of the simple fra, e.g. 9M.'

<dayCountFraction> DayCountFraction </dayCountFraction> [0..1]
'The day count basis for the FRA.'

</simpleFra>

```

**Diagram**

**Schema Component Representation**

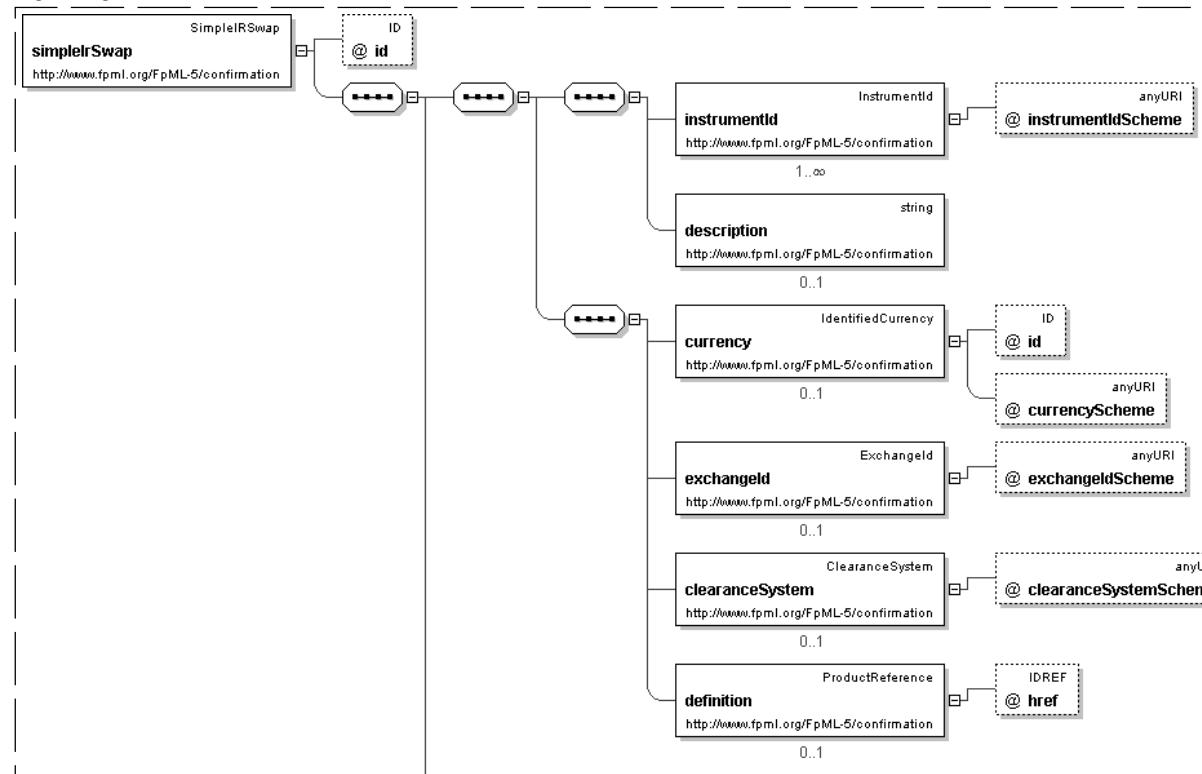
```
<xsd:element name="simpleFra" type=" SimpleFra " substitutionGroup="curveInstrument"/>
```

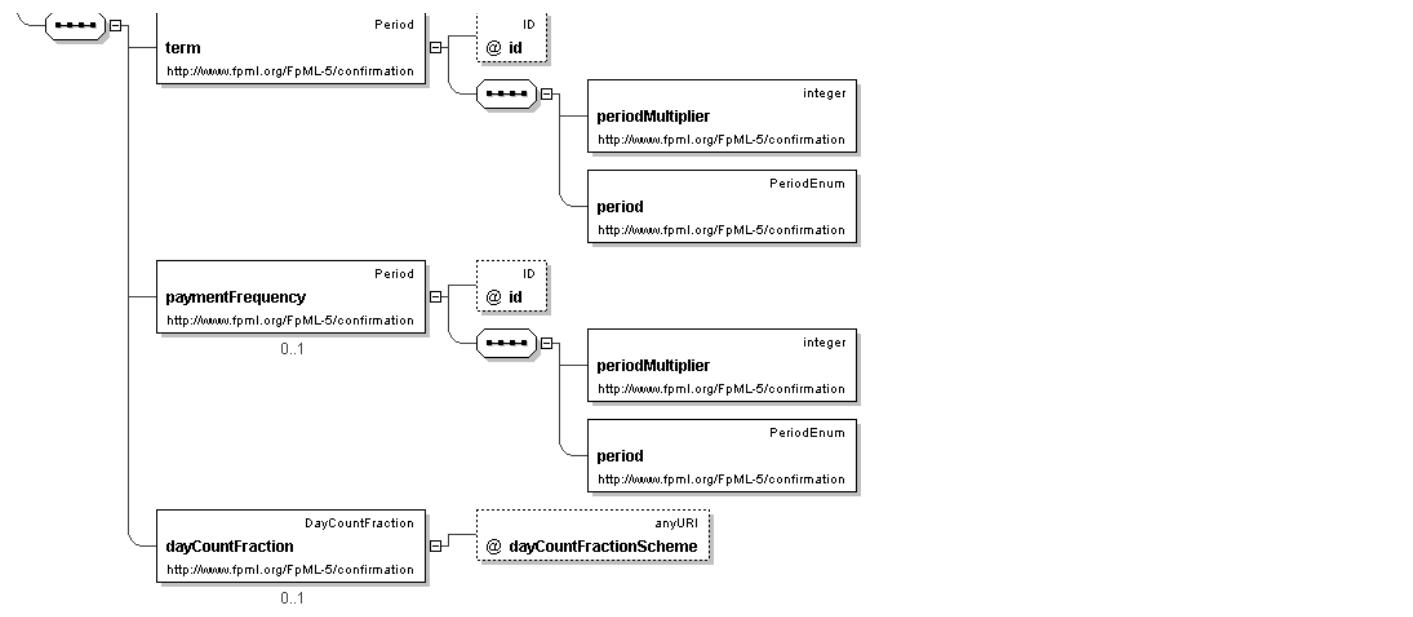
top

**Element: simpleIrSwap**

- This element can be used wherever the following element is referenced:
  - [curveInstrument](#)

Name	simpleIrSwap
Type	<a href="#">SimpleIRSwap</a>
Nullable	no
Abstract	no
Documentation	Identifies a simple underlying asset that is a swap.

**Logical Diagram**

**XML Instance Representation**

```

<simpleIrSwap
  id=" xsd:ID [0..1]">
  <instrumentId> InstrumentId </instrumentId> [1..*]
    'Identification of the underlying asset, using public and/or private identifiers.'

  <description> xsd:string </description> [0..1]
    'Long name of the underlying asset.'

  <currency> IdentifiedCurrency </currency> [0..1]
    'Trading currency of the underlyer when transacted as a cash instrument.'

  <exchangeId> ExchangeId </exchangeId> [0..1]
    'Identification of the exchange on which this asset is transacted for the purposes
    of calculating a contractual payoff. The term \"Exchange\" is assumed to have the meaning
    as defined in the ISDA 2002 Equity Derivatives Definitions.'

  <clearanceSystem> ClearanceSystem </clearanceSystem> [0..1]
    'Identification of the clearance system associated with the transaction exchange.'

  <definition> ProductReference </definition> [0..1]
    'An optional reference to a full FpML product that defines the simple product in
    greater detail. In case of inconsistency between the terms of the simple product and those
    of the detailed definition, the values in the simple product override those in the
    detailed definition.'

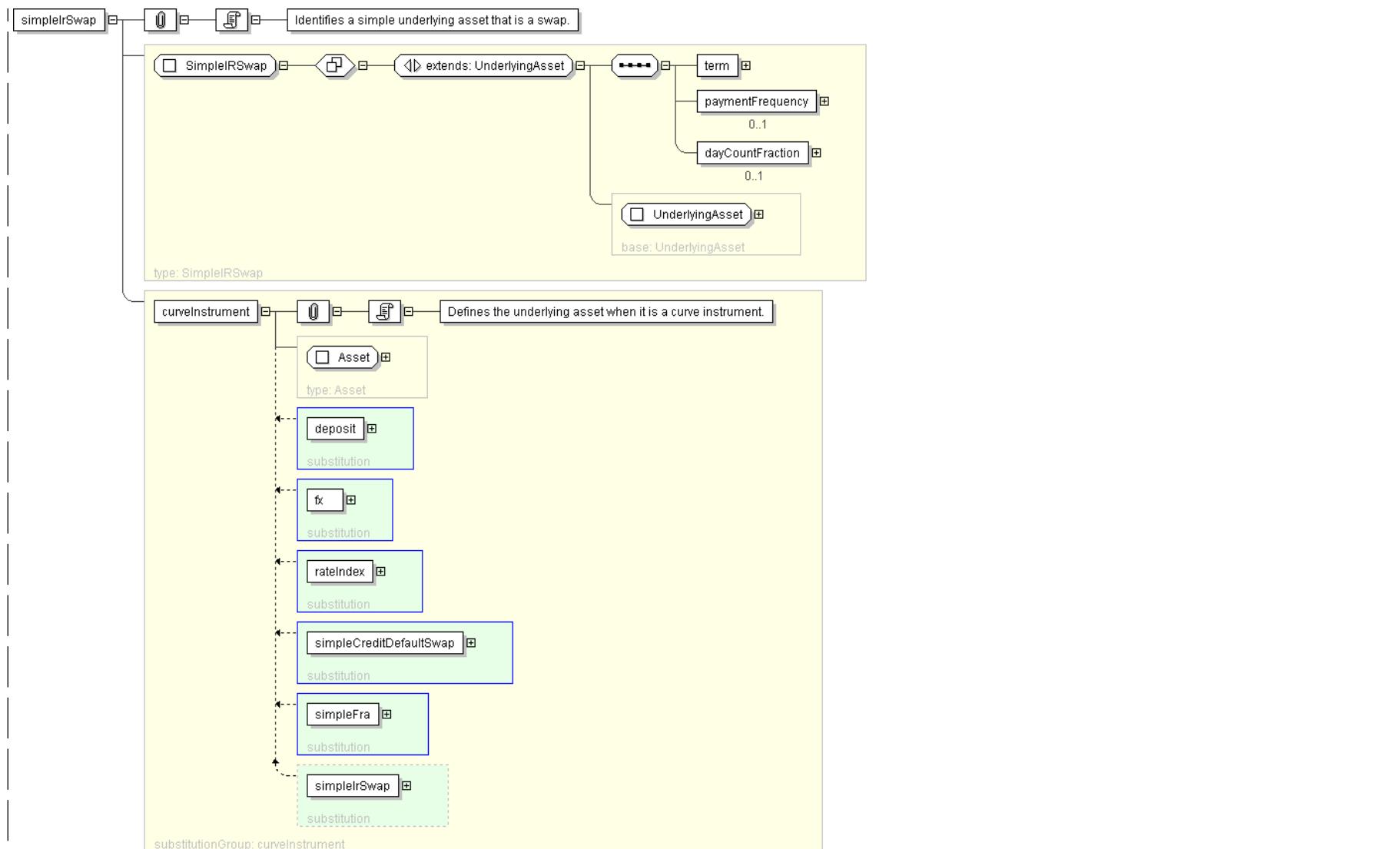
  <term> Period </term> [1]
    'Specifies the term of the simple swap, e.g. 5Y.'

  <paymentFrequency> Period </paymentFrequency> [0..1]
    'Specifies the frequency at which the swap pays, e.g. 6M.'

  <dayCountFraction> DayCountFraction </dayCountFraction> [0..1]
    'The day count basis for the swap.'

</simpleIrSwap>
  
```

**Diagram**

**Schema Component Representation**

```
<xsd:element name="simpleIRSwap" type="SimpleIRSwap" substitutionGroup="curveInstrument"/>
```

top

**Element: underlyingAsset**

- The following elements can be used wherever this element is referenced:

- [basket](#)
- [bond](#)
- [cash](#)
- [commodity](#)
- [convertibleBond](#)
- [equity](#)
- [exchangeTradedFund](#)
- [future](#)
- [index](#)
- [loan](#)
- [mortgage](#)
- [mutualFund](#)

Name	underlyingAsset
Used by (from the same schema document)	Complex Type <a href="#">BasketConstituent</a> , Complex Type <a href="#">SingleUnderlyer</a>
Type	Asset
Nillable	no
Abstract	yes
Documentation	Define the underlying asset when it is a listed security.

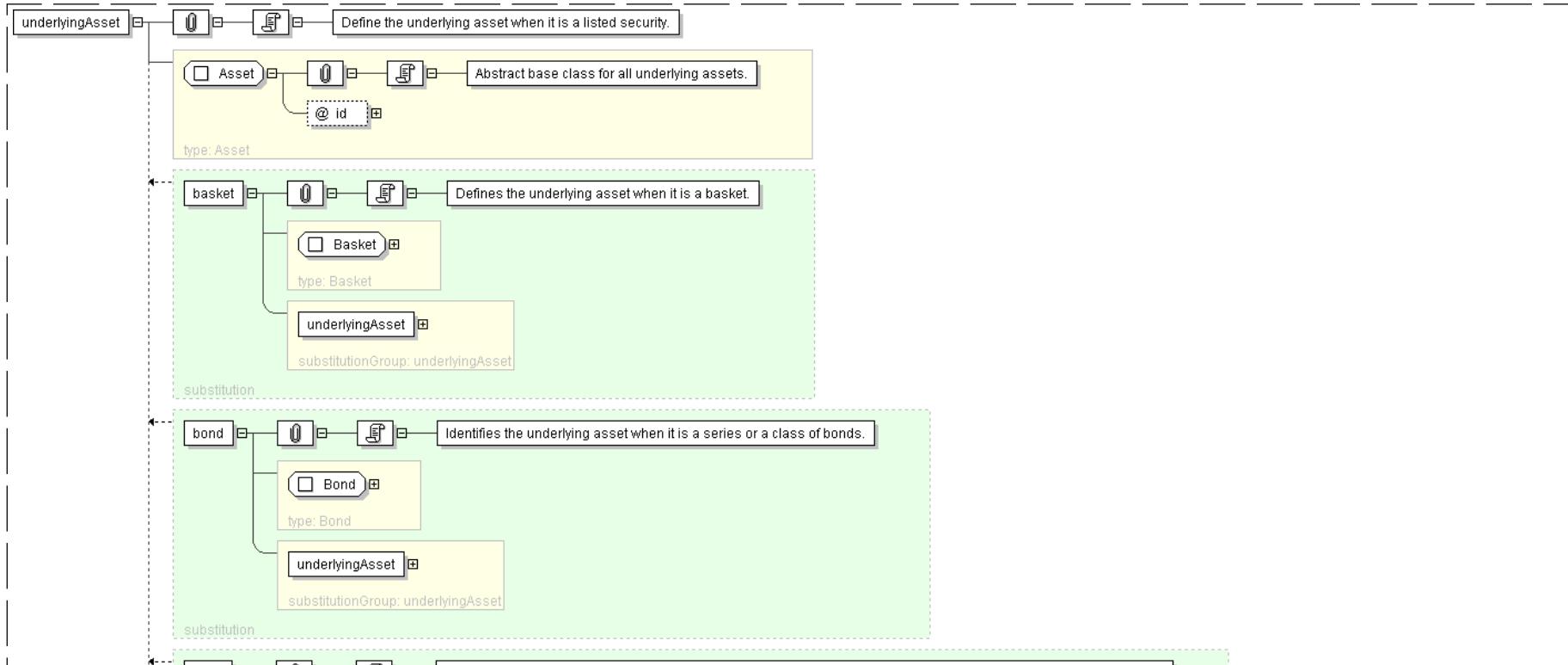
#### Logical Diagram

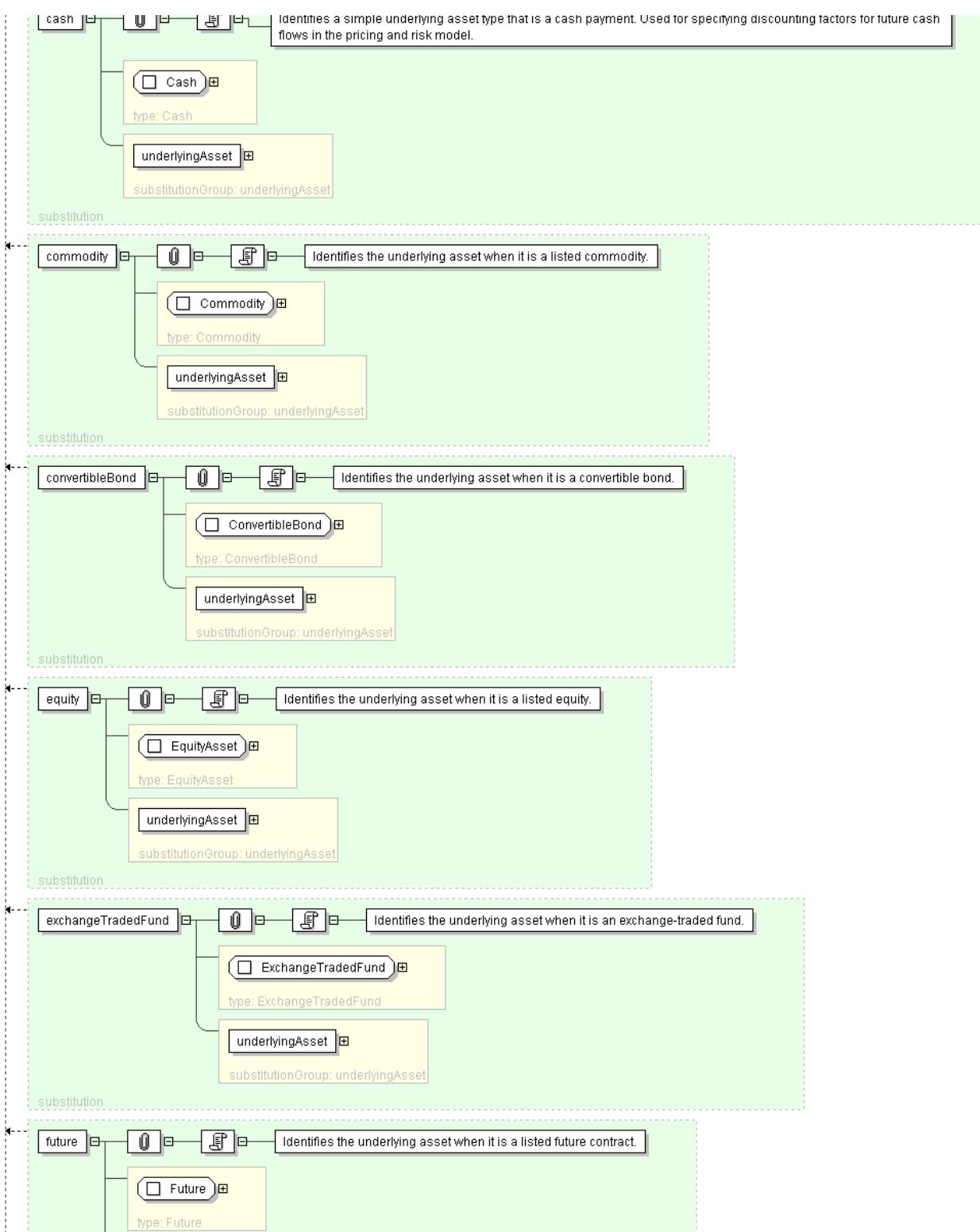


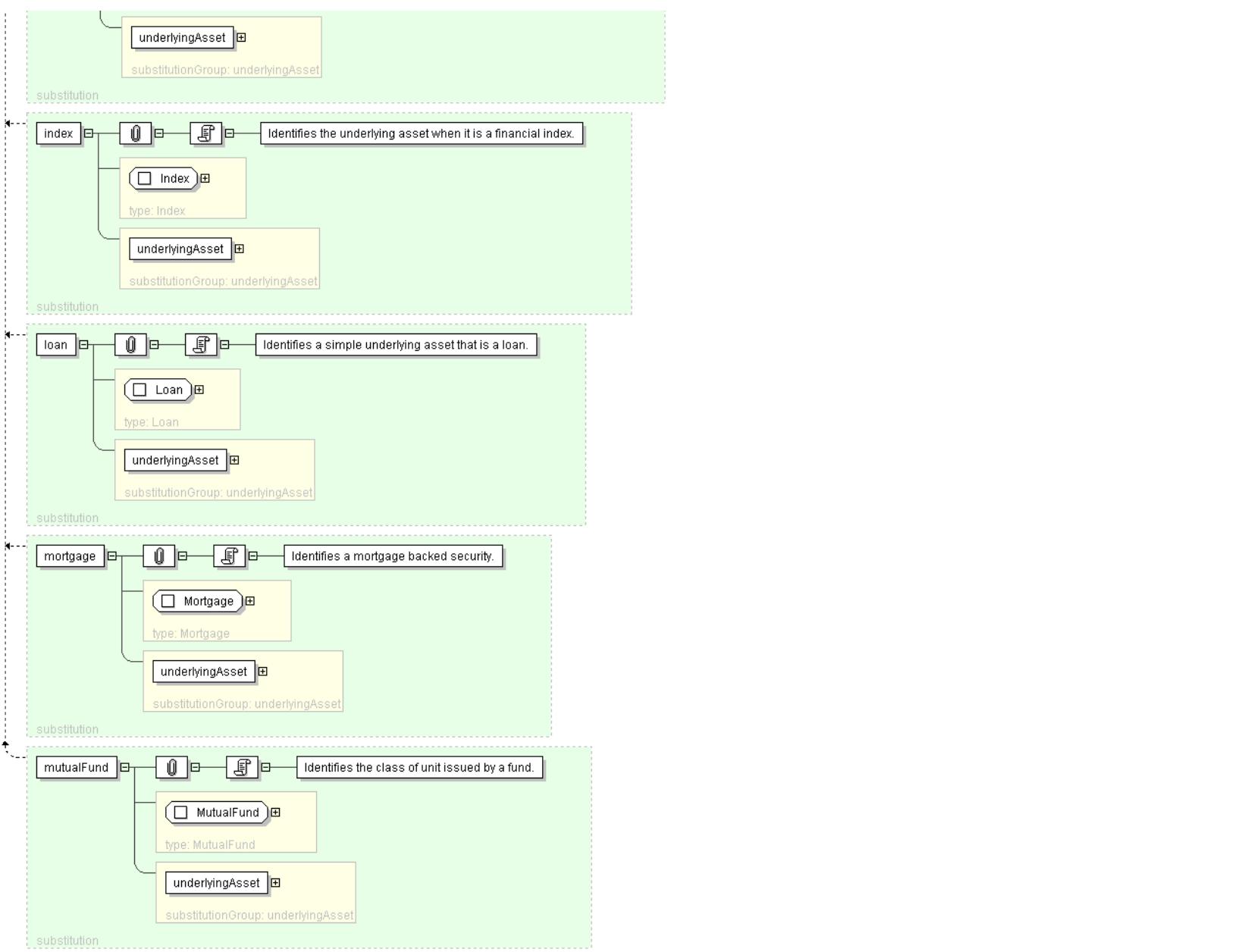
#### XML Instance Representation

```
<underlyingAsset  
id=" xsd:ID [0..1]" />
```

#### Diagram





**Schema Component Representation**

```
<xsd:element name="underlyingAsset" type="Asset" abstract="true"/>
```

top

**Global Definitions**Complex Type: **ActualPrice**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None

<b>Name</b>	ActualPrice
<b>Used by (from the same schema document)</b>	Model Group <a href="#">EquityPrice.model</a> , Model Group <a href="#">EquityPrice.model</a>
<b>Abstract</b>	no

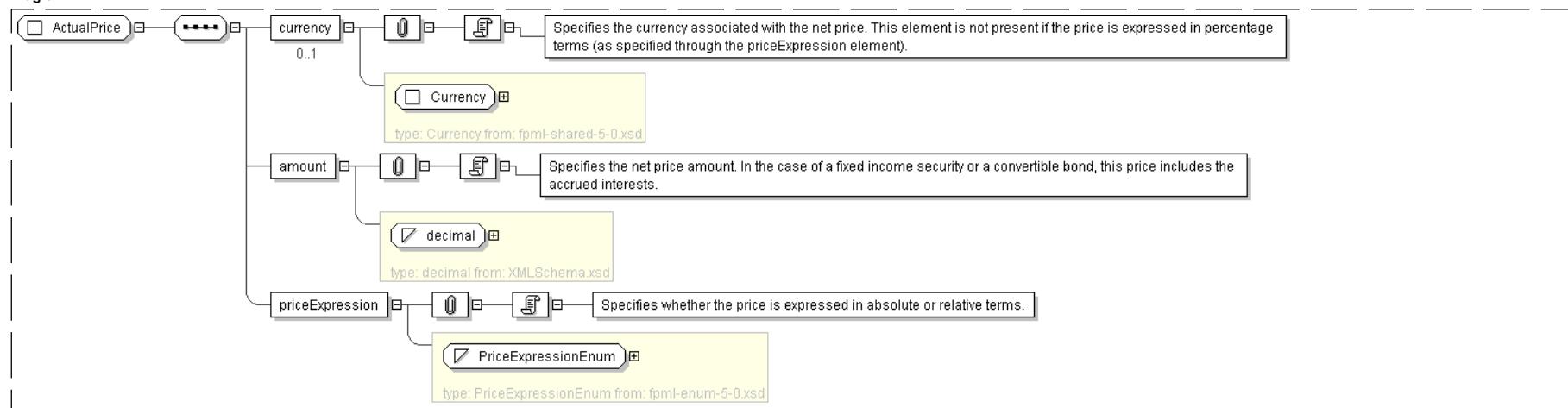
**XML Instance Representation**

```
<...>
<currency> Currency </currency> [0..1]
'Specifies the currency associated with the net price. This element is not present if the
price is expressed in percentage terms (as specified through the priceExpression element).'

<amount> xsd:decimal </amount> [1]
'Specifies the net price amount. In the case of a fixed income security or a convertible
bond, this price includes the accrued interests.'

<priceExpression> PriceExpressionEnum </priceExpression> [1]
'Specifies whether the price is expressed in absolute or relative terms.'

</...>
```

**Diagram****Schema Component Representation**

```
<xsd:complexType name="ActualPrice">
  <xsd:sequence>
    <xsd:element name="currency" type=" Currency " minOccurs="0" />
    <xsd:element name="amount" type=" xsd:decimal " />
    <xsd:element name="priceExpression" type=" PriceExpressionEnum " />
  </xsd:sequence>
</xsd:complexType>
```

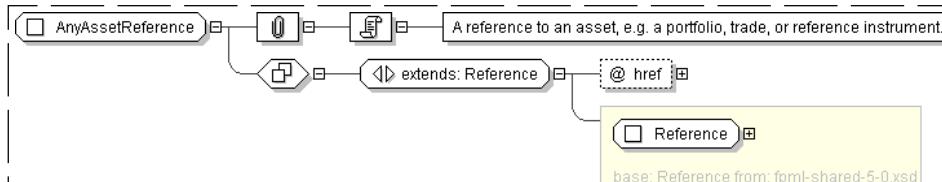
**Complex Type: AnyAssetReference**

<b>Super-types:</b>	<a href="#">Reference</a> < <b>AnyAssetReference</b> (by extension)
<b>Sub-types:</b>	None

<b>Name</b>	<b>AnyAssetReference</b>
<b>Abstract</b>	no
<b>Documentation</b>	A reference to an asset, e.g. a portfolio, trade, or reference instrument..

**XML Instance Representation**

```
<...  
    href=" xsd:IDREF [1]" />
```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="AnyAssetReference">
  <xsd:complexContent>
    <xsd:extension base=" Reference ">
      <xsd:attribute name="href" type=" xsd:IDREF " use="required"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
  
```

top

**Complex Type: Asset****Super-types:**

None

**Sub-types:**

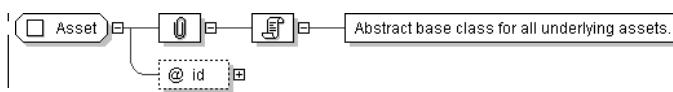
- [Basket](#) (by extension)
- [IdentifiedAsset](#) (by extension)
  - [Cash](#) (by extension)
  - [Commodity](#) (by extension)
  - [CurveInstrument](#) (by extension)
  - [UnderlyingAsset](#) (by extension)
    - [Bond](#) (by extension)
      - [ConvertibleBond](#) (by extension)
    - [Deposit](#) (by extension)
    - [ExchangeTraded](#) (by extension)
      - [EquityAsset](#) (by extension)
      - [ExchangeTradedCalculatedPrice](#) (by extension)
      - [ExchangeTradedFund](#) (by extension)
        - [Index](#) (by extension)
      - [ExchangeTradedContract](#) (by extension)
      - [Future](#) (by extension)
    - [FxRateAsset](#) (by extension)
    - [Loan](#) (by extension)
    - [Mortgage](#) (by extension)
    - [MutualFund](#) (by extension)
    - [RateIndex](#) (by extension)
    - [SimpleCreditDefaultSwap](#) (by extension)
    - [SimpleFra](#) (by extension)
    - [SimpleISwap](#) (by extension)

<b>Name</b>	Asset
<b>Used by (from the same schema document)</b>	Element <a href="#">curveInstrument</a> , Element <a href="#">underlyingAsset</a>
<b>Abstract</b>	yes
<b>Documentation</b>	Abstract base class for all underlying assets.

**XML Instance Representation**

```
<...  
    id=" xsd:ID [0..1]" />
```

**Diagram**

**Schema Component Representation**

```

<xsd:complexType name="Asset" abstract="true">
  <xsd:attribute name="id" type="xsd:ID" />
</xsd:complexType>

```

top

**Complex Type: AssetMeasureType**

**Super-types:** [Scheme](#) < **AssetMeasureType** (by extension)

**Sub-types:** None

**Name** AssetMeasureType

**Used by (from the same schema document)** Model Group [QuotationCharacteristics.model](#)

**Abstract** no

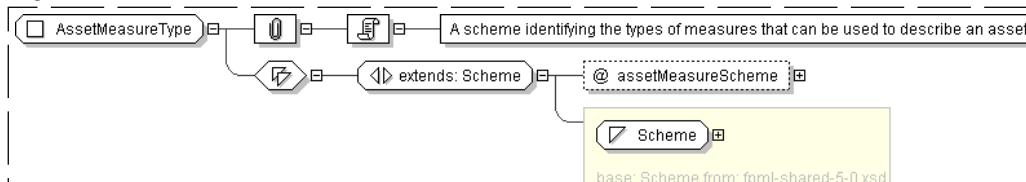
**Documentation** A scheme identifying the types of measures that can be used to describe an asset.

**XML Instance Representation**

```

<...
  assetMeasureScheme="xsd:anyURI [0..1]">
  Scheme
</...>

```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="AssetMeasureType">
  <xsd:simpleContent>
    <xsd:extension base="Scheme">
      <xsd:attribute name="assetMeasureScheme" type="xsd:anyURI" default="http://www.fpml.org/coding-scheme/asset-measure"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>

```

top

**Complex Type: AssetPool**

**Super-types:** None

**Sub-types:** None

**Name** AssetPool

**Used by (from the same schema document)** Complex Type [Mortgage](#)

**Abstract** no

**Documentation** Characterise the asset pool behind an asset backed bond.

**XML Instance Representation**

```

<...>
|

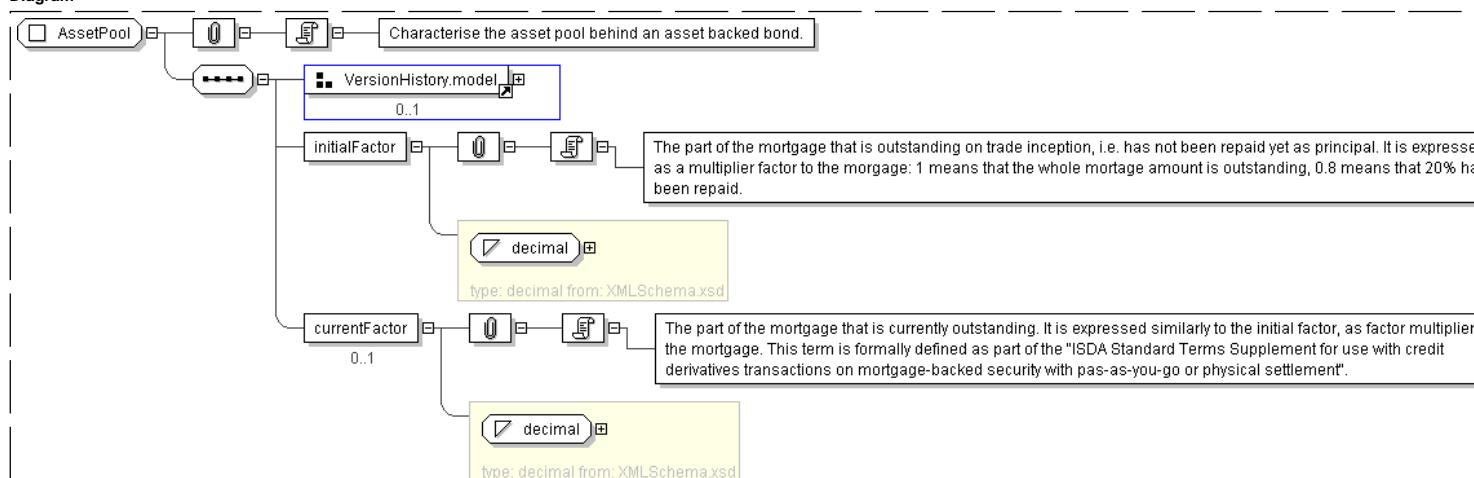
```

Start Group: [VersionHistory.model](#) [0..1]  
 <version> xsd:nonNegativeInteger </version> [1]  
 'The version number'  
 <effectiveDate> [IdentifiedDate](#) </effectiveDate> [0..1]  
 'Optionally it is possible to specify a version effective date when a versionId is supplied.'

End Group: [VersionHistory.model](#)  
 <initialFactor> xsd:decimal </initialFactor> [1]  
 'The part of the mortgage that is outstanding on trade inception, i.e. has not been repaid yet as principal. It is expressed as a multiplier factor to the mortgage: 1 means that the whole mortgage amount is outstanding, 0.8 means that 20% has been repaid.'

<currentFactor> xsd:decimal </currentFactor> [0..1]  
 'The part of the mortgage that is currently outstanding. It is expressed similarly to the initial factor, as factor multiplier to the mortgage. This term is formally defined as part of the "ISDA Standard Terms Supplement for use with credit derivatives transactions on mortgage-backed security with pas-as-you-go or physical settlement".'

&lt;/...&gt;

**Diagram****Schema Component Representation**

```

<xsd:complexType name="AssetPool">
  <xsd:sequence>
    <xsd:group ref=" VersionHistory.model " minOccurs="0"/>
    <xsd:element name="initialFactor" type=" xsd:decimal "/>
    <xsd:element name="currentFactor" type=" xsd:decimal " minOccurs="0 " />
  </xsd:sequence>
</xsd:complexType>
  
```

top

**Complex Type: AssetReference**

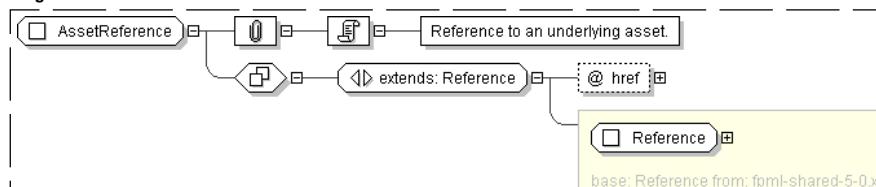
Super-types:	<a href="#">Reference</a> < <b>AssetReference</b> (by extension)
--------------	--

Sub-types:	None
------------	------

Name	AssetReference
Abstract	no
Documentation	Reference to an underlying asset.

**XML Instance Representation**

```
<...>
  <!-- href=" xsd:IDREF [1]" -->
```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="AssetReference">
  <xsd:complexContent>
    <xsd:extension base=" Reference ">
      <xsd:attribute name="href" type=" xsd:IDREF " use="required" reference="Asset"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
  
```

[top](#)**Complex Type: BasicQuotation**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None

<b>Name</b>	BasicQuotation
<b>Abstract</b>	no
<b>Documentation</b>	Some kind of numerical measure about an asset, eg. its NPV, together with characteristics of that measure.

**XML Instance Representation**

```

<....>
  <value> xsd:decimal </value> [0..1]
  'The value of the the quotation.'

  <measureType> AssetMeasureType </measureType> [0..1]
  'The type of the value that is measured. This could be an NPV, a cash flow, a clean price, etc.'

  <quoteUnits> PriceQuoteUnits </quoteUnits> [0..1]
  'The optional units that the measure is expressed in. If not supplied, this is assumed to be a price/value in currency units.'

  <side> QuotationSideEnum </side> [0..1]
  'The side (bid/mid/ask) of the measure.'

  <currency> Currency </currency> [0..1]
  'The optional currency that the measure is expressed in. If not supplied, this is defaulted from the reportingCurrency in the valuationScenarioDefinition.'

  <currencyType> ReportingCurrencyType </currencyType> [0..1]
  'The optional currency that the measure is expressed in. If not supplied, this is defaulted from the reportingCurrency in the valuationScenarioDefinition.'

  <timing> QuoteTiming </timing> [0..1]
  'When during a day the quote is for. Typically, if this element is supplied, the QuoteLocation needs also to be supplied.'

Start Group: QuoteLocation.model [0..1]
  'Where the quote is from.'

Start Choice [1]
  
```

```

<businessCenter> BusinessCenter </businessCenter> [1]
'A city or other business center.'

<exchangeId> ExchangeId </exchangeId> [1]
'The exchange (e.g. stock or futures exchange) from which the quote is obtained.'

End Choice
End Group: QuoteLocation.model
<informationSource> InformationSource </informationSource> [0..*]
'The information source where a published or displayed market rate will be obtained, e.g. Telerate Page 3750.'

<time> xsd:dateTime </time> [0..1]
'When the quote was observed or derived.'

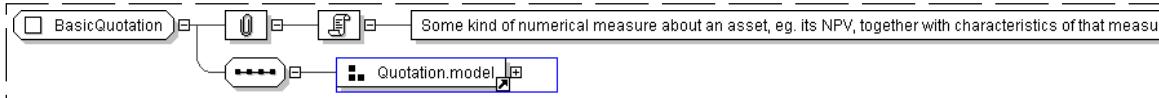
<valuationDate> xsd:date </valuationDate> [0..1]
'When the quote was computed.'

<expiryTime> xsd:dateTime </expiryTime> [0..1]
'When does the quote cease to be valid.'

<cashflowType> CashflowType </cashflowType> [0..1]
'For cash flows, the type of the cash flows. Examples include: Coupon payment, Premium Fee, Settlement Fee, Brokerage Fee, etc.'

```

&lt;/...&gt;

**Diagram****Schema Component Representation**

```

<xsd:complexType name="BasicQuotation">
  <xsd:sequence>
    <xsd:group ref="Quotation.model" />
  </xsd:sequence>
</xsd:complexType>

```

top

**Complex Type: Basket**

<b>Super-types:</b>	<a href="#">Asset</a> < <b>Basket</b> (by extension)
<b>Sub-types:</b>	None

<b>Name</b>	Basket
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">Underlyer</a> , Element <a href="#">basket</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type describing the underlyer features of a basket swap. Each of the basket constituents are described through an embedded component, the basketConstituentsType.

**XML Instance Representation**

```

<...
  id=" xsd:ID [0..1]">
  <openUnits> xsd:decimal </openUnits> [0..1]
  'The number of units (index or securities) that constitute the underlyer of the swap. In the case of a basket swap, this element is used to reference both the number of basket units, and the number of each asset components of the basket when these are expressed in absolute terms.'

  <basketConstituent> BasketConstituent </basketConstituent> [1..*]
  'Describes each of the components of the basket.'

```

```

<basketDivisor> xsd:decimal </basketDivisor> [0..1]
'Specifies the basket divisor amount. This value is normally used to adjust the
constituent weight for pricing or to adjust for dividends, or other corporate actions.'

Start Group: BasketIdentifier.model [0..1]
'Reuses the group that specifies a name and an identifier for a given basket.'

Start Choice [1]
<basketName> BasketName </basketName> [1]
'The name of the basket expressed as a free format string. FpML does not define usage rules
for this element.'

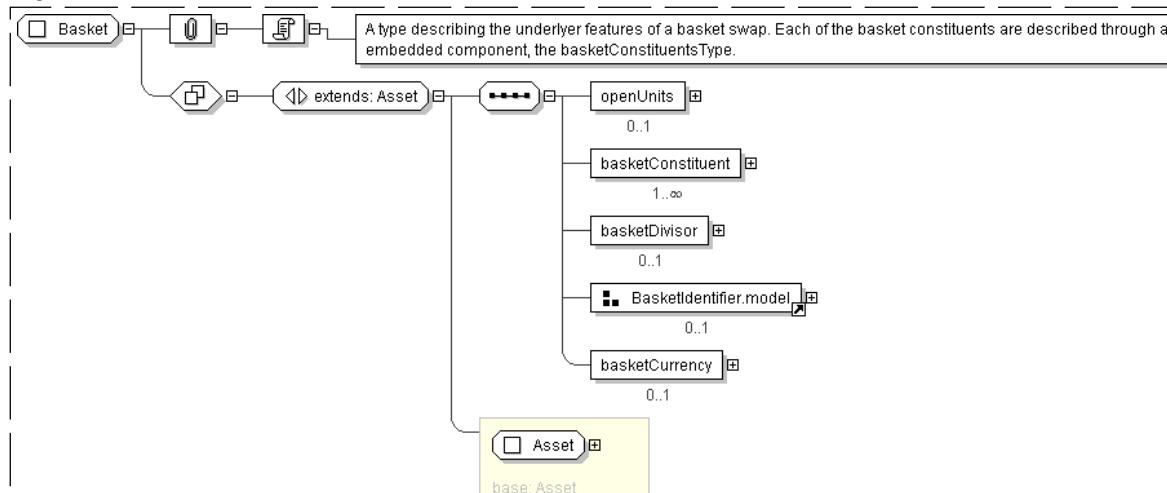
<basketId> BasketId </basketId> [0..*]
'A CDS basket identifier'

<basketId> BasketId </basketId> [1..*]
'A CDS basket identifier'

End Choice
End Group: BasketIdentifier.model
<basketCurrency> Currency </basketCurrency> [0..1]
'Specifies the currency for this basket.'

</...>

```

**Diagram****Schema Component Representation**

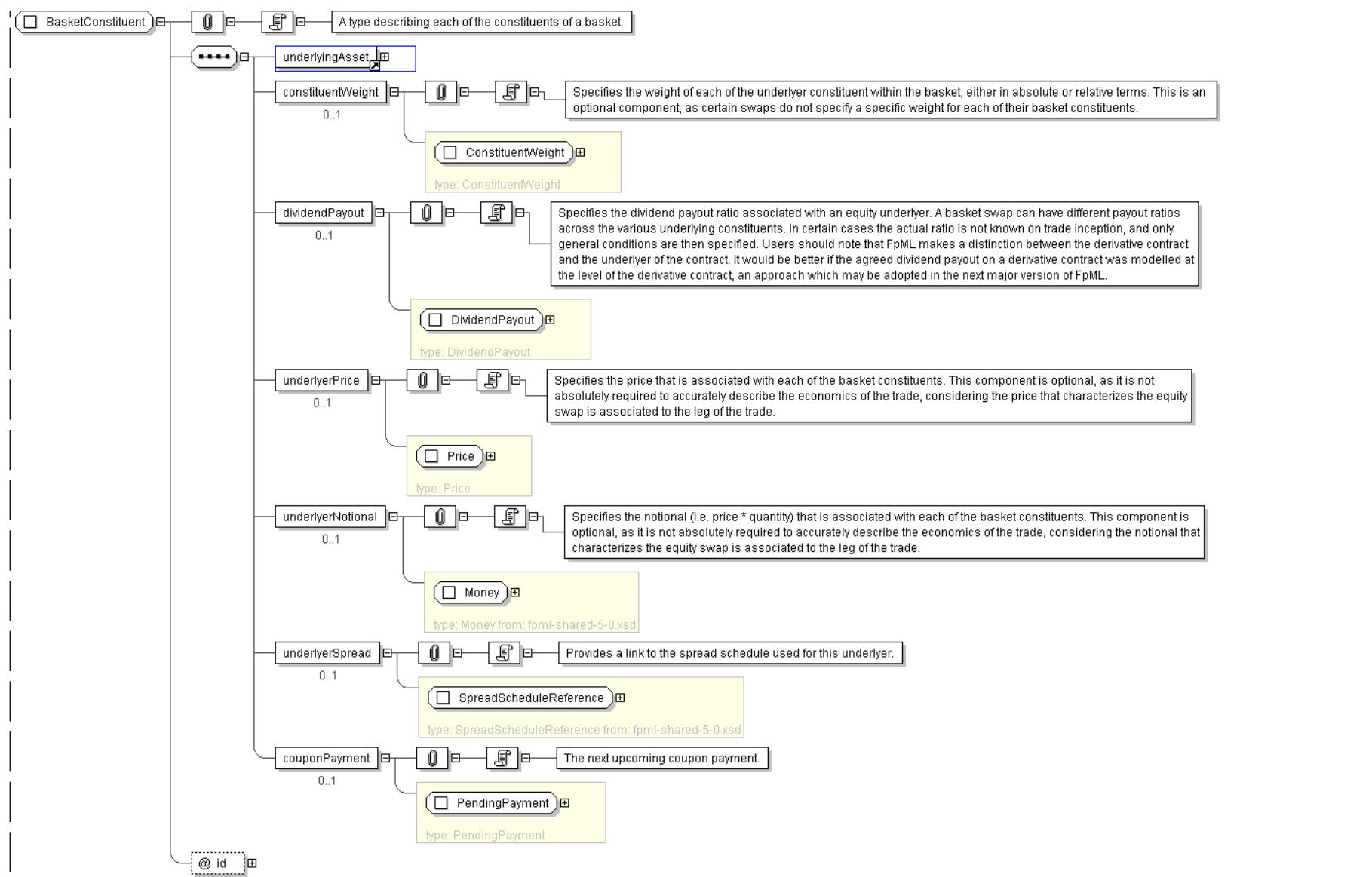
```

<xsd:complexType name="Basket">
  <xsd:complexContent>
    <xsd:extension base=" Asset ">
      <xsd:sequence>
        <xsd:element name="openUnits" type="xsd:decimal" minOccurs="0"/>
        <xsd:element name="basketConstituent" type=" BasketConstituent" maxOccurs="unbounded"/>
        <xsd:element name="basketDivisor" type="xsd:decimal" minOccurs="0"/>
        <xsd:group ref=" BasketIdentifier.model" minOccurs="0"/>
        <xsd:element name="basketCurrency" type=" Currency" minOccurs="0"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

**Complex Type: BasketConstituent**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None
<b>Name</b>	BasketConstituent
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">Basket</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type describing each of the constituents of a basket.
<b>XML Instance Representation</b>	
<pre>&lt;...&gt;   id="<a href="#">xsd:ID</a> [0..1]"*     &lt;<a href="#">underlyingAsset</a>&gt; ... &lt;/underlyingAsset&gt; [1]     &lt;<a href="#">constituentWeight</a>&gt; <a href="#">ConstituentWeight</a> &lt;/constituentWeight&gt; [0..1]       'Specifies the weight of each of the underlyer constituent within the basket, either        in absolute or relative terms. This is an optional component, as certain swaps do not specify        a specific weight for each of their basket constituents.'      &lt;<a href="#">dividendPayout</a>&gt; <a href="#">DividendPayout</a> &lt;/dividendPayout&gt; [0..1]       'Specifies the dividend payout ratio associated with an equity underlyer. A basket swap        can have different payout ratios across the various underlying constituents. In certain        cases the actual ratio is not known on trade inception, and only general conditions are        then specified. Users should note that FpML makes a distinction between the derivative        contract and the underlyer of the contract. It would be better if the agreed dividend payout        on a derivative contract was modelled at the level of the derivative contract, an        approach which may be adopted in the next major version of FpML.'      &lt;<a href="#">underlyerPrice</a>&gt; <a href="#">Price</a> &lt;/underlyerPrice&gt; [0..1]       'Specifies the price that is associated with each of the basket constituents. This component        is optional, as it is not absolutely required to accurately describe the economics of        the trade, considering the price that characterizes the equity swap is associated to the leg        of the trade.'      &lt;<a href="#">underlyerNotional</a>&gt; <a href="#">Money</a> &lt;/underlyerNotional&gt; [0..1]       'Specifies the notional (i.e. price * quantity) that is associated with each of the        basket constituents. This component is optional, as it is not absolutely required to        accurately describe the economics of the trade, considering the notional that characterizes        the equity swap is associated to the leg of the trade.'      &lt;<a href="#">underlyerSpread</a>&gt; <a href="#">SpreadScheduleReference</a> &lt;/underlyerSpread&gt; [0..1]       'Provides a link to the spread schedule used for this underlyer.'      &lt;<a href="#">couponPayment</a>&gt; <a href="#">PendingPayment</a> &lt;/couponPayment&gt; [0..1]       'The next upcoming coupon payment.'</pre>	
</...>	
<b>Diagram</b>	

**Schema Component Representation**

```

<xsd:complexType name="BasketConstituent">
  <xsd:sequence>
    <xsd:element ref=" underlyingAsset " />
    <xsd:element name="constituentWeight" type=" ConstituentWeight " minOccurs="0" />
    <xsd:element name="dividendPayout" type=" DividendPayout " minOccurs="0" />
    <xsd:element name="underlyerPrice" type=" Price " minOccurs="0" />
    <xsd:element name="underlyerNotional" type=" Money " minOccurs="0" />
    <xsd:element name="underlyerSpread" type=" SpreadScheduleReference " minOccurs="0" />
    <xsd:element name="couponPayment" type=" PendingPayment " minOccurs="0" />
  </xsd:sequence>
  <xsd:attribute name="id" type=" xsd:ID " use="optional" />
</xsd:complexType>

```

## Complex Type: BasketId

<b>Super-types:</b>	<a href="#">Scheme</a> < <b>BasketId</b> (by extension)
<b>Sub-types:</b>	None
<b>Name</b>	BasketId
<b>Used by (from the same schema document)</b>	Model Group <a href="#">BasketIdentifier.model</a> , Model Group <a href="#">BasketIdentifier.model</a>
<b>Abstract</b>	no
<b>XML Instance Representation</b>	<pre>&lt;...   basketIdScheme=" xsd:anyURI [0..1]"&gt;   Scheme &lt;/...&gt;</pre>
<b>Diagram</b>	<pre> classDiagram     class BasketId {         &lt;&lt;BasketId&gt;&gt;     }     class Scheme {         &lt;&lt;Scheme&gt;&gt;     }     BasketId "1..1" -- "0..1" Scheme : &lt;&lt;extends: Scheme&gt;&gt;     &lt;&lt;@ basketIdScheme &gt;&gt;     &lt;&lt;base: Scheme from: fpmi-shared-5-0.xsd&gt;&gt;   </pre>

### Schema Component Representation

```

<xsd:complexType name="BasketId">
  <xsd:simpleContent>
    <xsd:extension base=" Scheme ">
      <xsd:attribute name="basketIdScheme" type=" xsd:anyURI ">
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
  
```

top

## Complex Type: BasketName

<b>Super-types:</b>	<a href="#">Scheme</a> < <b>BasketName</b> (by extension)
<b>Sub-types:</b>	None
<b>Name</b>	BasketName
<b>Used by (from the same schema document)</b>	Model Group <a href="#">BasketIdentifier.model</a>
<b>Abstract</b>	no
<b>XML Instance Representation</b>	<pre>&lt;...   basketNameScheme=" xsd:anyURI [0..1]"&gt;   Scheme &lt;/...&gt;</pre>
<b>Diagram</b>	<pre> classDiagram     class BasketName {         &lt;&lt;BasketName&gt;&gt;     }     class Scheme {         &lt;&lt;Scheme&gt;&gt;     }     BasketName "1..1" -- "0..1" Scheme : &lt;&lt;extends: Scheme&gt;&gt;     &lt;&lt;@ basketNameScheme &gt;&gt;     &lt;&lt;base: Scheme from: fpmi-shared-5-0.xsd&gt;&gt;   </pre>
<b>Schema Component Representation</b>	<pre> &lt;xsd:complexType name="BasketName"&gt;   &lt;xsd:simpleContent&gt;     &lt;xsd:extension base=" Scheme "&gt;       &lt;xsd:attribute name="basketNameScheme" type=" xsd:anyURI "&gt;     &lt;/xsd:extension&gt;   &lt;/xsd:simpleContent&gt; &lt;/xsd:complexType&gt;   </pre>

```

  </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
```

**Complex Type: Bond****Super-types:**

[Asset](#) < [IdentifiedAsset](#) (by extension) < [UnderlyingAsset](#) (by extension) < **Bond** (by extension)

**Sub-types:**

- [ConvertibleBond](#) (by extension)

<b>Name</b>	Bond
<b>Used by (from the same schema document)</b>	Element <a href="#">bond</a>
<b>Abstract</b>	no
<b>Documentation</b>	An exchange traded bond.

**XML Instance Representation**

```

<...
  id=" xsd:ID [0..1]">
    <instrumentId> InstrumentId </instrumentId> [1..*]
      Identification of the underlying asset, using public and/or private identifiers.

    <description> xsd:string </description> [0..1]
      Long name of the underlying asset.

    <currency> IdentifiedCurrency </currency> [0..1]
      Trading currency of the underlyer when transacted as a cash instrument.

    <exchangeId> ExchangeId </exchangeId> [0..1]
      Identification of the exchange on which this asset is transacted for the purposes
      of calculating a contractual payoff. The term "Exchange" is assumed to have the meaning
      as defined in the ISDA 2002 Equity Derivatives Definitions.

    <clearanceSystem> ClearanceSystem </clearanceSystem> [0..1]
      Identification of the clearance system associated with the transaction exchange.

    <definition> ProductReference </definition> [0..1]
      An optional reference to a full FpML product that defines the simple product in
      greater detail. In case of inconsistency between the terms of the simple product and those
      of the detailed definition, the values in the simple product override those in the
      detailed definition.
```

Start [Choice](#) [0..1]

Specifies the issuer name of a fixed income security or convertible bond. This name can either be explicitly stated, or specified as an href into another element of the document, such as the obligor.

```

    <issuerName> xsd:string </issuerName> [1]
    <issuerPartyReference> PartyReference </issuerPartyReference> [1]
```

End Choice

```

    <seniority> CreditSeniority </seniority> [0..1]
      The repayment precedence of a debt instrument.
```

```

    <couponType> CouponType </couponType> [0..1]
      Specifies if the bond has a variable coupon, step-up/down coupon or a zero-coupon.
```

```

    <couponRate> xsd:decimal </couponRate> [0..1]
      Specifies the coupon rate (expressed in percentage) of a fixed income security or
      convertible bond.
```

```

    <maturity> xsd:date </maturity> [0..1]
      The date when the principal amount of a security becomes due and payable.
```

```

    <parValue> xsd:decimal </parValue> [0..1]
```

'Specifies the nominal amount of a fixed income security or convertible bond.'

```
<faceAmount> xsd:decimal </faceAmount> [0..1]
```

'Specifies the total amount of the issue. Corresponds to the par value multiplied by the number of issued security.'

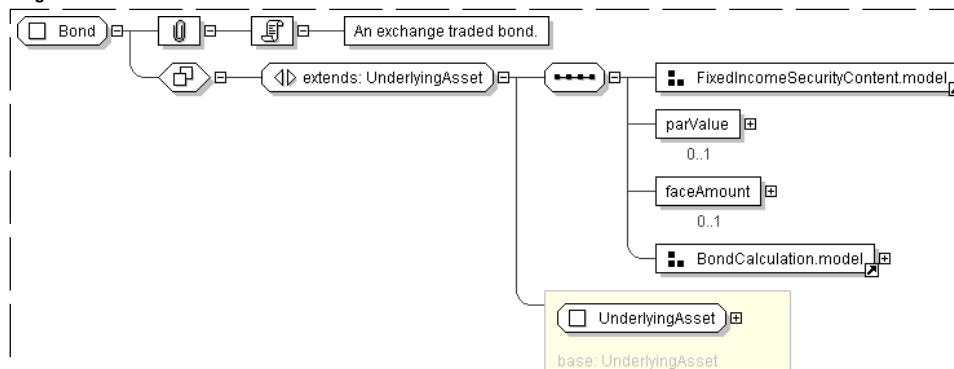
```
<paymentFrequency> Period </paymentFrequency> [0..1]
```

'Specifies the frequency at which the bond pays, e.g. 6M.'

```
<dayCountFraction> DayCountFraction </dayCountFraction> [0..1]
```

'The day count basis for the bond.'

```
</...>
```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="Bond">
  <xsd:complexContent>
    <xsd:extension base=" UnderlyingAsset ">
      <xsd:sequence>
        <xsd:group ref=" FixedIncomeSecurityContent.model ">
          <xsd:element name="parValue" type=" xsd:decimal " minOccurs="0"/>
          <xsd:element name="faceAmount" type=" xsd:decimal " minOccurs="0"/>
        </xsd:group>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
  
```

top

**Complex Type: Cash**

<b>Super-types:</b>	Asset < IdentifiedAsset (by extension) < Cash (by extension)
<b>Sub-types:</b>	None

<b>Name</b>	Cash
<b>Used by (from the same schema document)</b>	Element <a href="#">cash</a>
<b>Abstract</b>	no

**XML Instance Representation**

```

<...
  id=" xsd:ID [0..1]"*>
  <instrumentId> InstrumentId </instrumentId> [1..*]
  <Identification of the underlying asset, using public and/or private identifiers.>
  <description> xsd:string </description> [0..1]
  <Long name of the underlying asset.>
  
```

<currency> [Currency](#) </currency> [1]  
*'The currency in which an amount is denominated.'*

&lt;/...&gt;

**Diagram****Schema Component Representation**

```

<xsd:complexType name="Cash">
  <xsd:complexContent>
    <xsd:extension base=" IdentifiedAsset ">
      <xsd:sequence>
        <xsd:element name="currency" type="Currency" />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
  
```

top

**Complex Type: Commission**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None

<b>Name</b>	Commission
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">Price</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type describing the commission that will be charged for each of the hedge transactions.

**XML Instance Representation**

<...>  
 <commissionDenomination> [CommissionDenominationEnum](#) </commissionDenomination> [1]  
*'The type of units used to express a commission.'*

<commissionAmount> [xsd:decimal](#) </commissionAmount> [1]  
*'The commission amount, expressed in the way indicated by the commissionType element.'*

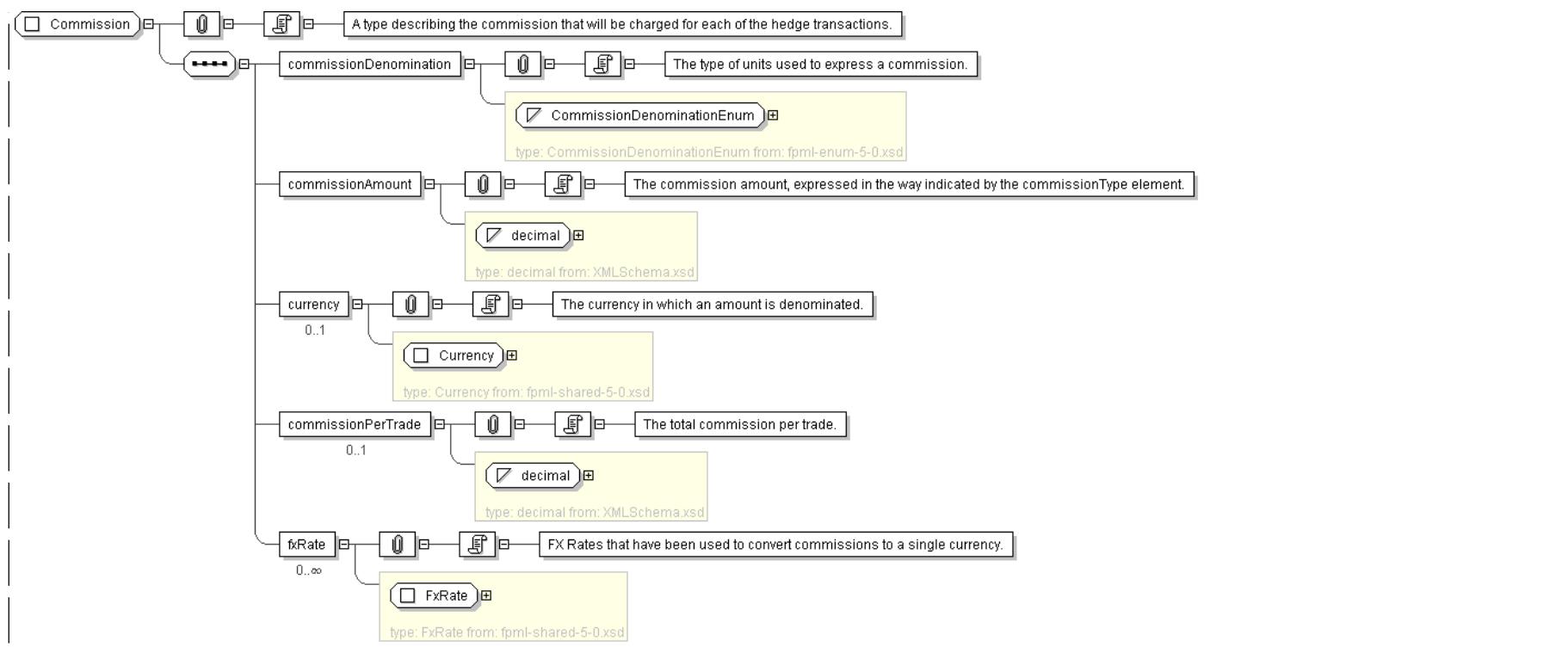
<currency> [Currency](#) </currency> [0..1]  
*'The currency in which an amount is denominated.'*

<commissionPerTrade> [xsd:decimal](#) </commissionPerTrade> [0..1]  
*'The total commission per trade.'*

<fxRate> [FxRate](#) </fxRate> [0..\*]  
*'FX Rates that have been used to convert commissions to a single currency.'*

&lt;/...&gt;

**Diagram**

**Schema Component Representation**

```

<xsd:complexType name="Commission">
  <xsd:sequence>
    <xsd:element name="commissionDenomination" type=" CommissionDenominationEnum " />
    <xsd:element name="commissionAmount" type=" xsd:decimal " />
    <xsd:element name="currency" type=" Currency " minOccurs="0"/>
    <xsd:element name="commissionPerTrade" type=" xsd:decimal " minOccurs="0"/>
    <xsd:element name="fxRate" type=" FxRate " minOccurs="0" maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>
  
```

top

**Complex Type: Commodity**

<b>Super-types:</b>	<a href="#">Asset</a> < <a href="#">IdentifiedAsset</a> (by extension) < <b>Commodity</b> (by extension)
<b>Sub-types:</b>	None
<b>Name</b>	Commodity
<b>Used by (from the same schema document)</b>	Element <a href="#">commodity</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type describing a commodity underlying asset.

**XML Instance Representation**

```

<...
  id=" xsd:ID [0..1]">
  <instrumentId> InstrumentId </instrumentId> [1..*]
  'Identification of the underlying asset, using public and/or private identifiers.'

  <description> xsd:string </description> [0..1]
  'Long name of the underlying asset.'
  
```

```

Start Group: CommodityReferencePriceFramework.model [0..1]
<commodityBase> CommodityBase </commodityBase> [1]
'A coding scheme value to identify the base type of the commodity being traded. Where
possible, this should follow the naming convention used in the 2005 ISDA Commodity
Definitions. For example, \'Oil\'.'


<commodityDetails> CommodityDetails </commodityDetails> [1]
'A coding scheme value to identify the commodity being traded more specifically.
Where possible, this should follow the naming convention used in the 2005 ISDA
Commodity Definitions. For example, \'Brent\''.


<unit> QuantityUnit </unit> [1]
'A coding scheme value to identify the unit in which the undelyer is denominated.
Where possible, this should follow the naming convention used in the 2005 ISDA
Commodity Definitions.'


<currency> Currency </currency> [1]
'The currency in which the Commodity Reference Price is published.'


Start Choice [1]
  <exchangeId> ExchangeId </exchangeId> [1]
    'For those commodities being traded with reference to the price of a listed future,
    the exchange where that future is listed should be specified here.'


  <publication> InformationSource </publication> [1]
    'For those commodities being traded with reference to a price distributed by a
    publication, that publication should be specified here.'


End Choice
End Group: CommodityReferencePriceFramework.model
<specifiedPrice> SpecifiedPriceEnum </specifiedPrice> [1]
'The Specified Price is not defined in the Commodity Reference Price and so needs to be
stated in the Underlyer definition as it will impact the calculation of the Floating Price.'


Start Sequence [0..1]
Start Choice [1]
  <deliveryDates> DeliveryDatesEnum </deliveryDates> [1]
    'The Delivery Date is a NearbyMonth, for use when the Commodity Transaction references
    Futures Contract.'


  <deliveryDate> AdjustableDate </deliveryDate> [1]
    'The Delivery Date is a fixed, single day.'


  <deliveryDateYearMonth> xsd:gYearMonth </deliveryDateYearMonth> [1]
    'The Delivery Date is a fixed, single month.'

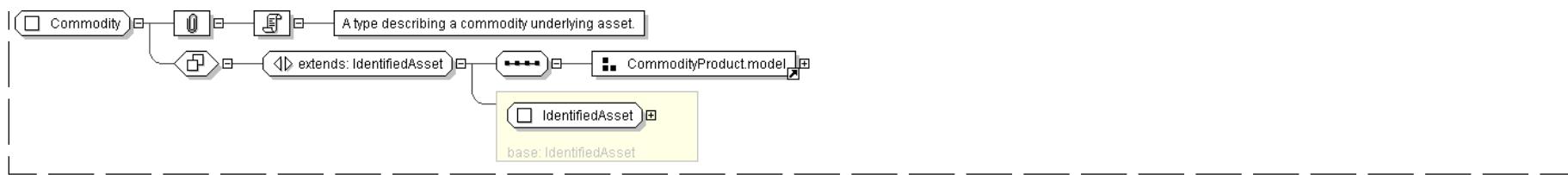

End Choice
<deliveryDateRollConvention> offset </deliveryDateRollConvention> [0..1]
'Specifies, for a Commodity Transaction that references a listed future via the
deliveryDates element, the day on which the specified future will roll to the next nearby
month when the referenced future expires. If the future will not roll at all - i.e. the
price will be taken from the expiring contract, 0 should be specified here. If the future
will roll to the next nearby on the last trading day - i.e. the price will be taken from
the next nearby on the last trading day, then 1 should be specified and so on.'


End Sequence
<multiplier> PositiveDecimal </multiplier> [0..1]
'Specifies the multiplier associated with a Transaction.'


</...>

```

**Diagram**

**Schema Component Representation**

```

<xsd:complexType name="Commodity">
  <xsd:complexContent>
    <xsd:extension base=" IdentifiedAsset ">
      <xsd:sequence>
        <xsd:group ref=" CommodityProduct.model ">/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
  
```

[top](#)**Complex Type: CommodityBase**

<b>Super-types:</b>	<a href="#">Scheme</a> < <b>CommodityBase</b> (by extension)
<b>Sub-types:</b>	None

<b>Name</b>	CommodityBase
<b>Used by (from the same schema document)</b>	Model Group <a href="#">CommodityReferencePriceFramework.model</a>
<b>Abstract</b>	no

**XML Instance Representation**

```

<...
  commodityBaseScheme=" xsd:anyURI [0..1]">
  Scheme
</...>
  
```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="CommodityBase">
  <xsd:simpleContent>
    <xsd:extension base=" Scheme ">
      <xsd:attribute name="commodityBaseScheme" type=" xsd:anyURI ">/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
  
```

[top](#)**Complex Type: CommodityBusinessCalendar**

<b>Super-types:</b>	<a href="#">Scheme</a> < <b>CommodityBusinessCalendar</b> (by extension)
<b>Sub-types:</b>	None

<b>Name</b>	CommodityBusinessCalendar
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">CommodityBusinessCalendarTime</a>

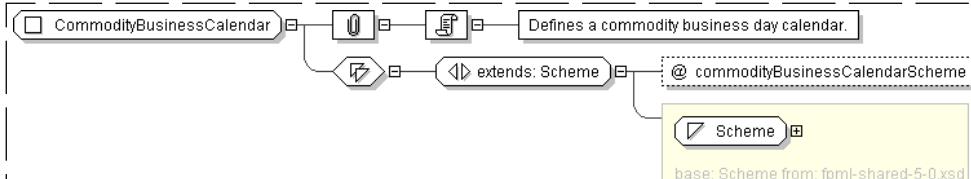
**Abstract** no

**Documentation** Defines a commodity business day calendar.

#### XML Instance Representation

```
<...>
  <commodityBusinessCalendarScheme=" xsd:anyURI [0..1]">
    Scheme
  </...>
```

#### Diagram



#### Schema Component Representation

```
<xsd:complexType name="CommodityBusinessCalendar">
  <xsd:simpleContent>
    <xsd:extension base=" Scheme ">
      <xsd:attribute name="commodityBusinessCalendarScheme" type=" xsd:anyURI " default="http://www.fpml.org/coding-scheme/commodity-business-calendar"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
```

top

### Complex Type: CommodityBusinessCalendarTime

**Super-types:** None

**Sub-types:** None

**Name** CommodityBusinessCalendarTime

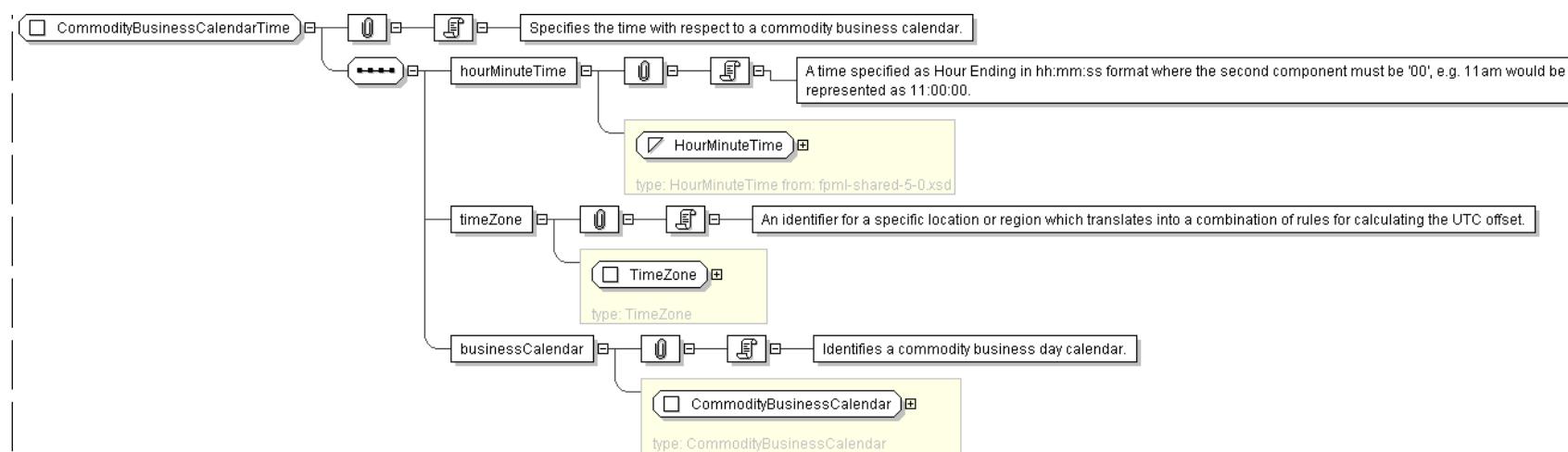
**Abstract** no

**Documentation** Specifies the time with respect to a commodity business calendar.

#### XML Instance Representation

```
<...>
  <hourMinuteTime> HourMinuteTime </hourMinuteTime> [1]
  'A time specified as Hour Ending in hh:mm:ss format where the second component must be \'00
  \', e.g. 11am would be represented as 11:00:00.'
  <timeZone> TimeZone </timeZone> [1]
  'An identifier for a specific location or region which translates into a combination of
  rules for calculating the UTC offset.'
  <businessCalendar> CommodityBusinessCalendar </businessCalendar> [1]
  'Identifies a commodity business day calendar.'
</...>
```

#### Diagram

**Schema Component Representation**

```

<xsd:complexType name="CommodityBusinessCalendarTime">
  <xsd:sequence>
    <xsd:element name="hourMinuteTime" type=" HourMinuteTime " />
    <xsd:element name="timeZone" type=" TimeZone " />
    <xsd:element name="businessCalendar" type=" CommodityBusinessCalendar " />
  </xsd:sequence>
</xsd:complexType>
  
```

top

**Complex Type: CommodityDetails**

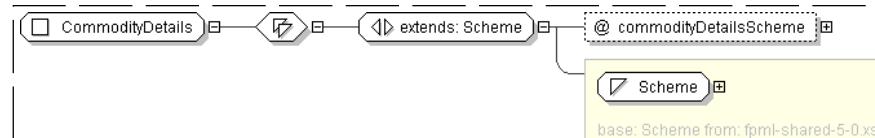
<b>Super-types:</b>	Scheme < CommodityDetails (by extension)
<b>Sub-types:</b>	None

<b>Name</b>	CommodityDetails
<b>Used by (from the same schema document)</b>	Model Group <a href="#">CommodityReferencePriceFramework.model</a>
<b>Abstract</b>	no

**XML Instance Representation**

```

<...
commodityDetailsScheme=" xsd:anyURI [0..1]">
Scheme
</...>
  
```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="CommodityDetails">
  <xsd:simpleContent>
    <xsd:extension base=" Scheme ">
      <xsd:attribute name="commodityDetailsScheme" type=" xsd:anyURI " />
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
  
```

## Complex Type: ConstituentWeight

**Super-types:** None  
**Sub-types:** None

<b>Name</b>	ConstituentWeight
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">BasketConstituent</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type describing the weight of each of the underlyer constituent within the basket, either in absolute or relative terms.

### XML Instance Representation

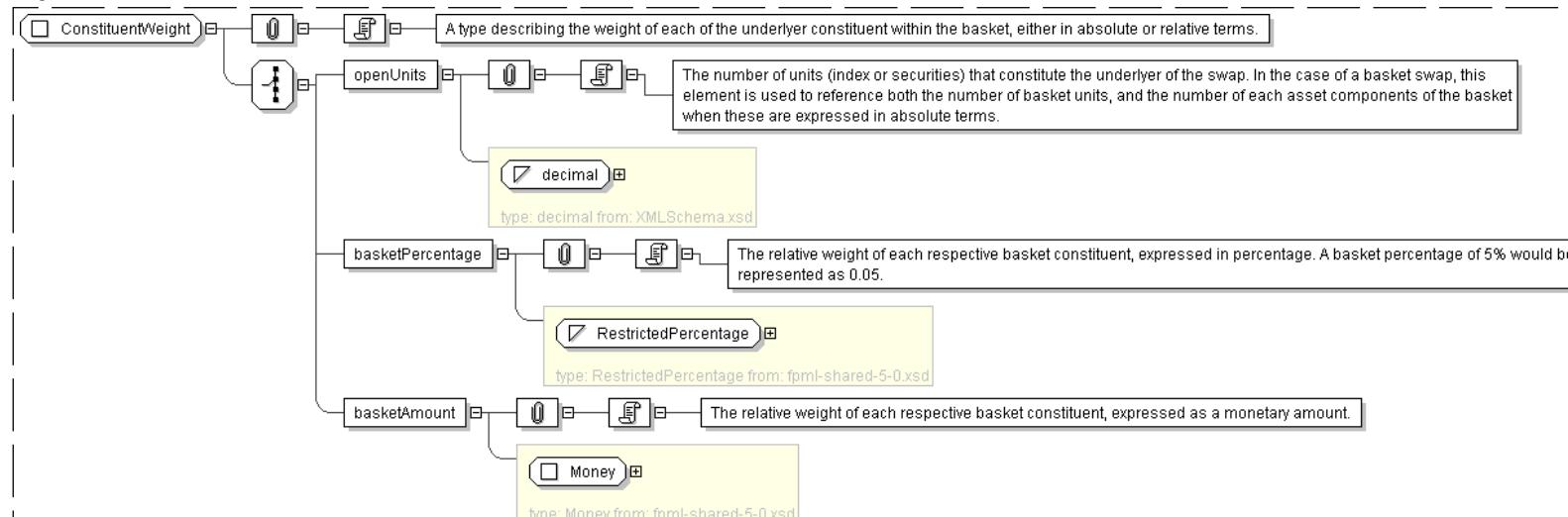
```
<....>
Start Choice [1]
<openUnits> xsd:decimal </openUnits> [1]
'The number of units (index or securities) that constitute the underlyer of the swap. In
the case of a basket swap, this element is used to reference both the number of basket
units, and the number of each asset components of the basket when these are expressed
in absolute terms.'

<basketPercentage> RestrictedPercentage </basketPercentage> [1]
'The relative weight of each respective basket constituent, expressed in percentage. A
basket percentage of 5% would be represented as 0.05.'

<basketAmount> Money </basketAmount> [1]
'The relative weight of each respective basket constituent, expressed as a monetary amount.'

End Choice
<....>
```

### Diagram



### Schema Component Representation

```

<xsd:complexType name="ConstituentWeight">
  <xsd:choice>
    <xsd:element name="openUnits" type="xsd:decimal" />
    <xsd:element name="basketPercentage" type="RestrictedPercentage" />
    <xsd:element name="basketAmount" type="Money" />
  </xsd:choice>
</xsd:complexType>

```

## Complex Type: ConvertibleBond

Super-types:	<a href="#">Asset</a> < <a href="#">IdentifiedAsset</a> (by extension) < <a href="#">UnderlyingAsset</a> (by extension) < <a href="#">Bond</a> (by extension) < <b>ConvertibleBond</b> (by extension)
Sub-types:	None

Name	ConvertibleBond
Used by (from the same schema document)	Element <a href="#">convertibleBond</a>
Abstract	no

### XML Instance Representation

```
<...>
  <id="" xsd:ID [0..1]*>
    <instrumentId> InstrumentId </instrumentId> [1..*]
      'Identification of the underlying asset, using public and/or private identifiers.'

    <description> xsd:string </description> [0..1]
      'Long name of the underlying asset.'

    <currency> IdentifiedCurrency </currency> [0..1]
      'Trading currency of the underlyer when transacted as a cash instrument.'

    <exchangeId> ExchangeId </exchangeId> [0..1]
      'Identification of the exchange on which this asset is transacted for the purposes
      of calculating a contractual payoff. The term "Exchange" is assumed to have the meaning
      as defined in the ISDA 2002 Equity Derivatives Definitions.'

    <clearanceSystem> ClearanceSystem </clearanceSystem> [0..1]
      'Identification of the clearance system associated with the transaction exchange.'
```

<definition> [ProductReference](#) </definition> [0..1]

'An optional reference to a full FpML product that defines the simple product in  
greater detail. In case of inconsistency between the terms of the simple product and those  
of the detailed definition, the values in the simple product override those in the  
detailed definition.'

Start [Choice](#) [0..1]

'Specifies the issuer name of a fixed income security or convertible bond. This name can  
either be explicitly stated, or specified as an href into another element of the document,  
such as the obligor.'

<issuerName> xsd:string </issuerName> [1]  
<issuerPartyReference> [PartyReference](#) </issuerPartyReference> [1]

End Choice  
<seniority> [CreditSeniority](#) </seniority> [0..1]

'The repayment precedence of a debt instrument.'

<couponType> [CouponType](#) </couponType> [0..1]

'Specifies if the bond has a variable coupon, step-up/down coupon or a zero-coupon.'

<couponRate> xsd:decimal </couponRate> [0..1]

'Specifies the coupon rate (expressed in percentage) of a fixed income security or  
convertible bond.'

<maturity> xsd:date </maturity> [0..1]

'The date when the principal amount of a security becomes due and payable.'

<parValue> xsd:decimal </parValue> [0..1]

'Specifies the nominal amount of a fixed income security or convertible bond.'

<faceAmount> xsd:decimal </faceAmount> [0..1]

'Specifies the total amount of the issue. Corresponds to the par value multiplied by the  
number of issued security.'

```

<paymentFrequency> Period </paymentFrequency> [0..1]
'Specifies the frequency at which the bond pays, e.g. 6M.'

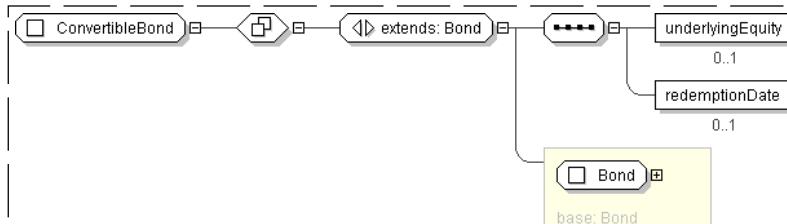
<dayCountFraction> DayCountFraction </dayCountFraction> [0..1]
'The day count basis for the bond.'

<underlyingEquity> EquityAsset </underlyingEquity> [0..1]
'Specifies the equity in which the convertible bond can be converted.'

<redemptionDate> xsd:date </redemptionDate> [0..1]
'Earlier date between the convertible bond put dates and its maturity date.'

</...>

```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="ConvertibleBond">
  <xsd:complexContent>
    <xsd:extension base=" Bond ">
      <xsd:sequence>
        <xsd:element name="underlyingEquity" type=" EquityAsset " minOccurs="0"/>
        <xsd:element name="redemptionDate" type=" xsd:date " minOccurs="0"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

top

**Complex Type: CouponType**

<b>Super-types:</b>	<a href="#">Scheme</a> < <b>CouponType</b> (by extension)
<b>Sub-types:</b>	None
<b>Name</b>	CouponType
<b>Used by (from the same schema document)</b>	Model Group <a href="#">FixedIncomeSecurityContent.model</a>
<b>Abstract</b>	no
<b>Documentation</b>	Defines a scheme of values for specifying if the bond has a variable coupon, step-up/down coupon or a zero-coupon.

**XML Instance Representation**

```

<...
  couponTypeScheme=" xsd:anyURI [0..1]">
  <u>Scheme</u>
</...>

```

**Diagram**

**Schema Component Representation**

```
<xsd:complexType name="CouponType">
  <xsd:simpleContent>
    <xsd:extension base=" Scheme ">
      <xsd:attribute name="couponTypeScheme" type=" xsd:anyURI " default="http://www.fpml.org/
        coding-scheme/coupon-type"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
```

top

**Complex Type: CurveInstrument**

**Super-types:** Asset < IdentifiedAsset (by extension) < CurveInstrument (by extension)

**Sub-types:** None

Name	CurveInstrument
------	-----------------

Abstract	yes
----------	-----

Documentation	Abstract base class for instruments intended to be used primarily for building curves.
---------------	--

**XML Instance Representation**

```
<...>
  id=" xsd:ID [0..1]">
    <instrumentId> InstrumentId </instrumentId> [1..*]
    'Identification of the underlying asset, using public and/or private identifiers.'

    <description> xsd:string </description> [0..1]
    'Long name of the underlying asset.'

    <currency> Currency </currency> [0..1]
    'Currency in which the underlying asset is denominated.'

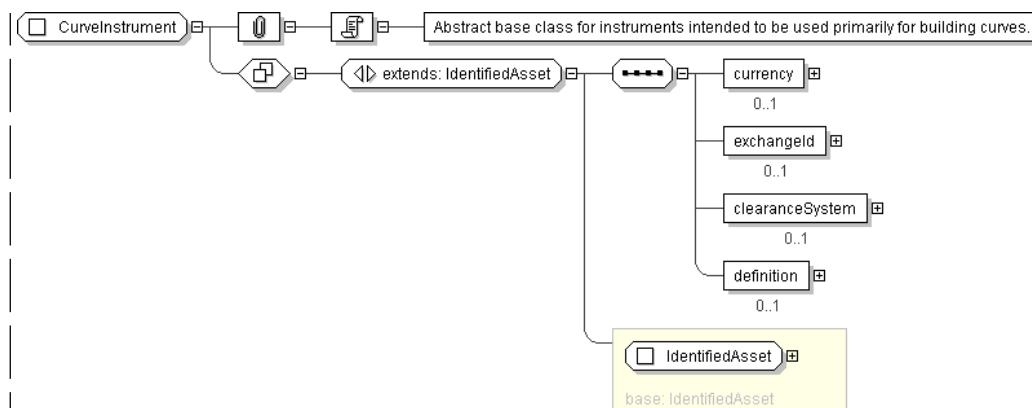
    <exchangeId> ExchangeId </exchangeId> [0..1]
    'Identification of the exchange on which this asset is transacted for the purposes
    of calculating a contractual payoff. The term "Exchange" is assumed to have the meaning
    as defined in the ISDA 2002 Equity Derivatives Definitions.'

    <clearanceSystem> ClearanceSystem </clearanceSystem> [0..1]
    'Identification of the clearance system associated with the transaction exchange.'

    <definition> ProductReference </definition> [0..1]
    'An optional reference to a full FpML product that defines the simple product in
    greater detail. In case of inconsistency between the terms of the simple product and those
    of the detailed definition, the values in the simple product override those in the
    detailed definition.'

  </...>
```

**Diagram**

**Schema Component Representation**

```

<xsd:complexType name="CurveInstrument" abstract="true">
  <xsd:complexContent>
    <xsd:extension base=" IdentifiedAsset ">
      <xsd:sequence>
        <xsd:element name="currency" type=" Currency " minOccurs="0" />
        <xsd:element name="exchangeId" type=" ExchangeId " minOccurs="0" />
        <xsd:element name="clearanceSystem" type=" ClearanceSystem " minOccurs="0" />
        <xsd:element name="definition" type=" ProductReference " minOccurs="0" />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
  
```

top

**Complex Type: Deposit**

<b>Super-types:</b>	<code>Asset</code> < <code>IdentifiedAsset</code> (by extension) < <code>UnderlyingAsset</code> (by extension) < <code>Deposit</code> (by extension)
<b>Sub-types:</b>	None

<b>Name</b>	Deposit
<b>Used by (from the same schema document)</b>	Element <a href="#">deposit</a>
<b>Abstract</b>	no

**XML Instance Representation**

```

<...
  id=" xsd:ID [0..1]">
  <instrumentId> InstrumentId </instrumentId> [1..*]
  'Identification of the underlying asset, using public and/or private identifiers.'

  <description> xsd:string </description> [0..1]
  'Long name of the underlying asset.'

  <currency> IdentifiedCurrency </currency> [0..1]
  'Trading currency of the underlyer when transacted as a cash instrument.'

  <exchangeId> ExchangeId </exchangeId> [0..1]
  'Identification of the exchange on which this asset is transacted for the purposes
  of calculating a contractual payoff. The term \"Exchange\" is assumed to have the meaning
  as defined in the ISDA 2002 Equity Derivatives Definitions.'

  <clearanceSystem> ClearanceSystem </clearanceSystem> [0..1]
  'Identification of the clearance system associated with the transaction exchange.'

  <definition> ProductReference </definition> [0..1]
  
```

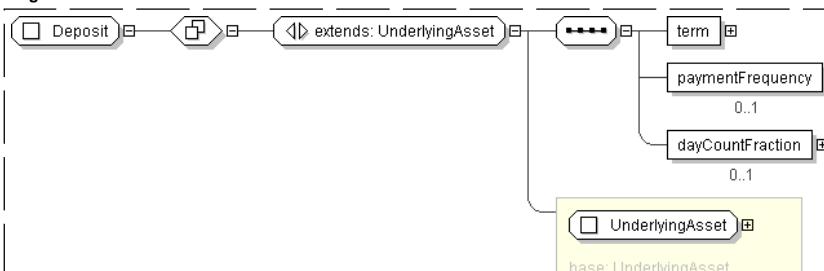
'An optional reference to a full FpML product that defines the simple product in greater detail. In case of inconsistency between the terms of the simple product and those of the detailed definition, the values in the simple product override those in the detailed definition.'

<term> Period </term> [1]  
*'Specifies the term of the deposit, e.g. 5Y.'*

<paymentFrequency> Period </paymentFrequency> [0..1]  
*'Specifies the frequency at which the deposit pays, e.g. 6M.'*

<dayCountFraction> DayCountFraction </dayCountFraction> [0..1]  
*'The day count basis for the deposit.'*

&lt;/...&gt;

**Diagram****Schema Component Representation**

```

<xsd:complexType name="Deposit">
  <xsd:complexContent>
    <xsd:extension base=" UnderlyingAsset ">
      <xsd:sequence>
        <xsd:element name="term" type=" Period "/>
        <xsd:element name="paymentFrequency" type=" Period " minOccurs="0"/>
        <xsd:element name="dayCountFraction" type=" DayCountFraction " minOccurs="0"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
  
```

top

**Complex Type: DividendPayout**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None

<b>Name</b>	DividendPayout
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">BasketConstituent</a> , Complex Type <a href="#">SingleUnderlyer</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type describing the dividend payout ratio associated with an equity underlyer. In certain cases the actual ratio is not known on trade inception, and only general conditions are then specified.

**XML Instance Representation**

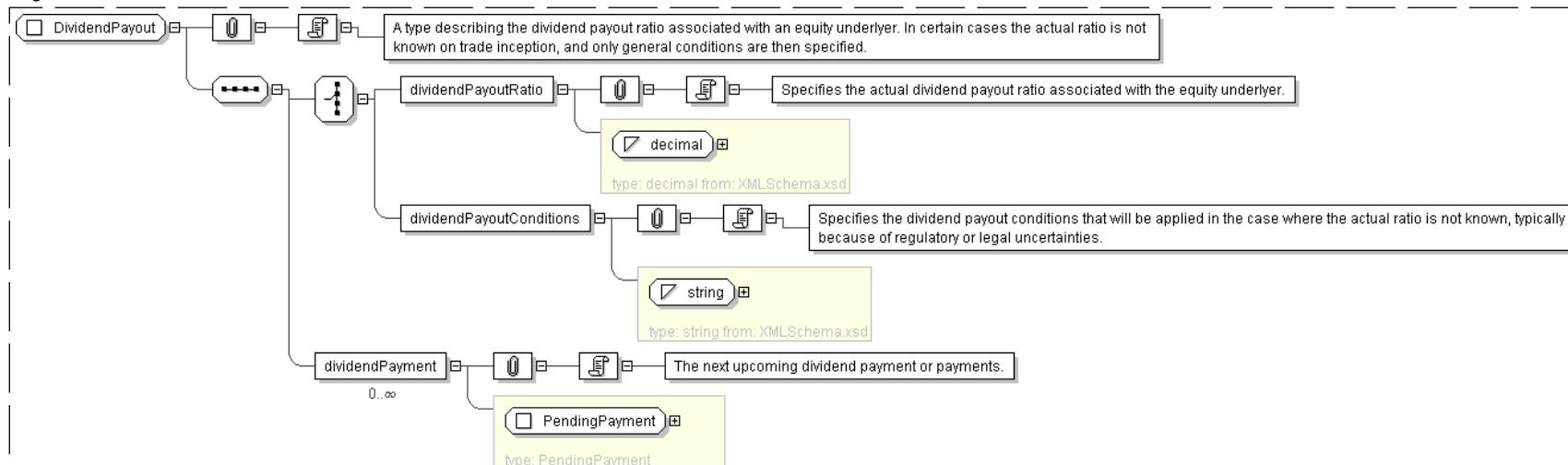
```

<....>
Start Choice [1]
  <dividendPayoutRatio> xsd:decimal </dividendPayoutRatio> [1]
  'Specifies the actual dividend payout ratio associated with the equity underlyer.'

  <dividendPayoutConditions> xsd:string </dividendPayoutConditions> [1]
  'Specifies the dividend payout conditions that will be applied in the case where the
actual ratio is not known, typically because of regulatory or legal uncertainties.'
  
```

```
End Choice
<dividendPayment> PendingPayment </dividendPayment> [0..*]
'The next upcoming dividend payment or payments.'
```

&lt;/...&gt;

**Diagram****Schema Component Representation**

```
<xsd:complexType name="DividendPayout">
  <xsd:sequence>
    <xsd:choice>
      <xsd:element name="dividendPayoutRatio" type="xsd:decimal" />
      <xsd:element name="dividendPayoutConditions" type="xsd:string" />
    </xsd:choice>
    <xsd:element name="dividendPayment" type="PendingPayment" minOccurs="0"
      maxOccurs="unbounded" />
  </xsd:sequence>
</xsd:complexType>
```

top

**Complex Type: EquityAsset**

<b>Super-types:</b>	<a href="#">Asset</a> < <a href="#">IdentifiedAsset</a> (by extension) < <a href="#">UnderlyingAsset</a> (by extension) < <a href="#">ExchangeTraded</a> (by extension) < <b>EquityAsset</b> (by extension)
<b>Sub-types:</b>	None

<b>Name</b>	EquityAsset
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">ConvertibleBond</a> , Element <a href="#">equity</a>
<b>Abstract</b>	no
<b>Documentation</b>	An exchange traded equity asset.

**XML Instance Representation**

```
<...
  id="xsd:ID [0..1]">
  <instrumentId> InstrumentId </instrumentId> [1..*]
  'Identification of the underlying asset, using public and/or private identifiers.'

  <description> xsd:string </description> [0..1]
  'Long name of the underlying asset.'

  <currency> IdentifiedCurrency </currency> [0..1]
```

'Trading currency of the underlyer when transacted as a cash instrument.'

```
<exchangeId> ExchangeId </exchangeId> [0..1]
```

'Identification of the exchange on which this asset is transacted for the purposes of calculating a contractual payoff. The term \"Exchange\" is assumed to have the meaning as defined in the ISDA 2002 Equity Derivatives Definitions.'

```
<clearanceSystem> ClearanceSystem </clearanceSystem> [0..1]
```

'Identification of the clearance system associated with the transaction exchange.'

```
<definition> productReference </definition> [0..1]
```

'An optional reference to a full FpML product that defines the simple product in greater detail. In case of inconsistency between the terms of the simple product and those of the detailed definition, the values in the simple product override those in the detailed definition.'

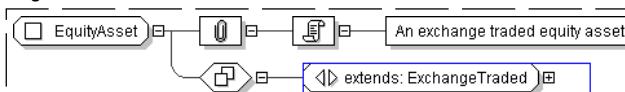
```
<relatedExchangeId> ExchangeId </relatedExchangeId> [0..*]
```

'A short form unique identifier for a related exchange. If the element is not present then the exchange shall be the primary exchange on which listed futures and options on the underlying are listed. The term \"Exchange\" is assumed to have the meaning as defined in the ISDA 2002 Equity Derivatives Definitions.'

```
<optionsExchangeId> ExchangeId </optionsExchangeId> [0..*]
```

'A short form unique identifier for an exchange on which the reference option contract is listed. This is to address the case where the reference exchange for the future is different than the one for the option. The options Exchange is referenced on share options when Merger Elections are selected as Options Exchange Adjustment.'

&lt;/...&gt;

**Diagram****Schema Component Representation**

```

<xsd:complexType name="EquityAsset">
  <xsd:complexContent>
    <xsd:extension base=" ExchangeTraded " />
  </xsd:complexContent>
</xsd:complexType>
  
```

top

**Complex Type: ExchangeTraded****Super-types:**

Asset < IdentifiedAsset (by extension) < UnderlyingAsset (by extension) < **ExchangeTraded** (by extension)

**Sub-types:**

- EquityAsset (by extension)
- ExchangeTradedCalculatedPrice (by extension)
  - ExchangeTradedFund (by extension)
  - Index (by extension)
- ExchangeTradedContract (by extension)
- Future (by extension)

<b>Name</b>	ExchangeTraded
-------------	----------------

<b>Abstract</b>	yes
-----------------	-----

<b>Documentation</b>	An abstract base class for all exchange traded financial products.
----------------------	--

**XML Instance Representation**

```

<...
id=" xsd:ID [0..1]">
<instrumentId> InstrumentId </instrumentId> [1..*]
'Identification of the underlying asset, using public and/or private identifiers.'

<description> xsd:string </description> [0..1]
  
```

'Long name of the underlying asset.'

<currency> [IdentifiedCurrency](#) </currency> [0..1]

'Trading currency of the underlyer when transacted as a cash instrument.'

<exchangeId> [ExchangeId](#) </exchangeId> [0..1]

'Identification of the exchange on which this asset is transacted for the purposes of calculating a contractual payoff. The term "Exchange" is assumed to have the meaning as defined in the ISDA 2002 Equity Derivatives Definitions.'

<clearanceSystem> [ClearanceSystem](#) </clearanceSystem> [0..1]

'Identification of the clearance system associated with the transaction exchange.'

<definition> [ProductReference](#) </definition> [0..1]

'An optional reference to a full FpML product that defines the simple product in greater detail. In case of inconsistency between the terms of the simple product and those of the detailed definition, the values in the simple product override those in the detailed definition.'

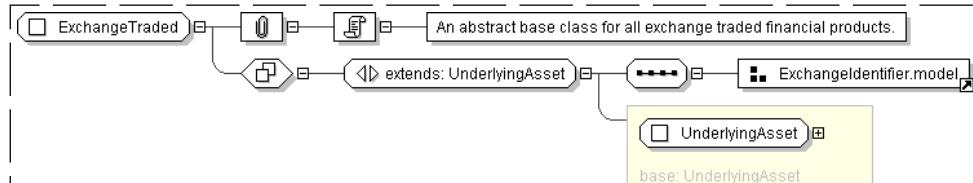
<relatedExchangeId> [ExchangeId](#) </relatedExchangeId> [0..\*]

'A short form unique identifier for a related exchange. If the element is not present then the exchange shall be the primary exchange on which listed futures and options on the underlying are listed. The term "Exchange" is assumed to have the meaning as defined in the ISDA 2002 Equity Derivatives Definitions.'

<optionsExchangeId> [ExchangeId](#) </optionsExchangeId> [0..\*]

'A short form unique identifier for an exchange on which the reference option contract is listed. This is to address the case where the reference exchange for the future is different than the one for the option. The options Exchange is referenced on share options when Merger Elections are selected as Options Exchange Adjustment.'

&lt;/...&gt;

**Diagram****Schema Component Representation**

```

<xsd:complexType name="ExchangeTraded" abstract="true">
  <xsd:complexContent>
    <xsd:extension base="UnderlyingAsset ">
      <xsd:sequence>
        <xsd:group ref=" ExchangeIdentifier.model ">
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
  
```

top

**Complex Type: ExchangeTradedCalculatedPrice****Super-types:**

[Asset](#) < [IdentifiedAsset](#) (by extension) < [UnderlyingAsset](#) (by extension) < [ExchangeTraded](#) (by extension) < **ExchangeTradedCalculatedPrice**  
(by extension)

**Sub-types:**

- [ExchangeTradedFund](#) (by extension)
- [Index](#) (by extension)

**Name**

ExchangeTradedCalculatedPrice

**Abstract**

yes

**Documentation**

Abstract base class for all exchange traded financial products with a price which is calculated from exchange traded constituents.

**XML Instance Representation**

```
<...>
  <id=" xsd:ID [0..1]">
    <instrumentId> InstrumentId </instrumentId> [1..*]
    'Identification of the underlying asset, using public and/or private identifiers.'

    <description> xsd:string </description> [0..1]
    'Long name of the underlying asset.'

    <currency> IdentifiedCurrency </currency> [0..1]
    'Trading currency of the underlyer when transacted as a cash instrument.'

    <exchangeId> ExchangeId </exchangeId> [0..1]
    'Identification of the exchange on which this asset is transacted for the purposes
    of calculating a contractual payoff. The term \"Exchange\" is assumed to have the meaning
    as defined in the ISDA 2002 Equity Derivatives Definitions.'

    <clearanceSystem> ClearanceSystem </clearanceSystem> [0..1]
    'Identification of the clearance system associated with the transaction exchange.'

    <definition> ProductReference </definition> [0..1]
    'An optional reference to a full FpML product that defines the simple product in
    greater detail. In case of inconsistency between the terms of the simple product and those
    of the detailed definition, the values in the simple product override those in the
    detailed definition.'

    <relatedExchangeId> ExchangeId </relatedExchangeId> [0..*]
    'A short form unique identifier for a related exchange. If the element is not present then
    the exchange shall be the primary exchange on which listed futures and options on
    the underlying are listed. The term \"Exchange\" is assumed to have the meaning as defined
    in the ISDA 2002 Equity Derivatives Definitions.'

    <optionsExchangeId> ExchangeId </optionsExchangeId> [0..*]
    'A short form unique identifier for an exchange on which the reference option contract
    is listed. This is to address the case where the reference exchange for the future is
    different than the one for the option. The options Exchange is referenced on share options
    when Merger Elections are selected as Options Exchange Adjustment.'

    <constituentExchangeId> ExchangeId </constituentExchangeId> [0..*]
    'Identification of all the exchanges where constituents are traded. The term \"Exchange\"
    is assumed to have the meaning as defined in the ISDA 2002 Equity Derivatives Definitions.'

<...>
```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="ExchangeTradedCalculatedPrice" abstract="true">
  <xsd:complexContent>
    <xsd:extension base=" ExchangeTraded ">
      <xsd:sequence>
        <xsd:element name="constituentExchangeId" type=" ExchangeId "
          minOccurs="0" maxOccurs="unbounded" />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
  
```

## Complex Type: ExchangeTradedContract

**Super-types:** Asset < IdentifiedAsset (by extension) < UnderlyingAsset (by extension) < ExchangeTraded (by extension) < ExchangeTradedContract (by extension)

**Sub-types:** None

<b>Name</b>	ExchangeTradedContract
<b>Abstract</b>	no
<b>Documentation</b>	An exchange traded derivative contract.

### XML Instance Representation

```

<...
  id=" xsd:ID [0..1]"*
    <instrumentId> InstrumentId </instrumentId> [1..*]
      'Identification of the underlying asset, using public and/or private identifiers.'

    <description> xsd:string </description> [0..1]
      'Long name of the underlying asset.'

    <currency> IdentifiedCurrency </currency> [0..1]
      'Trading currency of the underlyer when transacted as a cash instrument.'

    <exchangeId> ExchangeId </exchangeId> [0..1]
      'Identification of the exchange on which this asset is transacted for the purposes
      of calculating a contractual payoff. The term \'Exchange\' is assumed to have the meaning
      as defined in the ISDA 2002 Equity Derivatives Definitions.'

    <clearanceSystem> ClearanceSystem </clearanceSystem> [0..1]
      'Identification of the clearance system associated with the transaction exchange.'

    <definition> ProductReference </definition> [0..1]
      'An optional reference to a full FpML product that defines the simple product in
      greater detail. In case of inconsistency between the terms of the simple product and those
      of the detailed definition, the values in the simple product override those in the
      detailed definition.'

    <relatedExchangeId> ExchangeId </relatedExchangeId> [0..*]
      'A short form unique identifier for a related exchange. If the element is not present then
      the exchange shall be the primary exchange on which listed futures and options on
      the underlying are listed. The term \'Exchange\' is assumed to have the meaning as defined
      in the ISDA 2002 Equity Derivatives Definitions.'

    <optionsExchangeId> ExchangeId </optionsExchangeId> [0..*]
      'A short form unique identifier for an exchange on which the reference option contract
      is listed. This is to address the case where the reference exchange for the future is
      different than the one for the option. The Options Exchange is referenced on share options
      when Merger Elections are selected as Options Exchange Adjustment.'

    <multiplier> xsd:positiveInteger </multiplier> [0..1]
      'Specifies the contract multiplier that can be associated with the number of units.'

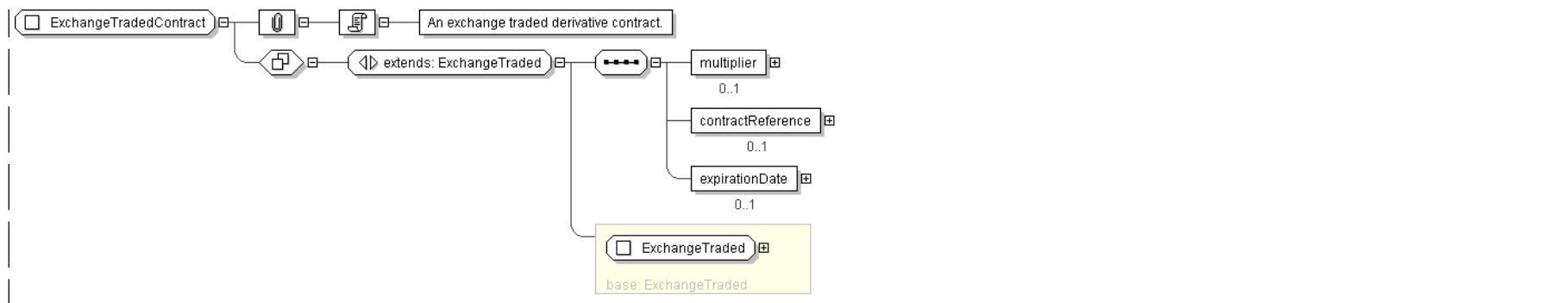
    <contractReference> xsd:string </contractReference> [0..1]
      'Specifies the contract that can be referenced, besides the undelyer type.'

    <expirationDate> AdjustableOrRelativeDate </expirationDate> [0..1]
      'The date when the contract expires.'

  </...>

```

### Diagram

**Schema Component Representation**

```

<xsd:complexType name="ExchangeTradedContract">
  <xsd:complexContent>
    <xsd:extension base=" ExchangeTraded ">
      <xsd:sequence>
        <xsd:element name="multiplier" type=" xsd:positiveInteger " minOccurs="0" />
        <xsd:element name="contractReference" type=" xsd:string " minOccurs="0" />
        <xsd:element name="expirationDate" type=" AdjustableOrRelativeDate " minOccurs="0" />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

top

**Complex Type: ExchangeTradedFund**

<b>Super-types:</b>	Asset < IdentifiedAsset (by extension) < UnderlyingAsset (by extension) < ExchangeTraded (by extension) < ExchangeTradedCalculatedPrice (by extension) < ExchangeTradedFund (by extension)
<b>Sub-types:</b>	None

<b>Name</b>	ExchangeTradedFund
<b>Used by (from the same schema document)</b>	Element <a href="#">exchangeTradedFund</a>
<b>Abstract</b>	no
<b>Documentation</b>	An exchange traded fund whose price depends on exchange traded constituents.

**XML Instance Representation**

```

<...>
<id=" xsd:ID [0..1]">
  <instrumentId> InstrumentId </instrumentId> [1..*]
  'Identification of the underlying asset, using public and/or private identifiers.'

  <description> xsd:string </description> [0..1]
  'Long name of the underlying asset.'

  <currency> IdentifiedCurrency </currency> [0..1]
  'Trading currency of the underlyer when transacted as a cash instrument.'

  <exchangeId> ExchangeId </exchangeId> [0..1]
  'Identification of the exchange on which this asset is transacted for the purposes
  of calculating a contractual payoff. The term "Exchange" is assumed to have the meaning
  as defined in the ISDA 2002 Equity Derivatives Definitions.'

  <clearanceSystem> ClearanceSystem </clearanceSystem> [0..1]
  'Identification of the clearance system associated with the transaction exchange.'

  <definition> ProductReference </definition> [0..1]
  'An optional reference to a full FpML product that defines the simple product in
  greater detail. In case of inconsistency between the terms of the simple product and those
  of the detailed definition, the values in the simple product override those in the

```

*detailed definition.'*

<relatedExchangeId> `ExchangeId` </relatedExchangeId> [0..\*]

'A short form unique identifier for a related exchange. If the element is not present then the exchange shall be the primary exchange on which listed futures and options on the underlying are listed. The term \"Exchange\" is assumed to have the meaning as defined in the ISDA 2002 Equity Derivatives Definitions.'

<optionsExchangeId> `ExchangeId` </optionsExchangeId> [0..\*]

'A short form unique identifier for an exchange on which the reference option contract is listed. This is to address the case where the reference exchange for the future is different than the one for the option. The options Exchange is referenced on share options when Merger Elections are selected as Options Exchange Adjustment.'

<constituentExchangeId> `ExchangeId` </constituentExchangeId> [0..\*]

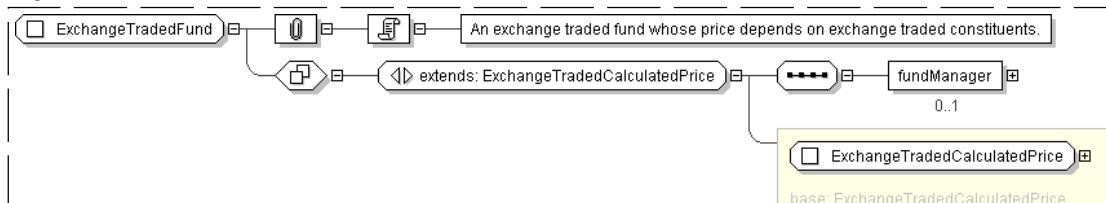
'Identification of all the exchanges where constituents are traded. The term \"Exchange\" is assumed to have the meaning as defined in the ISDA 2002 Equity Derivatives Definitions.'

<fundManager> `xsd:string` </fundManager> [0..1]

'Specifies the fund manager that is in charge of the fund.'

</...>

#### Diagram



#### Schema Component Representation

```

<xsd:complexType name="ExchangeTradedFund">
  <xsd:complexContent>
    <xsd:extension base=" ExchangeTradedCalculatedPrice ">
      <xsd:sequence>
        <xsd:element name="fundManager" type=" xsd:string " minOccurs="0 "/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
  
```

top

## Complex Type: FacilityType

Super-types:

[Scheme](#) < **FacilityType** (by extension)

Sub-types:

None

Name

FacilityType

Used by (from the same schema document)

Complex Type [Loan](#)

Abstract

no

Documentation

A type describing the type of loan facility.

#### XML Instance Representation

```

<...
  facilityTypeScheme=" xsd:anyURI [0..1]">
  Scheme
</...>
  
```

#### Diagram

**Schema Component Representation**

```

<xsd:complexType name="FacilityType">
  <xsd:simpleContent>
    <xsd:extension base=" Scheme ">
      <xsd:attribute name="facilityTypeScheme" type=" xsd:anyURI " default="http://www.fpml.org/coding-scheme/facility-type"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
  
```

top

**Complex Type: Future**

<b>Super-types:</b>	Asset < IdentifiedAsset (by extension) < UnderlyingAsset (by extension) < ExchangeTraded (by extension) < Future (by extension)
<b>Sub-types:</b>	None

<b>Name</b>	Future
<b>Used by (from the same schema document)</b>	Element <b>future</b>
<b>Abstract</b>	no
<b>Documentation</b>	An exchange traded future contract.

**XML Instance Representation**

```

<...
  id=" xsd:ID [0..1]">
  <instrumentId> InstrumentId </instrumentId> [1..*]
  'Identification of the underlying asset, using public and/or private identifiers.'

  <description> xsd:string </description> [0..1]
  'Long name of the underlying asset.'

  <currency> IdentifiedCurrency </currency> [0..1]
  'Trading currency of the underlyer when transacted as a cash instrument.'

  <exchangeId> ExchangeId </exchangeId> [0..1]
  'Identification of the exchange on which this asset is transacted for the purposes
  of calculating a contractual payoff. The term \'Exchange\' is assumed to have the meaning
  as defined in the ISDA 2002 Equity Derivatives Definitions.'

  <clearanceSystem> ClearanceSystem </clearanceSystem> [0..1]
  'Identification of the clearance system associated with the transaction exchange.'

  <definition> ProductReference </definition> [0..1]
  'An optional reference to a full FpML product that defines the simple product in
  greater detail. In case of inconsistency between the terms of the simple product and those
  of the detailed definition, the values in the simple product override those in the
  detailed definition.'

  <relatedExchangeId> ExchangeId </relatedExchangeId> [0..*]
  'A short form unique identifier for a related exchange. If the element is not present then
  the exchange shall be the primary exchange on which listed futures and options on
  the underlying are listed. The term \'Exchange\' is assumed to have the meaning as defined
  in the ISDA 2002 Equity Derivatives Definitions.'

  <optionsExchangeId> ExchangeId </optionsExchangeId> [0..*]
  'A short form unique identifier for an exchange on which the reference option contract
  is listed. This is to address the case where the reference exchange for the future is
  
```

different than the one for the option. The options Exchange is referenced on share options when Merger Elections are selected as Options Exchange Adjustment.'

<multiplier> xsd:positiveInteger </multiplier> [0..1]

'Specifies the contract multiplier that can be associated with the number of units.'

<futureContractReference> xsd:string </futureContractReference> [0..1]

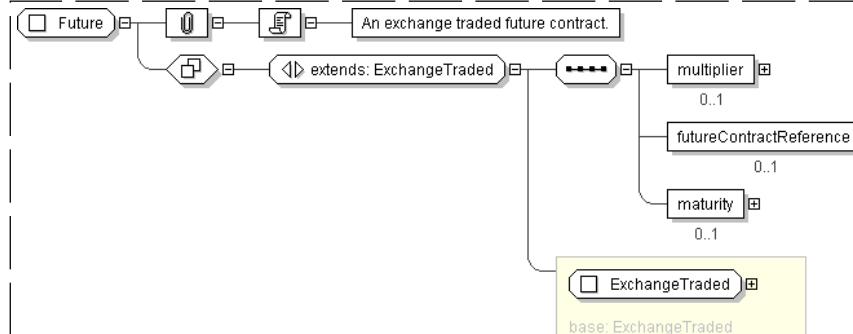
'Specifies the future contract that can be referenced, besides the equity or index reference defined as part of the UnderlyerAsset type.'

<maturity> xsd:date </maturity> [0..1]

'The date when the future contract expires.'

</...>

#### Diagram



#### Schema Component Representation

```

<xsd:complexType name="Future">
  <xsd:complexContent>
    <xsd:extension base=" ExchangeTraded ">
      <xsd:sequence>
        <xsd:element name="multiplier" type=" xsd:positiveInteger " minOccurs="0"/>
        <xsd:element name="futureContractReference" type=" xsd:string " minOccurs="0"/>
        <xsd:element name="maturity" type=" xsd:date " minOccurs="0"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
  
```

[top](#)

## Complex Type: FutureId

Super-types:

[Scheme](#) < FutureId (by extension)

Sub-types:

None

Name

FutureId

Used by (from the same schema document)

Complex Type [Index](#)

Abstract

no

Documentation

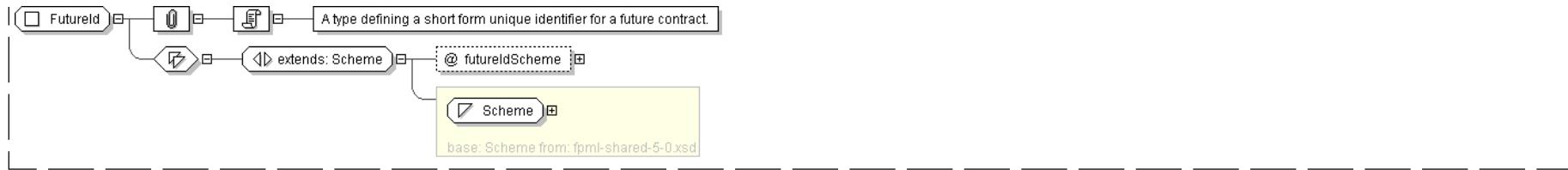
A type defining a short form unique identifier for a future contract.

#### XML Instance Representation

```

<...
  futureIdScheme=" xsd:anyURI [0..1]">
  Scheme
</...>
  
```

#### Diagram



## Schema Component Representation

```
<xsd:complexType name="FutureId">
  <xsd:simpleContent>
    <xsd:extension base="Scheme">
      <xsd:attribute name="futureIdsScheme" type="xsd:anyURI" />
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
```

top

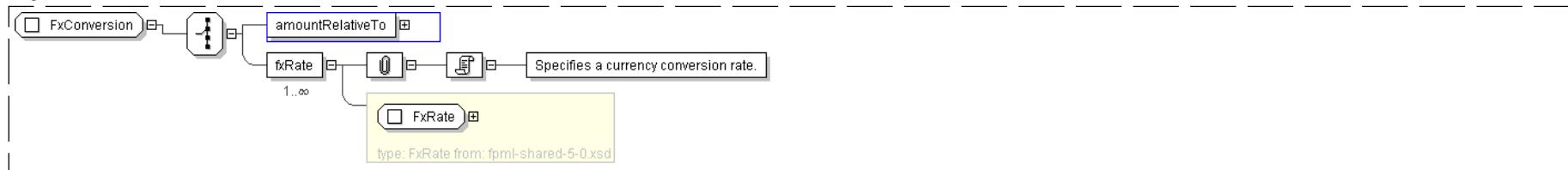
## Complex Type: **FxConversion**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None
<b>Name</b>	FxConversion
<b>Used by (from the same schema document)</b>	Model Group <a href="#">EquityPrice.model</a>
<b>Abstract</b>	no

## XML Instance Representation

```
<...>
Start Choice [1]
  <amountRelativeTo> AmountReference </amountRelativeTo> [1..*]
  <fxRate> FxRate </fxRate> [1..*]
    'Specifies a currency conversion rate.'
End Choice
</...>
```

## Diagram



## Schema Component Representation

```
<xsd:complexType name="FxConversion">
  <xsd:choice>
    <xsd:element name="amountRelativeTo" type=" AmountReference " />
    <xsd:element name="fxRate" type=" FxRate " maxOccurs="unbounded" />
  </xsd:choice>
</xsd:complexType>
```

top

**Complex Type:** `FxRateAsset`

<b>Super-types:</b>	<a href="#">Asset</a> < <a href="#">IdentifiedAsset</a> (by extension) < <a href="#">UnderlyingAsset</a> (by extension) < <b>FxRateAsset</b> (by extension)
<b>Sub-types:</b>	None

Name	FxRateAsset
Used by (from the same schema document)	Element <a href="#">fx</a>
Abstract	no

**XML Instance Representation**

```
<...>
<id="xsd:ID [0..1]">
<instrumentId> InstrumentId </instrumentId> [1..*]
'Identification of the underlying asset, using public and/or private identifiers.'

<description> xsd:string </description> [0..1]
'Long name of the underlying asset.'

<currency> IdentifiedCurrency </currency> [0..1]
'Trading currency of the underlyer when transacted as a cash instrument.'

<exchangeId> ExchangeId </exchangeId> [0..1]
'Identification of the exchange on which this asset is transacted for the purposes of calculating a contractual payoff. The term \"Exchange\" is assumed to have the meaning as defined in the ISDA 2002 Equity Derivatives Definitions.'

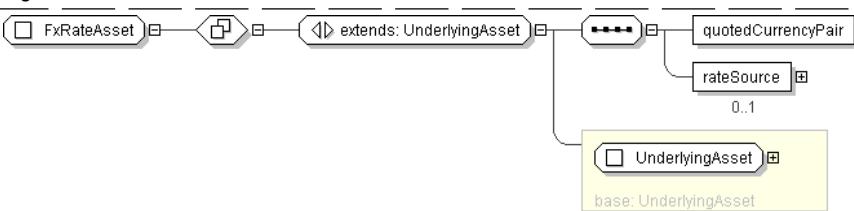
<clearanceSystem> ClearanceSystem </clearanceSystem> [0..1]
'Identification of the clearance system associated with the transaction exchange.'

<definition> ProductReference </definition> [0..1]
'An optional reference to a full FpML product that defines the simple product in greater detail. In case of inconsistency between the terms of the simple product and those of the detailed definition, the values in the simple product override those in the detailed definition.'

<quotedCurrencyPair> QuotedCurrencyPair </quotedCurrencyPair> [1]
'Defines the two currencies for an FX trade and the quotation relationship between the two currencies.'

<rateSource> FxSpotRateSource </rateSource> [0..1]
'Defines the source of the FX rate.'

</...>
```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="FxRateAsset">
  <xsd:complexContent>
    <xsd:extension base=" UnderlyingAsset ">
      <xsd:sequence>
        <xsd:element name="quotedCurrencyPair" type=" QuotedCurrencyPair "/>
        <xsd:element name="rateSource" type=" FxSpotRateSource " minOccurs="0"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
  
```

Super-types:

Sub-types:

- Asset < **IdentifiedAsset** (by extension)
  - [Cash](#) (by extension)
  - [Commodity](#) (by extension)
  - [CurveInstrument](#) (by extension)
  - [UnderlyingAsset](#) (by extension)
    - [Bond](#) (by extension)
      - [ConvertibleBond](#) (by extension)
    - [Deposit](#) (by extension)
    - [ExchangeTraded](#) (by extension)
      - [EquityAsset](#) (by extension)
      - [ExchangeTradedCalculatedPrice](#) (by extension)
        - [ExchangeTradedFund](#) (by extension)
        - [Index](#) (by extension)
      - [ExchangeTradedContract](#) (by extension)
      - [Future](#) (by extension)
    - [FxRateAsset](#) (by extension)
    - [Loan](#) (by extension)
    - [Mortgage](#) (by extension)
    - [MutualFund](#) (by extension)
    - [RateIndex](#) (by extension)
    - [SimpleCreditDefaultSwap](#) (by extension)
    - [SimpleFra](#) (by extension)
    - [SimpleIRSwap](#) (by extension)

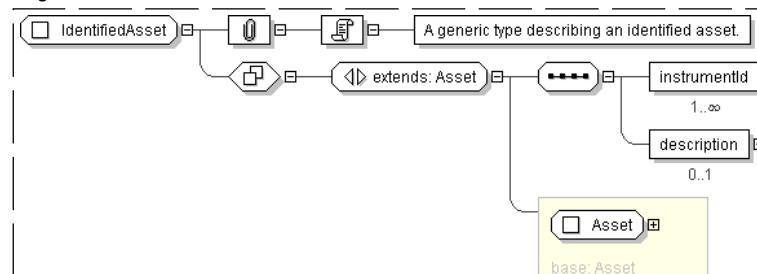
Name	IdentifiedAsset
Abstract	yes
Documentation	A generic type describing an identified asset.

**XML Instance Representation**

```
<...>
  id=" xsd:ID [0..1]"*
  <instrumentId> InstrumentId </instrumentId> [1..*]
  'Identification of the underlying asset, using public and/or private identifiers.'

  <description> xsd:string </description> [0..1]
  'Long name of the underlying asset.'

</...>
```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="IdentifiedAsset" abstract="true">
  <xsd:complexContent>
    <xsd:extension base=" Asset ">
      <xsd:sequence>
        <xsd:element name="instrumentId" type=" InstrumentId " maxOccurs="unbounded" />
        <xsd:element name="description" type=" xsd:string " minOccurs="0" />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
  
```

**Complex Type: Index**

<b>Super-types:</b>	<a href="#">Asset</a> < <a href="#">IdentifiedAsset</a> (by extension) < <a href="#">UnderlyingAsset</a> (by extension) < <a href="#">ExchangeTraded</a> (by extension) < <a href="#">ExchangeTradedCalculatedPrice</a> (by extension) < <a href="#">Index</a> (by extension)
<b>Sub-types:</b>	None

<b>Name</b>	Index
<b>Used by (from the same schema document)</b>	Element <a href="#">index</a>
<b>Abstract</b>	no
<b>Documentation</b>	A published index whose price depends on exchange traded constituents.

**XML Instance Representation**

```

<...
  id=" xsd:ID [0..1]">
    <instrumentId> InstrumentId </instrumentId> [1..*]
      'Identification of the underlying asset, using public and/or private identifiers.'

      <description> xsd:string </description> [0..1]
        'Long name of the underlying asset.'

      <currency> IdentifiedCurrency </currency> [0..1]
        'Trading currency of the underlyer when transacted as a cash instrument.'

      <exchangeId> ExchangeId </exchangeId> [0..1]
        'Identification of the exchange on which this asset is transacted for the purposes
        of calculating a contractual payoff. The term \"Exchange\" is assumed to have the meaning
        as defined in the ISDA 2002 Equity Derivatives Definitions.'

      <clearanceSystem> ClearanceSystem </clearanceSystem> [0..1]
        'Identification of the clearance system associated with the transaction exchange.'

      <definition> ProductReference </definition> [0..1]
        'An optional reference to a full FpML product that defines the simple product in
        greater detail. In case of inconsistency between the terms of the simple product and those
        of the detailed definition, the values in the simple product override those in the
        detailed definition.'

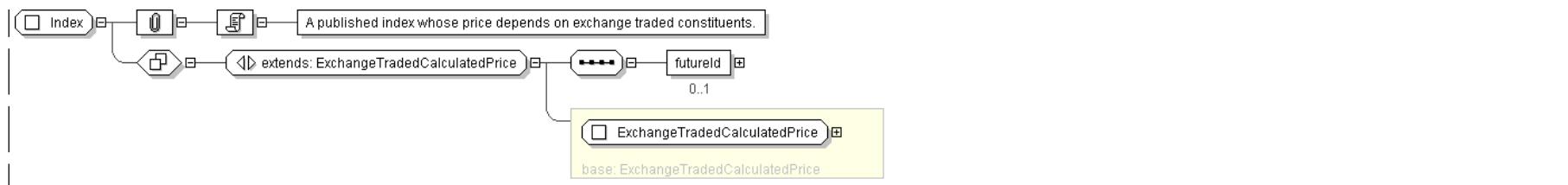
      <relatedExchangeId> ExchangeId </relatedExchangeId> [0..*]
        'A short form unique identifier for a related exchange. If the element is not present then
        the exchange shall be the primary exchange on which listed futures and options on
        the underlying are listed. The term \"Exchange\" is assumed to have the meaning as defined
        in the ISDA 2002 Equity Derivatives Definitions.'

      <optionsExchangeId> ExchangeId </optionsExchangeId> [0..*]
        'A short form unique identifier for an exchange on which the reference option contract
        is listed. This is to address the case where the reference exchange for the future is
        different than the one for the option. The options Exchange is referenced on share options
        when Merger Elections are selected as Options Exchange Adjustment.'

      <constituentExchangeId> ExchangeId </constituentExchangeId> [0..*]
        'Identification of all the exchanges where constituents are traded. The term \"Exchange\""
        is assumed to have the meaning as defined in the ISDA 2002 Equity Derivatives Definitions.'

      <futureId> FutureId </futureId> [0..1]
        'A short form unique identifier for the reference future contract in the case of an
        index underlyer.'
    </...>
  
```

**Diagram**

**Schema Component Representation**

```

<xsd:complexType name="Index">
  <xsd:complexContent>
    <xsd:extension base=" ExchangeTradedCalculatedPrice ">
      <xsd:sequence>
        <xsd:element name="futureId" type=" FutureId " minOccurs="0" />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
  
```

[top](#)**Complex Type: Lien**

<b>Super-types:</b>	<a href="#">Scheme</a> < Lien (by extension)
<b>Sub-types:</b>	None

<b>Name</b>	Lien
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">Loan</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type describing the liens associated with a loan facility.

**XML Instance Representation**

```

<...
  lienScheme=" xsd:anyURI [0..1]>
  Scheme
</...>
  
```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="Lien">
  <xsd:simpleContent>
    <xsd:extension base=" Scheme ">
      <xsd:attribute name="lienScheme" type=" xsd:anyURI " default="http://www.fpml.org/coding-
        scheme/designated-priority"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
  
```

[top](#)**Complex Type: Loan**

<b>Super-types:</b>	<a href="#">Asset</a> < <a href="#">IdentifiedAsset</a> (by extension) < <a href="#">UnderlyingAsset</a> (by extension) < <a href="#">Loan</a> (by extension)
<b>Sub-types:</b>	None

<b>Name</b>	Loan
<b>Used by (from the same schema document)</b>	Element <a href="#">loan</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type describing a loan underlying asset.

**XML Instance Representation**

```

<...
  id="xsd:ID [0..1]">
    <instrumentId> InstrumentId </instrumentId> [1..*]
      'Identification of the underlying asset, using public and/or private identifiers.'

    <description> xsd:string </description> [0..1]
      'Long name of the underlying asset.'

    <currency> IdentifiedCurrency </currency> [0..1]
      'Trading currency of the underlyer when transacted as a cash instrument.'

    <exchangeId> ExchangeId </exchangeId> [0..1]
      'Identification of the exchange on which this asset is transacted for the purposes
       of calculating a contractual payoff. The term \"Exchange\" is assumed to have the meaning
       as defined in the ISDA 2002 Equity Derivatives Definitions.'

    <clearanceSystem> ClearanceSystem </clearanceSystem> [0..1]
      'Identification of the clearance system associated with the transaction exchange.'

    <definition> ProductReference </definition> [0..1]
      'An optional reference to a full FpML product that defines the simple product in
       greater detail. In case of inconsistency between the terms of the simple product and those
       of the detailed definition, the values in the simple product override those in the
       detailed definition.'

Start Choice [0..*]
  'Specifies the borrower. There can be more than one borrower. It is meant to be used in
   the event that there is no Bloomberg Id or the Secured List isn't applicable.'

    <borrower> LegalEntity </borrower> [1]
    <borrowerReference> LegalEntityReference </borrowerReference> [1]

End Choice
  <lien> Lien </lien> [0..1]
    'Specifies the seniority level of the lien.'

    <facilityType> FacilityType </facilityType> [0..1]
    'The type of loan facility (letter of credit, revolving, ...).'

    <maturity> xsd:date </maturity> [0..1]
    'The date when the principal amount of the loan becomes due and payable.'

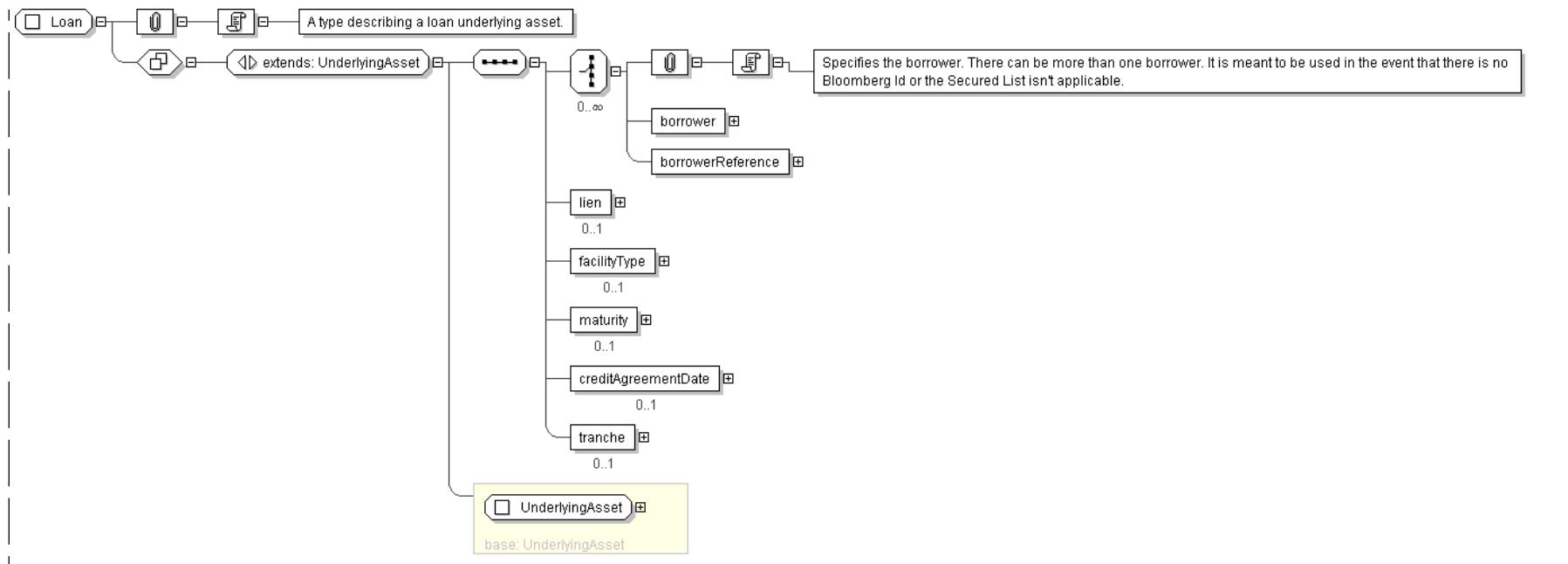
    <creditAgreementDate> xsd:date </creditAgreementDate> [0..1]
    'The credit agreement date is the closing date (the date where the agreement has been
     signed) for the loans in the credit agreement. Funding of the facilities occurs on
     (or sometimes a little after) the Credit Agreement date. This underlyer attribute is used
     to help identify which of the company\'s outstanding loans are being referenced by knowing
     to which credit agreement it belongs. ISDA Standards Terms Supplement term: Date of
     Original Credit Agreement.'

    <tranche> UnderlyingAssetTranche </tranche> [0..1]
    'The loan tranche that is subject to the derivative transaction. It will typically
     be referenced as the Bloomberg tranche number. ISDA Standards Terms Supplement term:
     Bloomberg Tranche Number.'

</...>

```

**Diagram**

**Schema Component Representation**

```

<xsd:complexType name="Loan">
  <xsd:complexContent>
    <xsd:extension base=" UnderlyingAsset ">
      <xsd:sequence>
        <xsd:choice minOccurs="0" maxOccurs="unbounded">
          <xsd:element name="borrower" type=" LegalEntity " />
          <xsd:element name="borrowerReference" type=" LegalEntityReference " />
        </xsd:choice>
        <xsd:element name="lien" type=" Lien " minOccurs="0" />
        <xsd:element name="facilityType" type=" FacilityType " minOccurs="0" />
        <xsd:element name="maturity" type=" xsd:date " minOccurs="0" />
        <xsd:element name="creditAgreementDate" type=" xsd:date " minOccurs="0" />
        <xsd:element name="tranche" type=" UnderlyingAssetTranche " minOccurs="0" />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

top

**Complex Type: Mortgage**

Super-types: Asset &lt; IdentifiedAsset (by extension) &lt; UnderlyingAsset (by extension) &lt; Mortgage (by extension)

Sub-types: None

Name	Mortgage
Used by (from the same schema document)	Element <a href="#">mortgage</a>
Abstract	no
Documentation	A type describing a mortgage asset.

**XML Instance Representation**

```

<...
  id=" xsd:ID [0..1]">
  <instrumentId> InstrumentId </instrumentId> [1..*]
  'Identification of the underlying asset, using public and/or private identifiers.'

```

```

<description> xsd:string </description> [0..1]
'Long name of the underlying asset.'

<currency> IdentifiedCurrency </currency> [0..1]
'Trading currency of the underlyer when transacted as a cash instrument.'

<exchangeId> ExchangeId </exchangeId> [0..1]
'Identification of the exchange on which this asset is transacted for the purposes
of calculating a contractual payoff. The term "Exchange" is assumed to have the meaning
as defined in the ISDA 2002 Equity Derivatives Definitions.'

<clearanceSystem> ClearanceSystem </clearanceSystem> [0..1]
'Identification of the clearance system associated with the transaction exchange.'

<definition> ProductReference </definition> [0..1]
'An optional reference to a full FpML product that defines the simple product in
greater detail. In case of inconsistency between the terms of the simple product and those
of the detailed definition, the values in the simple product override those in the
detailed definition.'

Start Choice [0..1]
'Applicable to the case of default swaps on MBS terms. For specifying the insurer name,
when applicable (when the element is not present, it signifies that the insurer is
Not Applicable)'

<insurer> LegalEntity </insurer> [1]
<insurerReference> LegalEntityReference </insurerReference> [1]
End Choice
Start Choice [0..1]
Specifies the issuer name of a fixed income security or convertible bond. This name can
either be explicitly stated, or specified as an href into another element of the document,
such as the obligor.

<issuerName> xsd:string </issuerName> [1]
<issuerPartyReference> PartyReference </issuerPartyReference> [1]
End Choice
<seniority> CreditSeniority </seniority> [0..1]
'The repayment precedence of a debt instrument.'

<couponType> CouponType </couponType> [0..1]
'Specifies if the bond has a variable coupon, step-up/down coupon or a zero-coupon.'

<couponRate> xsd:decimal </couponRate> [0..1]
'Specifies the coupon rate (expressed in percentage) of a fixed income security or
convertible bond.'

<maturity> xsd:date </maturity> [0..1]
'The date when the principal amount of a security becomes due and payable.'

<paymentFrequency> Period </paymentFrequency> [0..1]
'Specifies the frequency at which the bond pays, e.g. 6M.'

<dayCountFraction> DayCountFraction </dayCountFraction> [0..1]
'The day count basis for the bond.'

<originalPrincipalAmount> xsd:decimal </originalPrincipalAmount> [0..1]
'The initial issued amount of the mortgage obligation.'

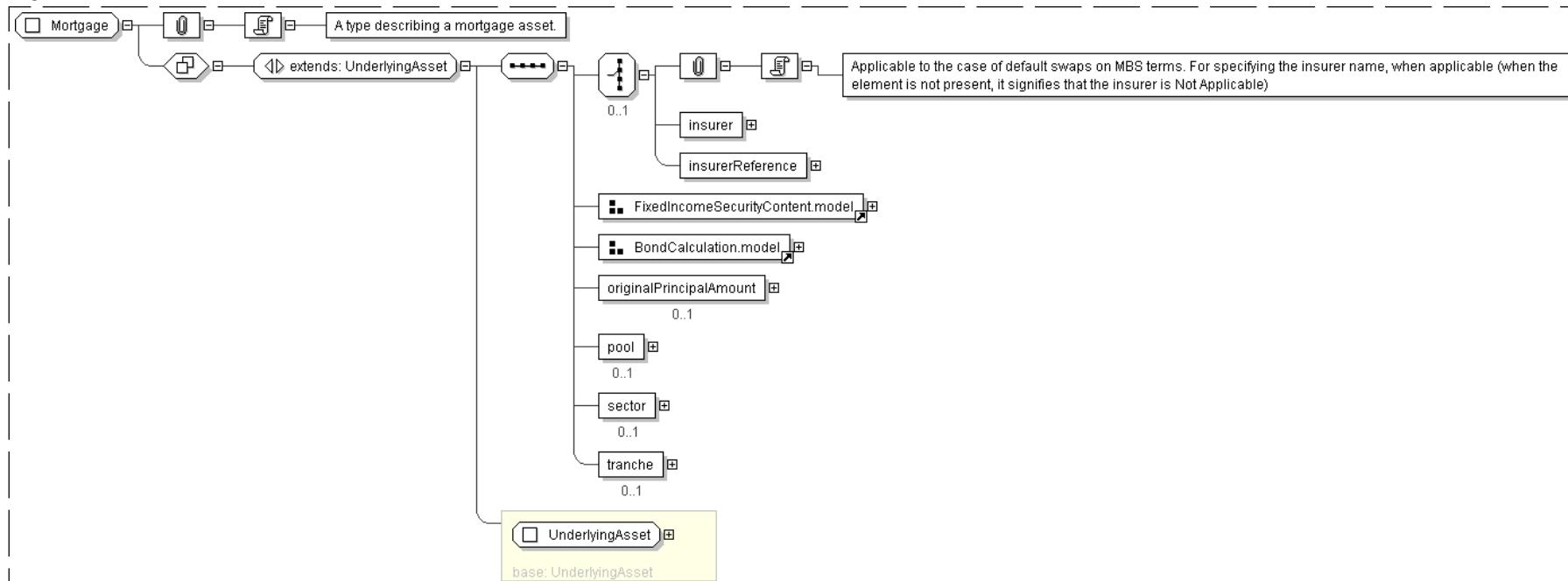
<pool> AssetPool </pool> [0..1]
'The morgage pool that is underneath the mortgage obligation.'

<sector> MortgageSector </sector> [0..1]
'The sector classification of the mortgage obligation.'

<tranche> xsd:token </tranche> [0..1]
'The mortgage obligation tranche that is subject to the derivative transaction.'

```

&lt;/...&gt;

**Diagram****Schema Component Representation**

```

<xsd:complexType name="Mortgage">
  <xsd:complexContent>
    <xsd:extension base=" UnderlyingAsset ">
      <xsd:sequence>
        <xsd:choice minOccurs="0">
          <xsd:element name="insurer" type=" LegalEntity "/>
          <xsd:element name="insurerReference" type=" LegalEntityReference "/>
        </xsd:choice>
        <xsd:group ref=" FixedIncomeSecurityContent.model "/>
        <xsd:group ref=" BondCalculation.model "/>
        <xsd:element name="originalPrincipalAmount" type=" xsd:decimal " minOccurs="0"/>
        <xsd:element name="pool" type=" AssetPool " minOccurs="0"/>
        <xsd:element name="sector" type=" MortgageSector " minOccurs="0"/>
        <xsd:element name="tranche" type=" xsd:token " minOccurs="0"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
  
```

top

**Complex Type: MortgageSector**

Super-types:

[Scheme](#) < **MortgageSector** (by extension)

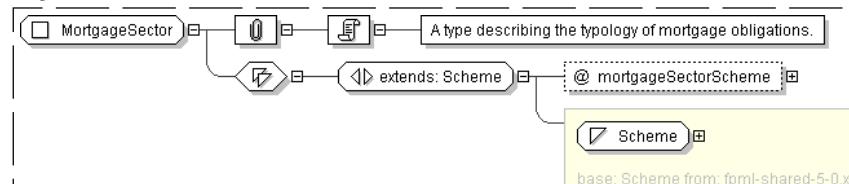
Sub-types:

None

Name	MortgageSector
Used by (from the same schema document)	Complex Type <a href="#">Mortgage</a>
Abstract	no
Documentation	A type describing the typology of mortgage obligations.

**XML Instance Representation**

```
<...
<!-- mortgageSectorScheme=" xsd:anyURI [0..1]">
<!-- Scheme
</...>
```

**Diagram****Schema Component Representation**

```
<xsd:complexType name="MortgageSector">
  <xsd:simpleContent>
    <xsd:extension base=" Scheme ">
      <xsd:attribute name="mortgageSectorScheme" type=" xsd:anyURI " default="http://www.fpml.org/coding-scheme/mortgage-sector"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
```

top

**Complex Type: MutualFund**

<b>Super-types:</b>	Asset < IdentifiedAsset (by extension) < UnderlyingAsset (by extension) < MutualFund (by extension)
<b>Sub-types:</b>	None

<b>Name</b>	MutualFund
<b>Used by (from the same schema document)</b>	Element <a href="#">mutualFund</a>
<b>Abstract</b>	no

**XML Instance Representation**

```
<...
<!-- id=" xsd:ID [0..1]">
<!-- instrumentId> InstrumentId </instrumentId> [1..*]
  'Identification of the underlying asset, using public and/or private identifiers.'

<description> xsd:string </description> [0..1]
  'Long name of the underlying asset.'

<currency> IdentifiedCurrency </currency> [0..1]
  'Trading currency of the underlyer when transacted as a cash instrument.'

<exchangeId> ExchangeId </exchangeId> [0..1]
  'Identification of the exchange on which this asset is transacted for the purposes
  of calculating a contractual payoff. The term \"Exchange\" is assumed to have the meaning
  as defined in the ISDA 2002 Equity Derivatives Definitions.'

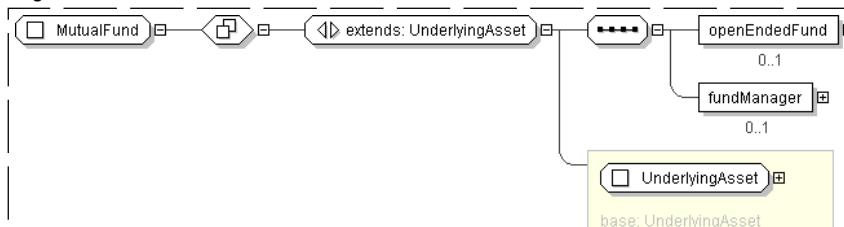
<clearanceSystem> ClearanceSystem </clearanceSystem> [0..1]
  'Identification of the clearance system associated with the transaction exchange.'

<definition> ProductReference </definition> [0..1]
  'An optional reference to a full FpML product that defines the simple product in
  greater detail. In case of inconsistency between the terms of the simple product and those
  of the detailed definition, the values in the simple product override those in the
  detailed definition.'

<openEndedFund> xsd:boolean </openEndedFund> [0..1]
  'Boolean indicator to specify whether the mutual fund is an open-ended mutual fund.'
```

```
<fundManager> xsd:string </fundManager> [0..1]
'Specifies the fund manager that is in charge of the fund.'
```

&lt;/...&gt;

**Diagram****Schema Component Representation**

```

<xsd:complexType name="MutualFund">
  <xsd:complexContent>
    <xsd:extension base=" UnderlyingAsset ">
      <xsd:sequence>
        <xsd:element name="openEndedFund" type=" xsd:boolean " minOccurs="0"/>
        <xsd:element name="fundManager" type=" xsd:string " minOccurs="0"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

[top](#)**Complex Type: PendingPayment**

<b>Super-types:</b>	<a href="#">PaymentBase</a> < <b>PendingPayment</b> (by extension)
<b>Sub-types:</b>	None

<b>Name</b>	PendingPayment
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">BasketConstituent</a> , Complex Type <a href="#">DividendPayout</a> , Complex Type <a href="#">SingleUnderlyer</a>
<b>Abstract</b>	no
<b>Documentation</b>	A structure representing a pending dividend or coupon payment.

**XML Instance Representation**

```

<...
  id=" xsd:ID [0..1]">
  <paymentDate> xsd:date </paymentDate> [1]
  'The date that the dividend or coupon is due.'

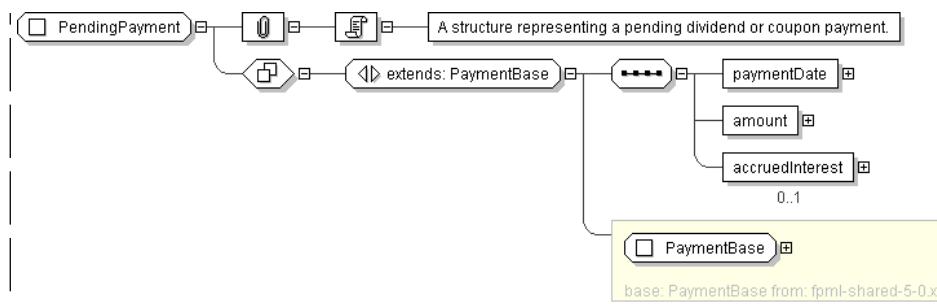
  <amount> Money </amount> [1]
  'The amount of the dividend or coupon payment. Value of dividends or coupon between ex and
  pay date. Stock: if we are between ex-date and pay-date and the dividend is payable under
  the swap, then this should be the ex-div amount * # of securities. Bond: regardless of where
  we are vis-a-vis resets: (coupon % * face of bonds on swap * (bond day count fraction
  using days last coupon pay date of the bond through today).'

  <accruedInterest> Money </accruedInterest> [0..1]
  'Accrued interest on the dividend or coupon payment. When the TRS is structured to pay
  a dividend or coupon on reset after payable date, you may earn interest on these amounts.
  This field indicates the interest accrued on dividend/coupon from pay date to statement
  date. This will only apply to a handful of agreements where dividends are held to the
  next reset AND you receive/pay interest on unpaid amounts.'

</...>

```

**Diagram**

**Schema Component Representation**

```

<xsd:complexType name="PendingPayment">
  <xsd:complexContent>
    <xsd:extension base="PaymentBase">
      <xsd:sequence>
        <xsd:element name="paymentDate" type="xsd:date"/>
        <xsd:element name="amount" type="Money"/>
        <xsd:element name="accruedInterest" type="Money" minOccurs="0"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
  
```

top

**Complex Type: Price**

Super-types:	None
Sub-types:	None

Name	Price
Used by (from the same schema document)	Complex Type <a href="#">BasketConstituent</a>
Abstract	no
Documentation	A type describing the strike price.

**XML Instance Representation**

```

<....>
<commission> Commission </commission> [0..1]
  'This optional component specifies the commission to be charged for executing the
  hedge transactions.'
Start Choice [1]
  <determinationMethod> DeterminationMethod </determinationMethod> [1]
  'Specifies the method according to which an amount or a date is determined.'
Start Group: EquityPrice.model [0..1]
  <grossPrice> ActualPrice </grossPrice> [0..1]
  'Specifies the price of the underlyer, before commissions.'
  <netPrice> ActualPrice </netPrice> [1]
  'Specifies the price of the underlyer, net of commissions.'
  <accruedInterestPrice> xsd:decimal </accruedInterestPrice> [0..1]
  'Specifies the accrued interest that are part of the dirty price in the case of a fixed
  income security or a convertible bond. Expressed in percentage of the notional.'
  <fxConversion> FxConversion </fxConversion> [0..1]
  'Specifies the currency conversion rate that applies to an amount. This rate can either
  be defined elsewhere in the document (case of a quanto swap), or explicitly described
  through this component.'
  
```

```

| End Group: EquityPrice.model
<amountRelativeTo> AmountReference </amountRelativeTo> [1]
'The href attribute value will be a pointer style reference to the element or
component elsewhere in the document where the anchor amount is defined.'

<grossPrice> ActualPrice </grossPrice> [0..1]
'Specifies the price of the underlyer, before commissions.'

<netPrice> ActualPrice </netPrice> [1]
'Specifies the price of the underlyer, net of commissions.'

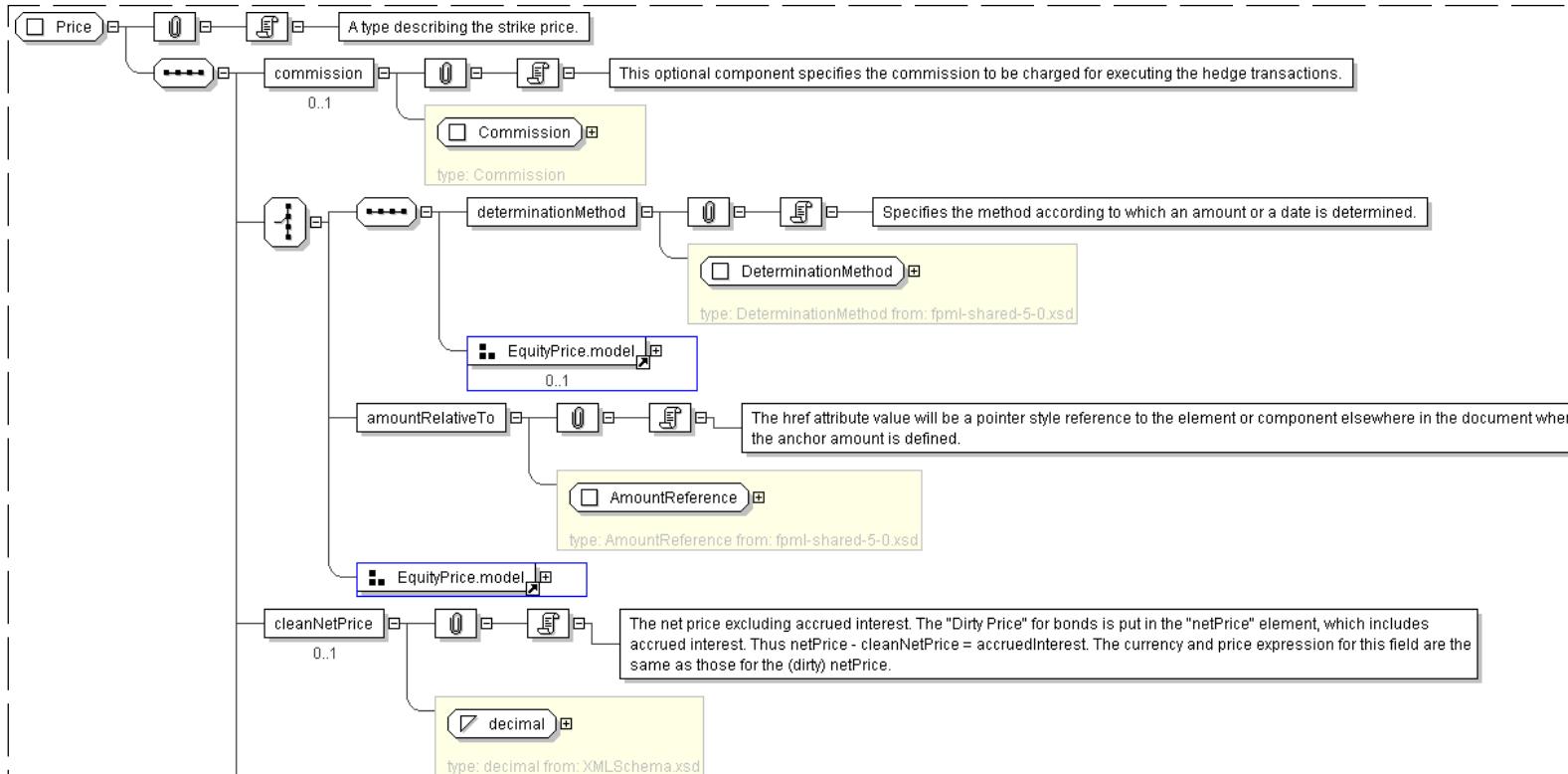
<accruedInterestPrice> xsd:decimal </accruedInterestPrice> [0..1]
'Specifies the accrued interest that are part of the dirty price in the case of a fixed
income security or a convertible bond. Expressed in percentage of the notional.'

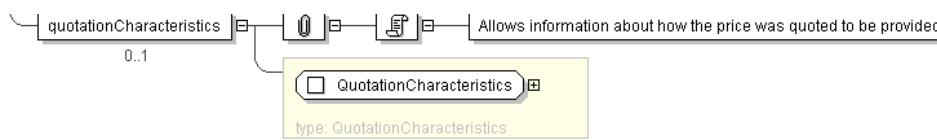
<fxConversion> FxConversion </fxConversion> [0..1]
'Specifies the currency conversion rate that applies to an amount. This rate can either
be defined elsewhere in the document (case of a quanto swap), or explicitly described
through this component.'

End Choice
<cleanNetPrice> xsd:decimal </cleanNetPrice> [0..1]
'The net price excluding accrued interest. The "Dirty Price" for bonds is put in
the "netPrice" element, which includes accrued interest. Thus netPrice - cleanNetPrice
= accruedInterest. The currency and price expression for this field are the same as those
for the (dirty) netPrice.'
<quotationCharacteristics> QuotationCharacteristics </quotationCharacteristics> [0..1]
'Allows information about how the price was quoted to be provided.

</...>

```

**Diagram**

**Schema Component Representation**

```
<xsd:complexType name="Price">
  <xsd:sequence>
    <xsd:element name="commission" type="#Commission" minOccurs="0"/>
    <xsd:choice>
      <xsd:sequence>
        <xsd:element name="determinationMethod" type="#DeterminationMethod"/>
        <xsd:group ref="#EquityPrice.model" minOccurs="0"/>
      </xsd:sequence>
      <xsd:element name="amountRelativeTo" type="#AmountReference"/>
      <xsd:group ref="#EquityPrice.model"/>
    </xsd:choice>
    <xsd:element name="cleanNetPrice" type="xsd:decimal" minOccurs="0"/>
    <xsd:element name="quotationCharacteristics" type="#QuotationCharacteristics" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>
```

[top](#)**Complex Type: PriceQuoteUnits**

**Super-types:** [Scheme](#) < **PriceQuoteUnits** (by extension)

**Sub-types:** None

<b>Name</b>	PriceQuoteUnits
-------------	-----------------

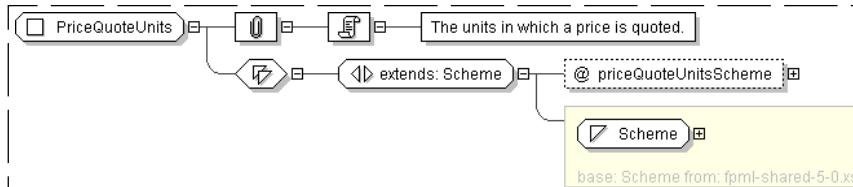
<b>Used by (from the same schema document)</b>	Model Group <a href="#">QuotationCharacteristics.model</a>
--	--

<b>Abstract</b>	no
-----------------	----

<b>Documentation</b>	The units in which a price is quoted.
----------------------	---------------------------------------

**XML Instance Representation**

```
<...
  priceQuoteUnitsScheme="xsd:anyURI [0..1]">
  Scheme
</...>
```

**Diagram****Schema Component Representation**

```
<xsd:complexType name="PriceQuoteUnits">
  <xsd:simpleContent>
    <xsd:extension base="Scheme">
      <xsd:attribute name="priceQuoteUnitsScheme" type="xsd:anyURI" default="http://www.fpml.org/coding-scheme/price-quote-units"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
```

[top](#)

**Complex Type: QuantityUnit**

<b>Super-types:</b>	<a href="#">Scheme</a> < <b>QuantityUnit</b> (by extension)
<b>Sub-types:</b>	None
<b>Name</b>	QuantityUnit
<b>Used by (from the same schema document)</b>	Model Group <a href="#">CommodityReferencePriceFramework.model</a>
<b>Abstract</b>	no
<b>XML Instance Representation</b>	<pre>&lt;...&gt; &lt;quantityUnitscheme=" xsd:anyURI [0..1]"&gt;   Scheme &lt;/...&gt;</pre>
<b>Diagram</b>	<pre> classDiagram     class QuantityUnit {         @ quantityUnitScheme : Scheme     }     class Scheme     QuantityUnit "1" --o "1" Scheme : extends     </pre>
<b>Schema Component Representation</b>	<pre> &lt;xsd:complexType name="QuantityUnit"&gt;   &lt;xsd:simpleContent&gt;     &lt;xsd:extension base=" Scheme "&gt;       &lt;xsd:attribute name="quantityUnitScheme" type=" xsd:anyURI " default="http://www.fpml.org/coding-scheme/price-quote-units"/&gt;     &lt;/xsd:extension&gt;   &lt;/xsd:simpleContent&gt; &lt;/xsd:complexType&gt; </pre>

[top](#)**Complex Type: QuotationCharacteristics**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None
<b>Name</b>	QuotationCharacteristics
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">Price</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type representing a set of characteristics that describe a quotation.
<b>XML Instance Representation</b>	<pre> &lt;...&gt; &lt;measureType&gt; AssetMeasureType &lt;/measureType&gt; [0..1]   'The type of the value that is measured. This could be an NPV, a cash flow, a clean price, etc.'  &lt;quoteUnits&gt; PriceQuoteUnits &lt;/quoteUnits&gt; [0..1]   'The optional units that the measure is expressed in. If not supplied, this is assumed to be a price/value in currency units.'  &lt;side&gt; QuotationSideEnum &lt;/side&gt; [0..1]   'The side (bid/mid/ask) of the measure.'  &lt;currency&gt; Currency &lt;/currency&gt; [0..1]   'The optional currency that the measure is expressed in. If not supplied, this is defaulted from the reportingCurrency in the valuationScenarioDefinition.'  &lt;currencyType&gt; ReportingCurrencyType &lt;/currencyType&gt; [0..1]   'The optional currency that the measure is expressed in. If not supplied, this is defaulted from the reportingCurrency in the valuationScenarioDefinition.' </pre>

**<timing>** *QuoteTiming* **</timing>** [0..1]  
*'When during a day the quote is for. Typically, if this element is supplied, the QuoteLocation needs also to be supplied.'*

Start Group: QuoteLocation.model [0..1]

'Where the quote is from.'

| Start Choice [1]

```
<businessCenter> BusinessCenter </businessCenter> [1]
```

*'A city or other business center.*

<exchangeId> ExchangeId </exchangeId> [1]

*'The exchange (e.g. stock or futures exchange) from which the quote is obtained.'*

End Choice

End Group: QuoteLocation.model

<informationSource> InformationSource </informationSource> [0..\*]

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

<time> xsd:dateTime </time> [0..1]

'When the quote was observed or derived.'

<evaluationDate> xsd:date </val

*'When the quote was computed.'*

```
<expiryTime> xsd:dateTime </expiryTime>
```

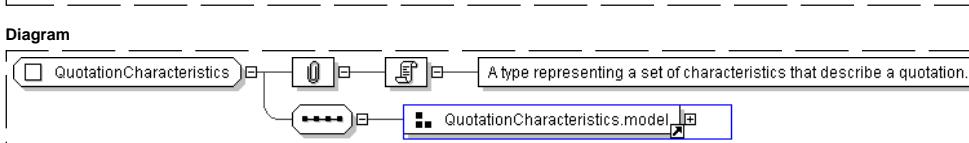
'When does the quote cease to be valid.'

```
<cashflowType> CashflowType </cashflowTy
```

'For cash flows, the type of the cash flows. Examples:

*Fee, Settlement Fee, Brokerage Fee, etc.'*

| </ . . . >



## Schema Component Representation

```
<xsd:complexType name="QuotationCharacteristics">
  <xsd:sequence>
    <xsd:group ref=" QuotationCharacteristics.model "/>
  </xsd:sequence>
</xsd:complexType>
```

top

## Complex Type: **QuoteTiming**

<b>Super-types:</b>	<a href="#">Scheme</a> < <b>QuoteTiming</b> (by extension)
<b>Sub-types:</b>	None
<b>Name</b>	QuoteTiming
<b>Used by (from the same schema document)</b>	Model Group <a href="#">QuotationCharacteristics.model</a>
<b>Abstract</b>	no
<b>Documentation</b>	The type of the time of the quote

## XML Instance Representation

5

```
quoteTimingScheme=" xsd:anyURI [0..1]">
  scheme
</...>
```

## Diagram



## Schema Component Representation

```
<xsd:complexType name="QuoteTiming">
  <xsd:simpleContent>
    <xsd:extension base=" Scheme ">
      <xsd:attribute name="quoteTimingScheme" type=" xsd:anyURI " default="http://www.fpml.org/
        coding-scheme/quote-timing"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
```

top

## Complex Type: RateIndex

Super-types:	<a href="#">Asset</a> < <a href="#">IdentifiedAsset</a> (by extension) < <a href="#">UnderlyingAsset</a> (by extension) < <b>RateIndex</b> (by extension)
Sub-types:	None

Name	RateIndex
Used by (from the same schema document)	Element <a href="#">rateIndex</a>
Abstract	no

## XML Instance Representation

```
<...
  id=" xsd:ID [0..1]">
  <instrumentId> InstrumentId </instrumentId> [1..*]
  'Identification of the underlying asset, using public and/or private identifiers.'

  <description> xsd:string </description> [0..1]
  'Long name of the underlying asset.'

  <currency> IdentifiedCurrency </currency> [0..1]
  'Trading currency of the underlyer when transacted as a cash instrument.'

  <exchangeId> ExchangeId </exchangeId> [0..1]
  'Identification of the exchange on which this asset is transacted for the purposes
  of calculating a contractual payoff. The term \"Exchange\" is assumed to have the meaning
  as defined in the ISDA 2002 Equity Derivatives Definitions.'

  <clearanceSystem> ClearanceSystem </clearanceSystem> [0..1]
  'Identification of the clearance system associated with the transaction exchange.'

  <definition> ProductReference </definition> [0..1]
  'An optional reference to a full FpML product that defines the simple product in
  greater detail. In case of inconsistency between the terms of the simple product and those
  of the detailed definition, the values in the simple product override those in the
  detailed definition.'

  <floatingRateIndex> FloatingRateIndex </floatingRateIndex> [1]
  <term> Period </term> [1]
  'Specifies the term of the simple swap, e.g. 5Y.'
```

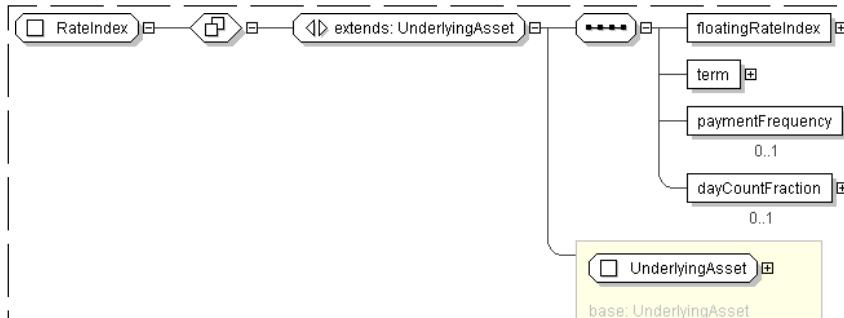
```

<paymentFrequency> Period </paymentFrequency> [0..1]
'Specifies the frequency at which the index pays, e.g. 6M.'

<dayCountFraction> DayCountFraction </dayCountFraction> [0..1]
'The day count basis for the index.'

</...>

```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="RateIndex">
  <xsd:complexContent>
    <xsd:extension base=" UnderlyingAsset ">
      <xsd:sequence>
        <xsd:element name="floatingRateIndex" type=" FloatingRateIndex "/>
        <xsd:element name="term" type=" Period "/>
        <xsd:element name="paymentFrequency" type=" Period " minOccurs="0"/>
        <xsd:element name="dayCountFraction" type=" DayCountFraction " minOccurs="0"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

top

**Complex Type: ReportingCurrencyType**

**Super-types:** [Scheme](#) < **ReportingCurrencyType** (by extension)  
**Sub-types:** None

<b>Name</b>	ReportingCurrencyType
<b>Used by (from the same schema document)</b>	Model Group <a href="#">QuotationCharacteristics.model</a>
<b>Abstract</b>	no
<b>Documentation</b>	A scheme identifying the type of currency that was used to report the value of an asset. For example, this could contain values like SettlementCurrency, QuoteCurrency, UnitCurrency, etc.

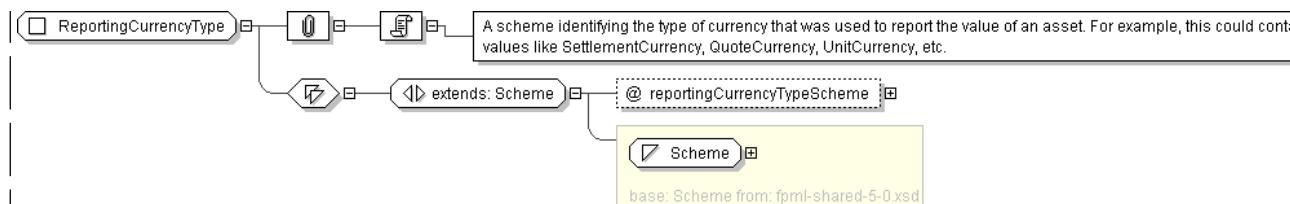
**XML Instance Representation**

```

<...
  reportingCurrencyTypeScheme=" xsd:anyURI [0..1]">
  Scheme
</...>

```

**Diagram**

**Schema Component Representation**

```

<xsd:complexType name="ReportingCurrencyType">
  <xsd:simpleContent>
    <xsd:extension base=" Scheme ">
      <xsd:attribute name="reportingCurrencyTypeScheme" type=" xsd:anyURI " default="http://www.
        fpml.org/coding-scheme/reporting-currency-type"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
  
```

top

**Complex Type: SimpleCreditDefaultSwap**

<b>Super-types:</b>	Asset < IdentifiedAsset (by extension) < UnderlyingAsset (by extension) < SimpleCreditDefaultSwap (by extension)
<b>Sub-types:</b>	None

<b>Name</b>	SimpleCreditDefaultSwap
<b>Used by (from the same schema document)</b>	Element <a href="#">simpleCreditDefaultSwap</a>
<b>Abstract</b>	no

**XML Instance Representation**

```

<...
  id=" xsd:ID [0..1]">
    <instrumentId> InstrumentId </instrumentId> [1..*]
    'Identification of the underlying asset, using public and/or private identifiers.'

    <description> xsd:string </description> [0..1]
    'Long name of the underlying asset.'

    <currency> IdentifiedCurrency </currency> [0..1]
    'Trading currency of the underlyer when transacted as a cash instrument.'

    <exchangeId> ExchangeId </exchangeId> [0..1]
    'Identification of the exchange on which this asset is transacted for the purposes
    of calculating a contractual payoff. The term \"Exchange\" is assumed to have the meaning
    as defined in the ISDA 2002 Equity Derivatives Definitions.'

    <clearanceSystem> ClearanceSystem </clearanceSystem> [0..1]
    'Identification of the clearance system associated with the transaction exchange.'

    <definition> ProductReference </definition> [0..1]
    'An optional reference to a full FpML product that defines the simple product in
    greater detail. In case of inconsistency between the terms of the simple product and those
    of the detailed definition, the values in the simple product override those in the
    detailed definition.'
  
```

```

Start Choice [1]
  <referenceEntity> LegalEntity </referenceEntity> [1]
  'The entity for which this is defined.'

  <creditEntityReference> LegalEntityReference </creditEntityReference> [1]
  'An XML reference a credit entity defined elsewhere in the document.'
  
```

End Choice

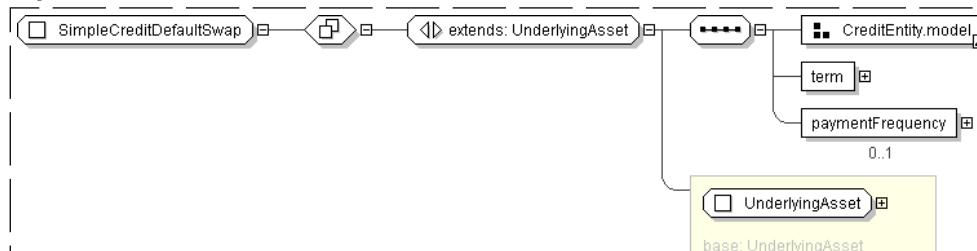
<term> Period </term> [1]

'Specifies the term of the simple CD swap, e.g. 5Y.'

<paymentFrequency> Period </paymentFrequency> [0..1]

'Specifies the frequency at which the swap pays, e.g. 6M.'

&lt;/...&gt;

**Diagram****Schema Component Representation**

```

<xsd:complexType name="SimpleCreditDefaultSwap">
  <xsd:complexContent>
    <xsd:extension base="UnderlyingAsset ">
      <xsd:sequence>
        <xsd:group ref="CreditEntity.model ">
          <xsd:element name="term" type="Period ">
          <xsd:element name="paymentFrequency" type="Period " minOccurs="0"/>
        </xsd:sequence>
      </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>
  
```

top

**Complex Type: SimpleFra**

<b>Super-types:</b>	Asset < <u>IdentifiedAsset</u> (by extension) < <u>UnderlyingAsset</u> (by extension) < <b>SimpleFra</b> (by extension)
<b>Sub-types:</b>	None

<b>Name</b>	SimpleFra
<b>Used by (from the same schema document)</b>	Element <a href="#">simpleFra</a>
<b>Abstract</b>	no

**XML Instance Representation**

```

<...
  id=" xsd:ID [0..1]">
  <instrumentId> InstrumentId </instrumentId> [1..*]
  'Identification of the underlying asset, using public and/or private identifiers.'

  <description> xsd:string </description> [0..1]
  'Long name of the underlying asset.'

  <currency> IdentifiedCurrency </currency> [0..1]
  'Trading currency of the underlyer when transacted as a cash instrument.'

  <exchangeId> ExchangeId </exchangeId> [0..1]
  'Identification of the exchange on which this asset is transacted for the purposes
  of calculating a contractual payoff. The term "Exchange" is assumed to have the meaning
  as defined in the ISDA 2002 Equity Derivatives Definitions.'

  <clearanceSystem> ClearanceSystem </clearanceSystem> [0..1]
  'Identification of the clearance system associated with the transaction exchange.'
  
```

```

<definition> ProductReference </definition> [0..1]
'An optional reference to a full FpML product that defines the simple product in
greater detail. In case of inconsistency between the terms of the simple product and those
of the detailed definition, the values in the simple product override those in the
detailed definition.'

<startTerm> Period </startTerm> [1]
'Specifies the start term of the simple fra, e.g. 3M.'

<endTerm> Period </endTerm> [1]
'Specifies the end term of the simple fra, e.g. 9M.'

<dayCountFraction> DayCountFraction </dayCountFraction> [0..1]
'The day count basis for the FRA.'

</...>

```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="SimpleFra">
  <xsd:complexContent>
    <xsd:extension base=" UnderlyingAsset ">
      <xsd:sequence>
        <xsd:element name="startTerm" type=" Period "/>
        <xsd:element name="endTerm" type=" Period "/>
        <xsd:element name="dayCountFraction" type=" DayCountFraction " minOccurs="0"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

top

**Complex Type: SimpleIRSwap**

<b>Super-types:</b>	Asset < IdentifiedAsset (by extension) < UnderlyingAsset (by extension) < SimpleIRSwap (by extension)
<b>Sub-types:</b>	None

<b>Name</b>	SimpleIRSwap
<b>Used by (from the same schema document)</b>	Element <a href="#">simpleIRSwap</a>
<b>Abstract</b>	no

**XML Instance Representation**

```

<...
  id=" xsd:ID [0..1]">
  <instrumentId> InstrumentId </instrumentId> [1..*]
    'Identification of the underlying asset, using public and/or private identifiers.'

  <description> xsd:string </description> [0..1]
    'Long name of the underlying asset.'

  <currency> IdentifiedCurrency </currency> [0..1]
    'Trading currency of the underlyer when transacted as a cash instrument.'

```

```

<exchangeId> ExchangeId </exchangeId> [0..1]
'Identification of the exchange on which this asset is transacted for the purposes
of calculating a contractual payoff. The term "Exchange" is assumed to have the meaning
as defined in the ISDA 2002 Equity Derivatives Definitions.'

<clearanceSystem> ClearanceSystem </clearanceSystem> [0..1]
'Identification of the clearance system associated with the transaction exchange.'

<definition> ProductReference </definition> [0..1]
'An optional reference to a full FpML product that defines the simple product in
greater detail. In case of inconsistency between the terms of the simple product and those
of the detailed definition, the values in the simple product override those in the
detailed definition.'

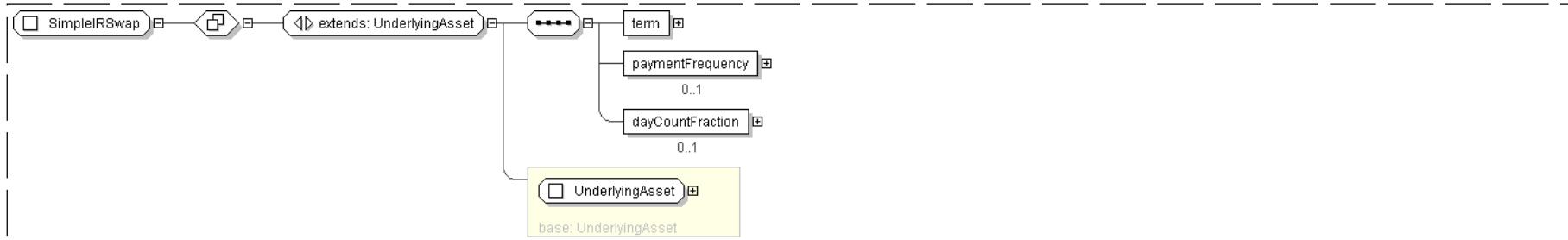
<term> Period </term> [1]
'Specifies the term of the simple swap, e.g. 5Y.'

<paymentFrequency> Period </paymentFrequency> [0..1]
'Specifies the frequency at which the swap pays, e.g. 6M.'

<dayCountFraction> DayCountFraction </dayCountFraction> [0..1]
'The day count basis for the swap.'

</...>

```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="SimpleIRSwap">
  <xsd:complexContent>
    <xsd:extension base=" UnderlyingAsset ">
      <xsd:sequence>
        <xsd:element name="term" type=" Period "/>
        <xsd:element name="paymentFrequency" type=" Period " minOccurs="0"/>
        <xsd:element name="dayCountFraction" type=" DayCountFraction " minOccurs="0"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

top

**Complex Type: SingleUnderlyer**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None

Name	SingleUnderlyer
Used by (from the same schema document)	Complex Type <a href="#">Underlyer</a>
Abstract	no
Documentation	A type describing a single underlyer

## XML Instance Representation

```
<...>
<underlyingAsset> ... </underlyingAsset> [1]
<openUnits> xsd:decimal </openUnits> [0..1]
```

'The number of units (index or securities) that constitute the underlyer of the swap. In the case of a basket swap, this element is used to reference both the number of basket units, and the number of each asset components of the basket when these are expressed in absolute terms.'

```
<dividendPayout> DividendPayout </dividendPayout> [0..1]
```

'Specifies the dividend payout ratio associated with an equity underlyer. A basket swap can have different payout ratios across the various underlying constituents. In certain cases the actual ratio is not known on trade inception, and only general conditions are then specified. Users should note that FpML makes a distinction between the derivative contract and the underlyer of the contract. It would be better if the agreed dividend payout on a derivative contract was modelled at the level of the derivative contract, an approach which may be adopted in the next major version of FpML.'

```
<couponPayment> PendingPayment </couponPayment> [0..1]
```

'The next upcoming coupon payment.'

```
<averageDailyTradingVolume> AverageDailyTradingVolumeLimit </averageDailyTradingVolume> [0..1]
```

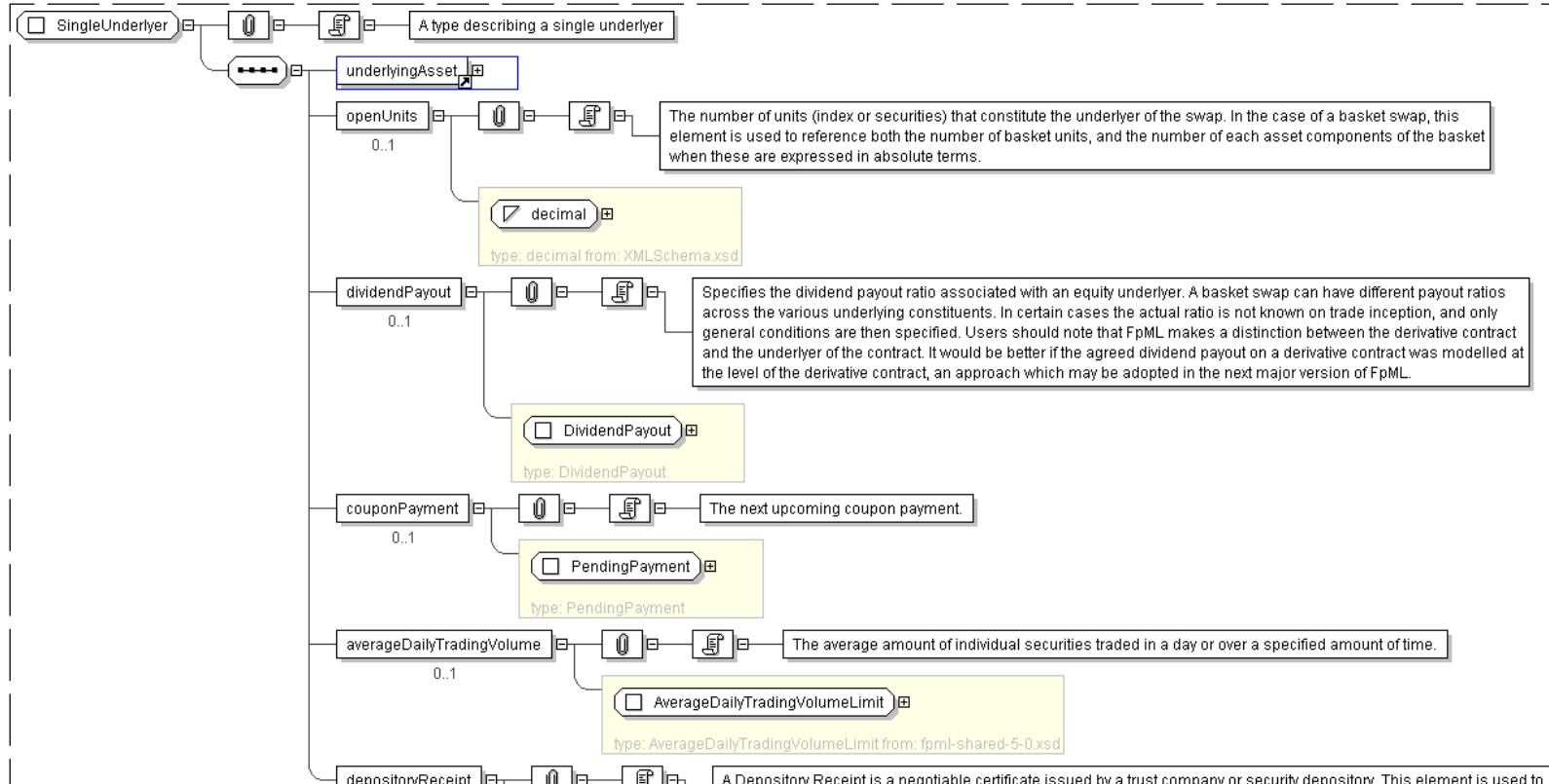
'The average amount of individual securities traded in a day or over a specified amount of time.'

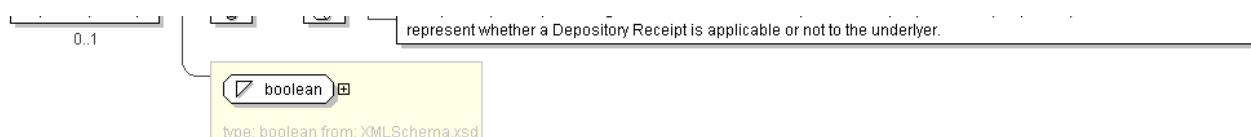
```
<depositoryReceipt> xsd:boolean </depositoryReceipt> [0..1]
```

'A Depository Receipt is a negotiable certificate issued by a trust company or security depository. This element is used to represent whether a Depository Receipt is applicable or not to the underlyer.'

```
</...>
```

## Diagram



**Schema Component Representation**

```

<xsd:complexType name="SingleUnderlyer">
  <xsd:sequence>
    <xsd:element ref=" underlyingAsset " />
    <xsd:element name="openUnits" type=" xsd:decimal " minOccurs="0" />
    <xsd:element name="dividendPayout" type=" DividendPayout " minOccurs="0" />
    <xsd:element name="couponPayment" type=" PendingPayment " minOccurs="0" />
    <xsd:element name="averageDailyTradingVolume" type=" AverageDailyTradingVolumeLimit "
      " minOccurs="0" />
    <xsd:element name="depositoryReceipt" type=" xsd:boolean " minOccurs="0" />
  </xsd:sequence>
</xsd:complexType>

```

top

**Complex Type: TimeZone**

**Super-types:** Scheme < TimeZone (by extension)  
**Sub-types:** None

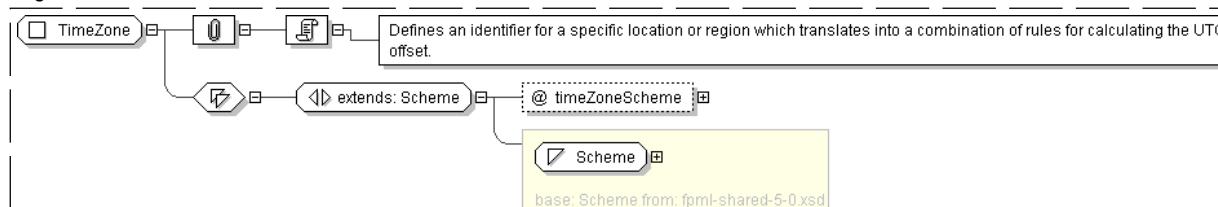
<b>Name</b>	TimeZone
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">CommodityBusinessCalendarTime</a>
<b>Abstract</b>	no
<b>Documentation</b>	Defines an identifier for a specific location or region which translates into a combination of rules for calculating the UTC offset.

**XML Instance Representation**

```

<...
  timeZoneScheme=" xsd:anyURI [0..1]">
  Scheme
</...>

```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="TimeZone">
  <xsd:simpleContent>
    <xsd:extension base=" Scheme ">
      <xsd:attribute name="timeZoneScheme" type=" xsd:anyURI " />
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>

```

top

**Complex Type: Underlyer**

**Super-types:** None

Sub-types:

None

Name	Underlyer
Abstract	no
Documentation	A type describing the whole set of possible underlyers: single underlyers or multiple underlyers, each of these having either security or index components.

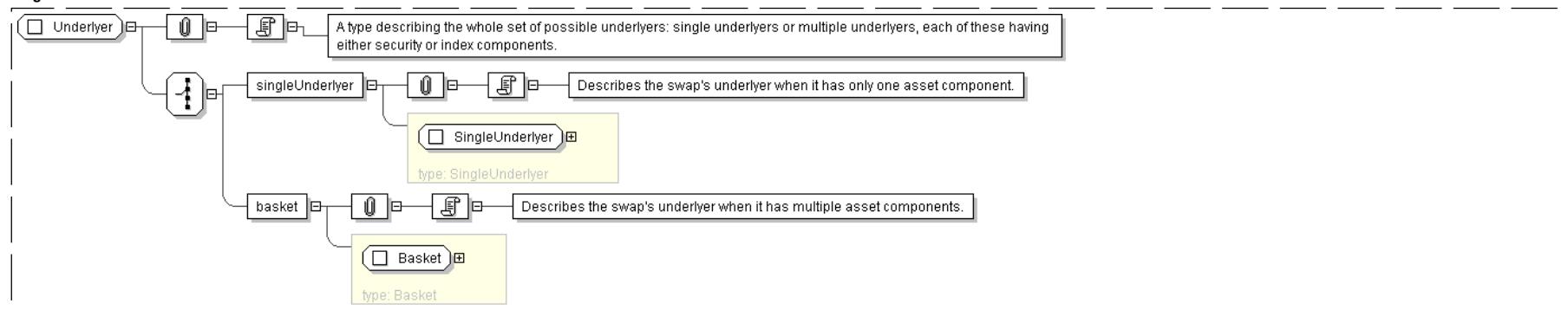
## XML Instance Representation

```
<...>
Start Choice [1]
<singleUnderlyer> SingleUnderlyer </singleUnderlyer> [1]
'Describes the swap\'s underlyer when it has only one asset component.'

<basket> Basket </basket> [1]
'Describes the swap\'s underlyer when it has multiple asset components.'

End Choice
</...>
```

## Diagram



## Schema Component Representation

```

<xsd:complexType name="Underlyer">
  <xsd:choice>
    <xsd:element name="singleUnderlyer" type=" SingleUnderlyer " />
    <xsd:element name="basket" type=" Basket " />
  </xsd:choice>
</xsd:complexType>
  
```

top

## Complex Type: UnderlyingAsset

Super-types:

Asset &lt; IdentifiedAsset (by extension) &lt; UnderlyingAsset (by extension)

Sub-types:

- [Bond](#) (by extension)
  - [ConvertibleBond](#) (by extension)
- [Deposit](#) (by extension)
- [ExchangeTraded](#) (by extension)
  - [EquityAsset](#) (by extension)
  - [ExchangeTradedCalculatedPrice](#) (by extension)
    - [ExchangeTradedFund](#) (by extension)
    - [Index](#) (by extension)
  - [ExchangeTradedContract](#) (by extension)
  - [Future](#) (by extension)
- [FxRateAsset](#) (by extension)
- [Loan](#) (by extension)
- [Mortgage](#) (by extension)
- [MutualFund](#) (by extension)
- [RateIndex](#) (by extension)
- [SimpleCreditDefaultSwap](#) (by extension)
- [SimpleFra](#) (by extension)

- SimpleIRSwap (by extension)

Name	UnderlyingAsset
Abstract	yes
Documentation	Abstract base class for all underlying assets.

## XML Instance Representation

```

<...
  id=" xsd:ID [0..1]">
  <instrumentId> InstrumentId </instrumentId> [1..*]
  'Identification of the underlying asset, using public and/or private identifiers.'

  <description> xsd:string </description> [0..1]
  'Long name of the underlying asset.'

  <currency> IdentifiedCurrency </currency> [0..1]
  'Trading currency of the underlyer when transacted as a cash instrument.'

  <exchangeId> ExchangeId </exchangeId> [0..1]
  'Identification of the exchange on which this asset is transacted for the purposes
  of calculating a contractual payoff. The term \"Exchange\" is assumed to have the meaning
  as defined in the ISDA 2002 Equity Derivatives Definitions.'

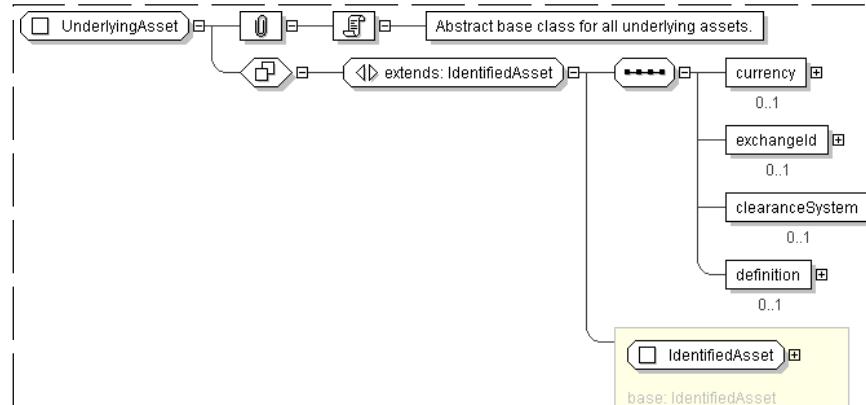
  <clearanceSystem> ClearanceSystem </clearanceSystem> [0..1]
  'Identification of the clearance system associated with the transaction exchange.'

  <definition> ProductReference </definition> [0..1]
  'An optional reference to a full FpML product that defines the simple product in
  greater detail. In case of inconsistency between the terms of the simple product and those
  of the detailed definition, the values in the simple product override those in the
  detailed definition.'

</...>

```

## Diagram



## Schema Component Representation

```

<xsd:complexType name="UnderlyingAsset" abstract="true">
  <xsd:complexContent>
    <xsd:extension base=" IdentifiedAsset ">
      <xsd:sequence>
        <xsd:element name="currency" type=" IdentifiedCurrency " minOccurs="0"/>
        <xsd:element name="exchangeId" type=" ExchangeId " minOccurs="0"/>
        <xsd:element name="clearanceSystem" type=" ClearanceSystem " minOccurs="0"/>
        <xsd:element name="definition" type=" ProductReference " minOccurs="0"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

**Complex Type: UnderlyingAssetTranche**

Super-types:	<a href="#">Scheme</a> < <b>UnderlyingAssetTranche</b> (by extension)
Sub-types:	None

Name	UnderlyingAssetTranche
Used by (from the same schema document)	Complex Type <a href="#">Loan</a>
Abstract	no

**XML Instance Representation**

```
<...>
<loanTrancheScheme=" xsd:anyURI [0..1]">
<Scheme>
</...>
```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="UnderlyingAssetTranche">
  <xsd:simpleContent>
    <xsd:extension base=" Scheme ">
      <xsd:attribute name="loanTrancheScheme" type=" xsd:anyURI " default="http://www.fpml.org/
        coding-scheme/underlying-asset-tranche"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
  
```

**Model Group: BasketIdentifier.model**

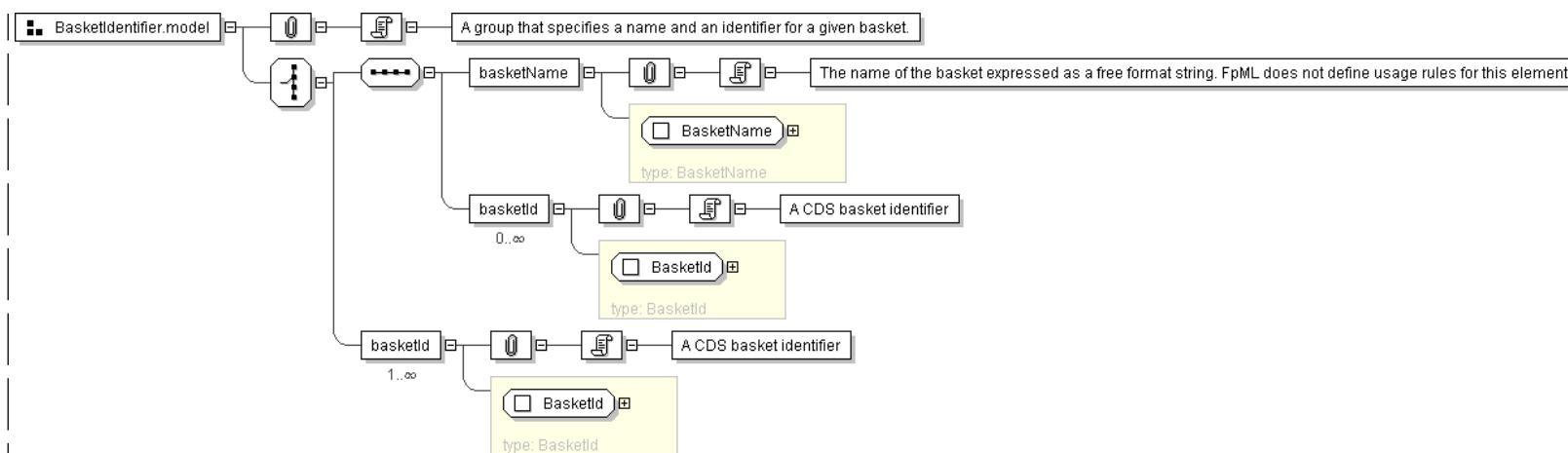
Name	BasketIdentifier.model
Used by (from the same schema document)	Complex Type <a href="#">Basket</a>
Documentation	A group that specifies a name and an identifier for a given basket.

**XML Instance Representation**

```

Start Choice [1]
  <basketName> BasketName </basketName> [1]
  'The name of the basket expressed as a free format string. FpML does not define usage rules
  for this element.'
  <basketId> BasketId </basketId> [0..*]
  'A CDS basket identifier'
  <basketId> BasketId </basketId> [1..*]
  'A CDS basket identifier'
End Choice
  
```

**Diagram**

**Schema Component Representation**

```

<xsd:group name="BasketIdentifier.model">
  <xsd:choice>
    <xsd:sequence>
      <xsd:element name="basketName" type=" BasketName " />
      <xsd:element name="basketId" type=" BasketId " minOccurs="0" maxOccurs="unbounded" />
    </xsd:sequence>
    <xsd:element name="basketId" type=" BasketId " maxOccurs="unbounded" />
  </xsd:choice>
</xsd:group>
  
```

[top](#)**Model Group: BondCalculation.model**

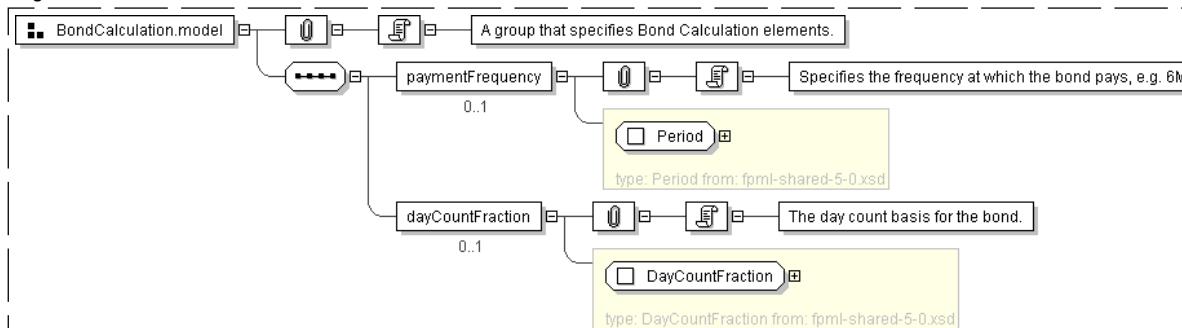
Name	BondCalculation.model
Used by (from the same schema document)	Complex Type <a href="#">Bond</a> , Complex Type <a href="#">Mortgage</a>
Documentation	A group that specifies Bond Calculation elements.

**XML Instance Representation**

```

<paymentFrequency> Period </paymentFrequency> [0..1]
'Specifies the frequency at which the bond pays, e.g. 6M.'

<dayCountFraction> DayCountFraction </dayCountFraction> [0..1]
'The day count basis for the bond.'
  
```

**Diagram****Schema Component Representation**

```

<xsd:group name="BondCalculation.model">
  <xsd:sequence>
    <xsd:element name="paymentFrequency" type=" Period " minOccurs="0"/>
    <xsd:element name="dayCountFraction" type=" DayCountFraction " minOccurs="0"/>
  </xsd:sequence>
</xsd:group>

```

[top](#)

## Model Group: BondChoice.model

**Name** BondChoice.model

**Documentation** A model group which provides choices between all bond underlyers.

### XML Instance Representation

```

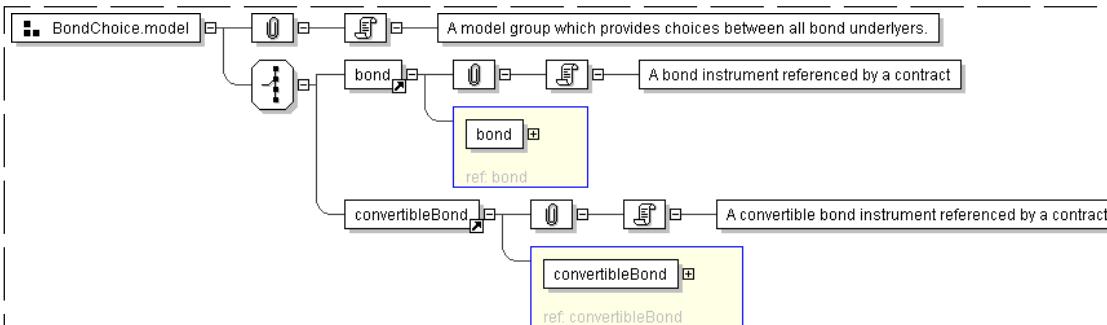
Start Choice [1]
<bond> ... </bond> [1]
'A bond instrument referenced by a contract'

<convertibleBond> ... </convertibleBond> [1]
'A convertible bond instrument referenced by a contract.'

```

End Choice

### Diagram



### Schema Component Representation

```

<xsd:group name="BondChoice.model">
  <xsd:choice>
    <xsd:element ref=" bond "/>
    <xsd:element ref=" convertibleBond "/>
  </xsd:choice>
</xsd:group>

```

[top](#)

## Model Group: CommodityProduct.model

**Name** CommodityProduct.model

**Used by (from the same schema document)** Complex Type **Commodity**

**Documentation** A group used to specify details of a commodity underlyer.

### XML Instance Representation

```

Start Group: CommodityReferencePriceFramework.model [0..1]
<commodityBase> CommodityBase </commodityBase> [1]
'A coding scheme value to identify the base type of the commodity being traded. Where
possible, this should follow the naming convention used in the 2005 ISDA Commodity
Definitions. For example, \'Oil\'.'

```

```

<commodityDetails> CommodityDetails </commodityDetails> [1]
'A coding scheme value to identify the commodity being traded more specifically.
Where possible, this should follow the naming convention used in the 2005 ISDA
Commodity Definitions. For example, \'Brent\'.'
```

<unit> QuantityUnit </unit> [1]
'A coding scheme value to identify the unit in which the undelyer is denominated.
Where possible, this should follow the naming convention used in the 2005 ISDA
Commodity Definitions.'

<currency> Currency </currency> [1]
'The currency in which the Commodity Reference Price is published.'

Start Choice [1]

<exchangeId> ExchangeId </exchangeId> [1]
'For those commodities being traded with reference to the price of a listed future,
the exchange where that future is listed should be specified here.'

<publication> InformationSource </publication> [1]
'For those commodities being traded with reference to a price distributed by a
publication, that publication should be specified here.'

End Choice

End Group: CommodityReferencePriceFramework\_model

<specifiedPrice> SpecifiedPriceEnum </specifiedPrice> [1]
'The Specified Price is not defined in the Commodity Reference Price and so needs to be
stated in the Underlyer definition as it will impact the calculation of the Floating Price.'

Start Sequence [0..1]

Start Choice [1]

<deliveryDates> DeliveryDatesEnum </deliveryDates> [1]
'The Delivery Date is a NearbyMonth, for use when the Commodity Transaction references
Futures Contract.'

<deliveryDate> AdjustableDate </deliveryDate> [1]
'The Delivery Date is a fixed, single day.'

<deliveryDateYearMonth> xsd:gYearMonth </deliveryDateYearMonth> [1]
'The Delivery Date is a fixed, single month.'

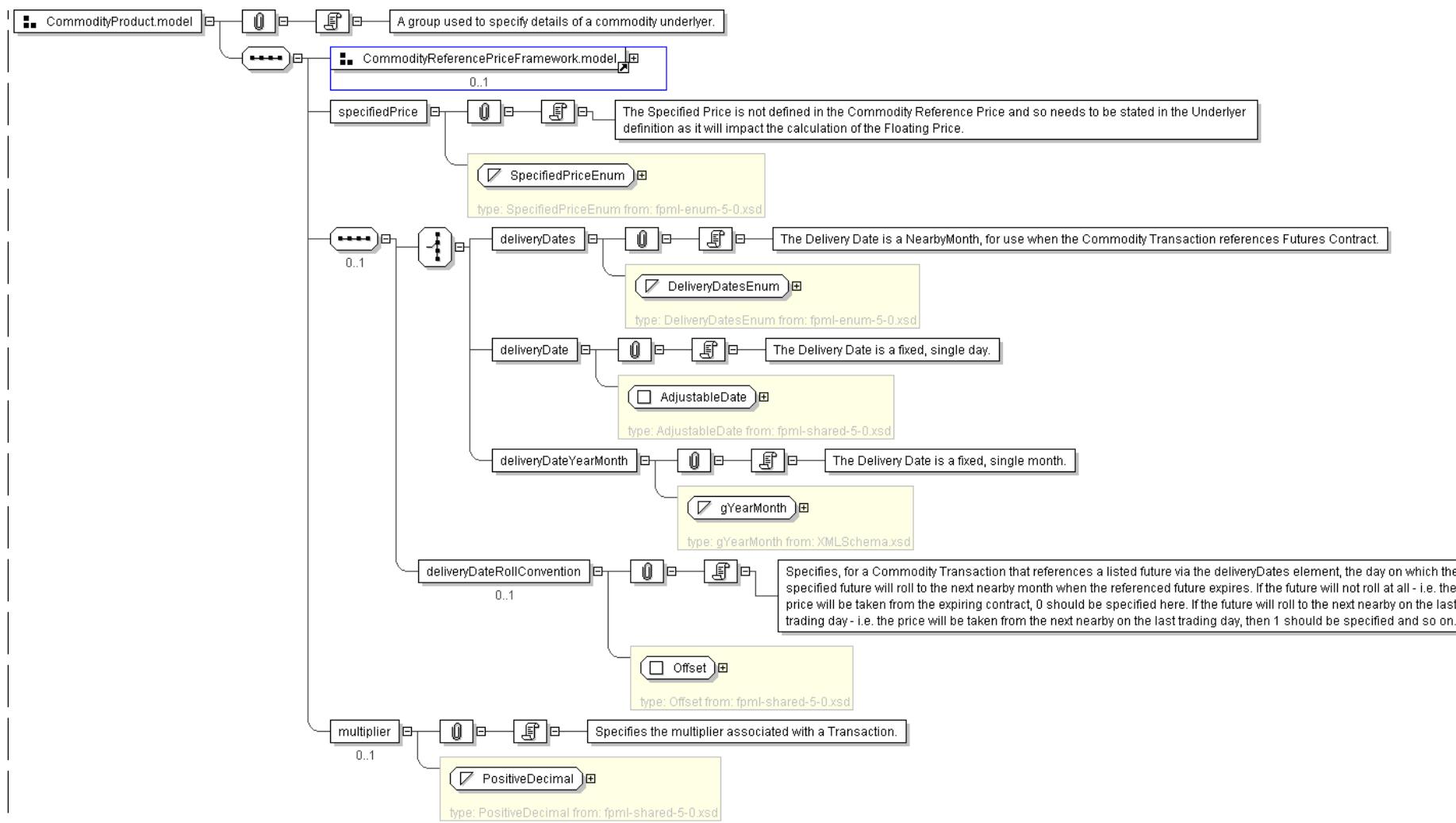
End Choice

<deliveryDateRollConvention> Offset </deliveryDateRollConvention> [0..1]
'Specifies, for a Commodity Transaction that references a listed future via the
deliveryDates element, the day on which the specified future will roll to the next nearby
month when the referenced future expires. If the future will not roll at all - i.e. the
price will be taken from the expiring contract, 0 should be specified here. If the future
will roll to the next nearby on the last trading day - i.e. the price will be taken from
the next nearby on the last trading day, then 1 should be specified and so on.'

End Sequence

<multiplier> PositiveDecimal </multiplier> [0..1]
'Specifies the multiplier associated with a Transaction.'

**Diagram**

**Schema Component Representation**

```

<xsd:group name="CommodityProduct.model">
  <xsd:sequence>
    <xsd:group ref="# CommodityReferencePriceFramework.model " minOccurs="0"/>
    <xsd:element name="specifiedPrice" type=" SpecifiedPriceEnum " />
    <xsd:sequence minOccurs="0">
      <xsd:choice>
        <xsd:element name="deliveryDates" type=" DeliveryDatesEnum " />
        <xsd:element name="deliveryDate" type=" AdjustableDate " />
        <xsd:element name="deliveryDateYearMonth" type=" xsd:gYearMonth " />
      </xsd:choice>
      <xsd:element name="deliveryDateRollConvention" type=" Offset " minOccurs="0"/>
    </xsd:sequence>
    <xsd:element name="multiplier" type=" PositiveDecimal " minOccurs="0"/>
  </xsd:sequence>
</xsd:group>

```

Name	CommodityReferencePriceFramework.model
Used by (from the same schema document)	Model Group <a href="#">CommodityProduct.model</a>
Documentation	A group used to specify the commodity underlyer in the event that no ISDA Commodity Reference Price exists.

**XML Instance Representation**

```
<commodityBase> CommodityBase </commodityBase> [1]
'A coding scheme value to identify the base type of the commodity being traded. Where possible, this should follow the naming convention used in the 2005 ISDA Commodity Definitions. For example, \'Oil\'.' 

<commodityDetails> CommodityDetails </commodityDetails> [1]
'A coding scheme value to identify the commodity being traded more specifically. Where possible, this should follow the naming convention used in the 2005 ISDA Commodity Definitions. For example, \'Brent\'.' 

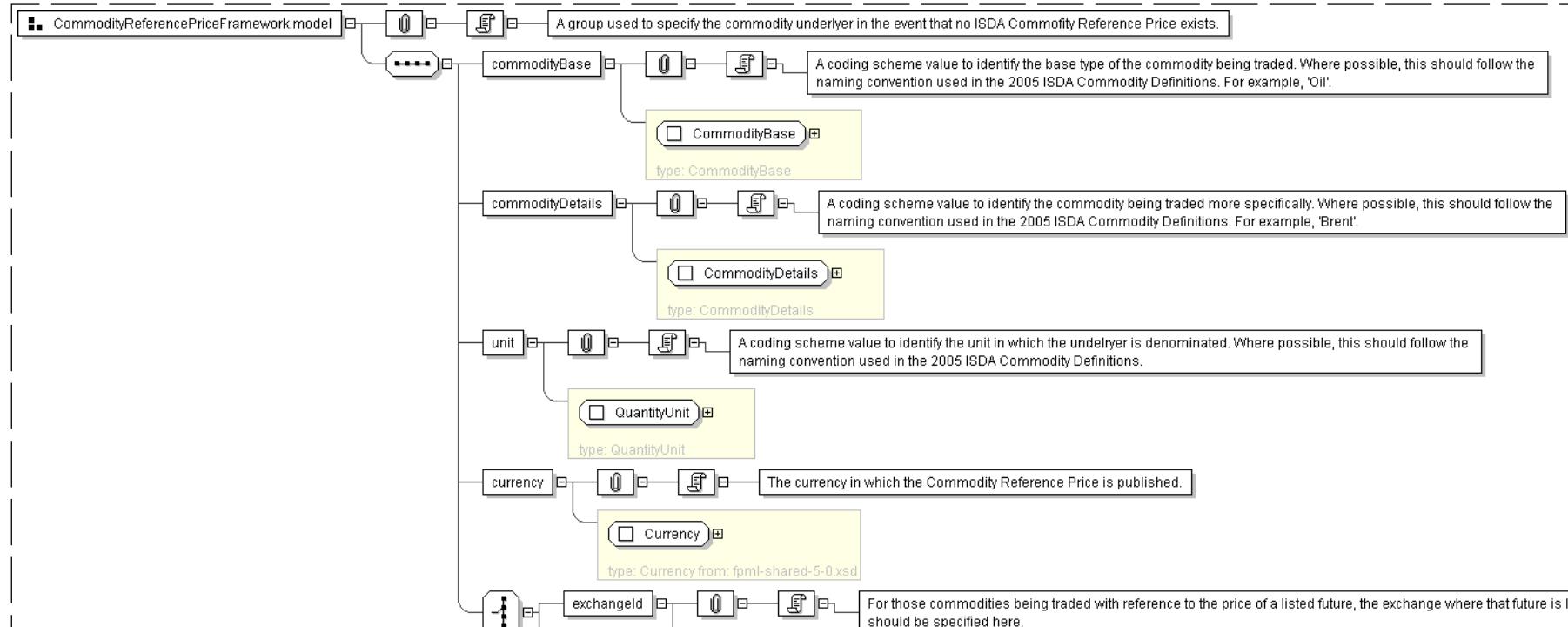
<unit> QuantityUnit </unit> [1]
'A coding scheme value to identify the unit in which the undelyer is denominated. Where possible, this should follow the naming convention used in the 2005 ISDA Commodity Definitions.' 

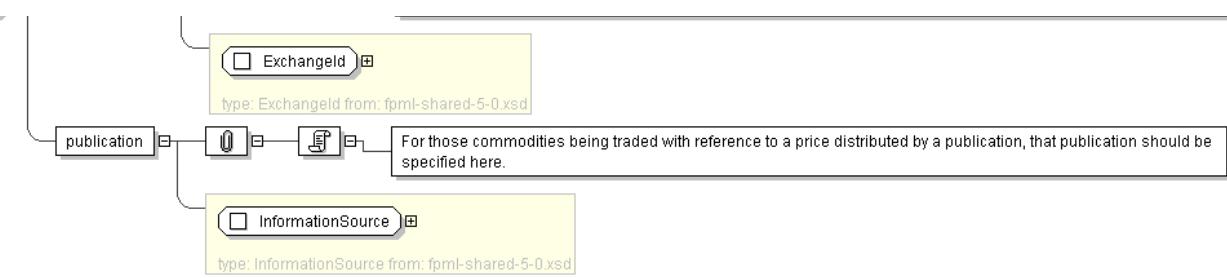
<currency> Currency </currency> [1]
'The currency in which the Commodity Reference Price is published.' 

Start Choice [1]
<exchangeId> ExchangeId </exchangeId> [1]
'For those commodities being traded with reference to the price of a listed future, the exchange where that future is listed should be specified here.' 

<publication> InformationSource </publication> [1]
'For those commodities being traded with reference to a price distributed by a publication, that publication should be specified here.' 

End Choice
```

**Diagram**

[top](#)

## Model Group: CreditEntity.model

Name	CreditEntity.model
Used by (from the same schema document)	Complex Type <b>SimpleCreditDefaultSwap</b>
Documentation	An item which has credit characteristics that can be modeled, e.g. a firm, index, or region.

### XML Instance Representation

```

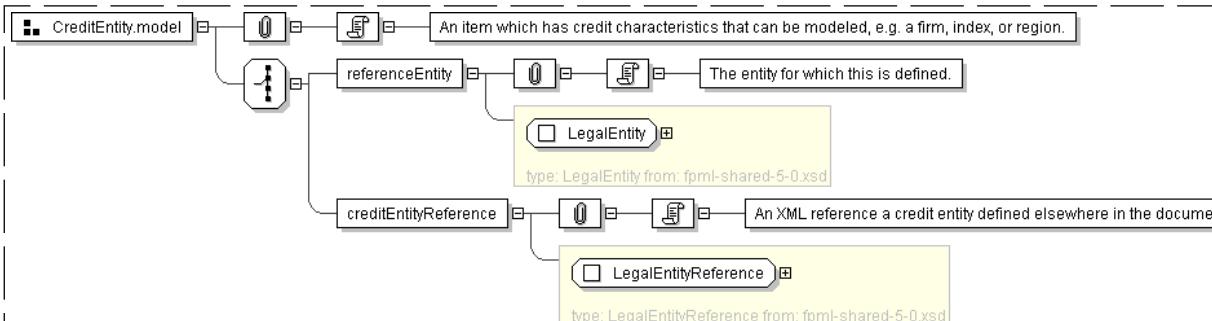
Start Choice [1]
<referenceEntity> LegalEntity </referenceEntity> [1]
'The entity for which this is defined.'

<creditEntityReference> LegalEntityReference </creditEntityReference> [1]
'An XML reference a credit entity defined elsewhere in the document.'

```

End Choice

### Diagram



### Schema Component Representation

```

<xsd:group name="CreditEntity.model">
  <xsd:choice>
    <xsd:element name="referenceEntity" type="LegalEntity" />

```

```
<xsd:element name="creditEntityReference" type=" LegalEntityReference " />
</xsd:choice>
</xsd:group>
```

Model Group: **EquityPrice.model**

<b>Name</b>	EquityPrice.model
<b>Used by (from the same schema document)</b>	Complex Type <b>Price</b> , Complex Type <b>Price</b>

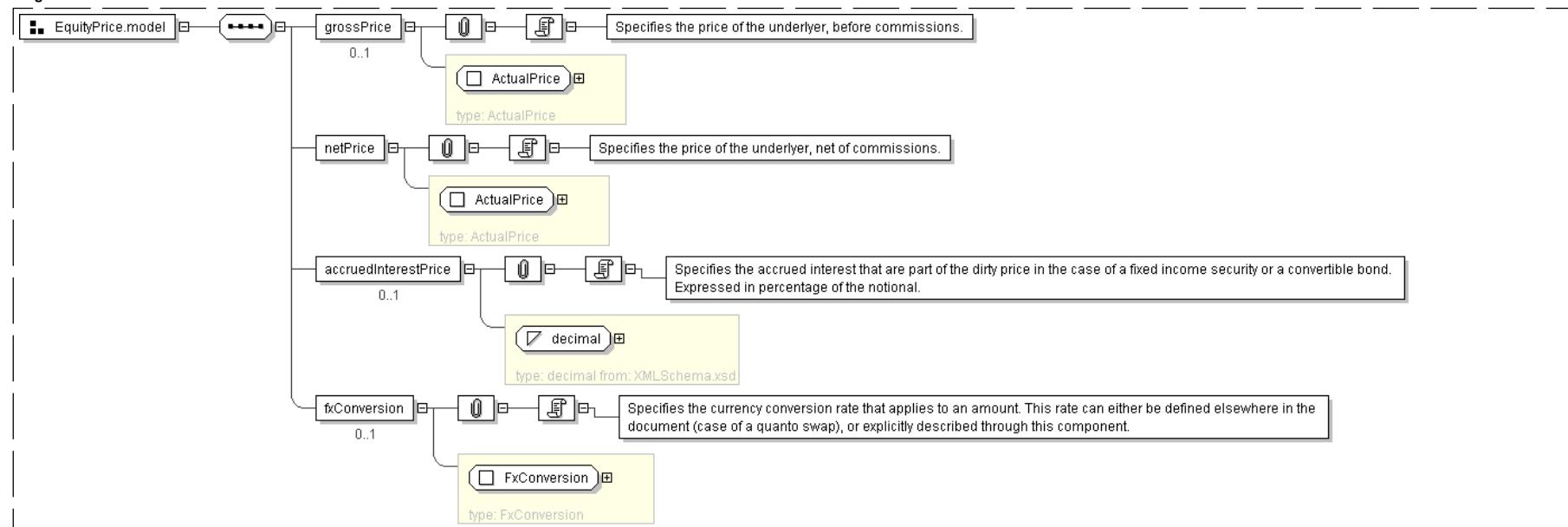
**XML Instance Representation**

```
<grossPrice> ActualPrice </grossPrice> [0..1]
'Specifies the price of the underlyer, before commissions.'

<netPrice> ActualPrice </netPrice> [1]
'Specifies the price of the underlyer, net of commissions.'

<accruedInterestPrice> xsd:decimal </accruedInterestPrice> [0..1]
'Specifies the accrued interest that are part of the dirty price in the case of a fixed
income security or a convertible bond. Expressed in percentage of the notional.'

<fxConversion> FxConversion </fxConversion> [0..1]
'Specifies the currency conversion rate that applies to an amount. This rate can either
be defined elsewhere in the document (case of a quanto swap), or explicitly described
through this component.'
```

**Diagram****Schema Component Representation**

```
<xsd:group name="EquityPrice.model">
  <xsd:sequence>
    <xsd:element name="grossPrice" type=" ActualPrice " minOccurs="0"/>
    <xsd:element name="netPrice" type=" ActualPrice "/>
    <xsd:element name="accruedInterestPrice" type=" xsd:decimal " minOccurs="0"/>
    <xsd:element name="fxConversion" type=" FxConversion " minOccurs="0"/>
  </xsd:sequence>
</xsd:group>
```

**Model Group: ExchangeIdentifier.model**

<b>Name</b>	ExchangeIdentifier.model
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">ExchangeTraded</a>

**XML Instance Representation**

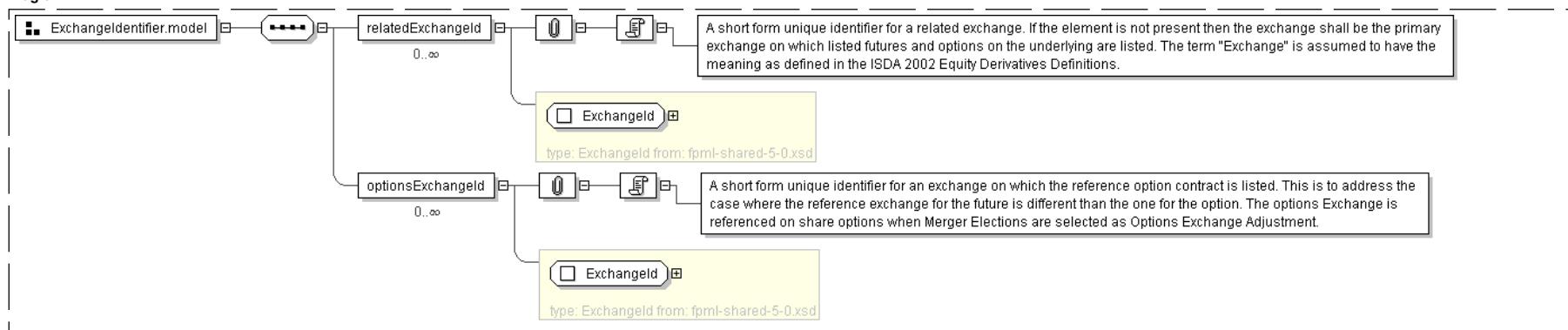
```
<relatedExchangeId> ExchangeID </relatedExchangeId> [0..*]
```

'A short form unique identifier for a related exchange. If the element is not present then the exchange shall be the primary exchange on which listed futures and options on the underlying are listed. The term \"Exchange\" is assumed to have the meaning as defined in the ISDA 2002 Equity Derivatives Definitions.'

```
<optionsExchangeId> ExchangeID </optionsExchangeId> [0..*]
```

'A short form unique identifier for an exchange on which the reference option contract is listed. This is to address the case where the reference exchange for the future is different than the one for the option. The options Exchange is referenced on share options when Merger Elections are selected as Options Exchange Adjustment.'

**Diagram****Schema Component Representation**

```

<xsd:group name="ExchangeIdentifier.model">
  <xsd:sequence>
    <xsd:element name="relatedExchangeId" type=" ExchangeID " minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element name="optionsExchangeId" type=" ExchangeID " minOccurs="0" maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:group>
  
```

**Model Group: FixedIncomeSecurityContent.model**

<b>Name</b>	FixedIncomeSecurityContent.model
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">Bond</a> , Complex Type <a href="#">Mortgage</a>
<b>Documentation</b>	

**XML Instance Representation**

```
Start Choice [0..1]
```

'Specifies the issuer name of a fixed income security or convertible bond. This name can either be explicitly stated, or specified as an href into another element of the document, such as the obligor.'

```

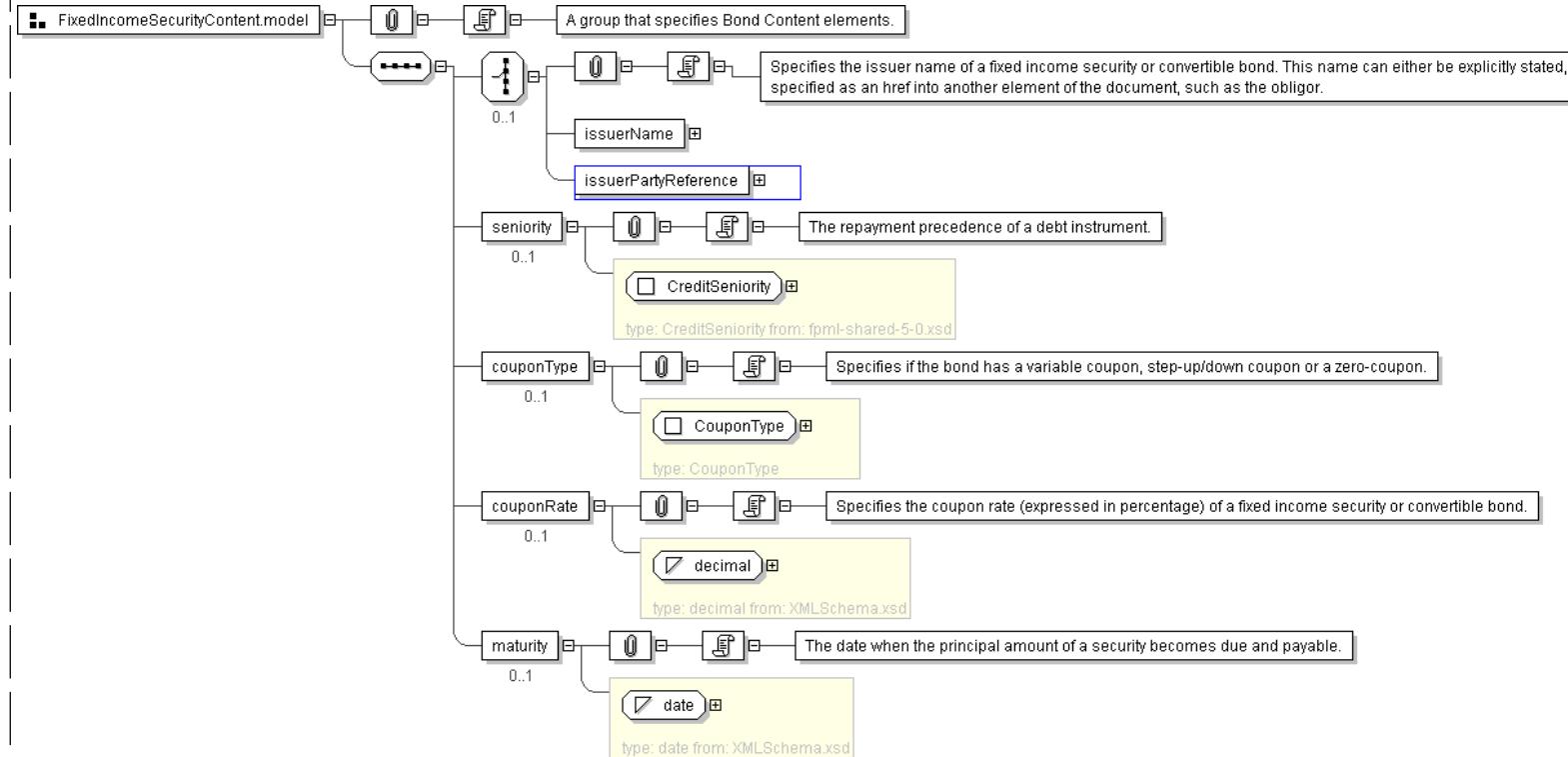
<issuerName> xsd:string </issuerName> [1]
<issuerPartyReference> PartyReference </issuerPartyReference> [1]
End Choice
  
```

```
'<seniority> CreditSeniority </seniority> [0..1]
'The repayment precedence of a debt instrument.'

<couponType> CouponType </couponType> [0..1]
'Specifies if the bond has a variable coupon, step-up/down coupon or a zero-coupon.'

<couponRate> xsd:decimal </couponRate> [0..1]
'Specifies the coupon rate (expressed in percentage) of a fixed income security or convertible bond.'

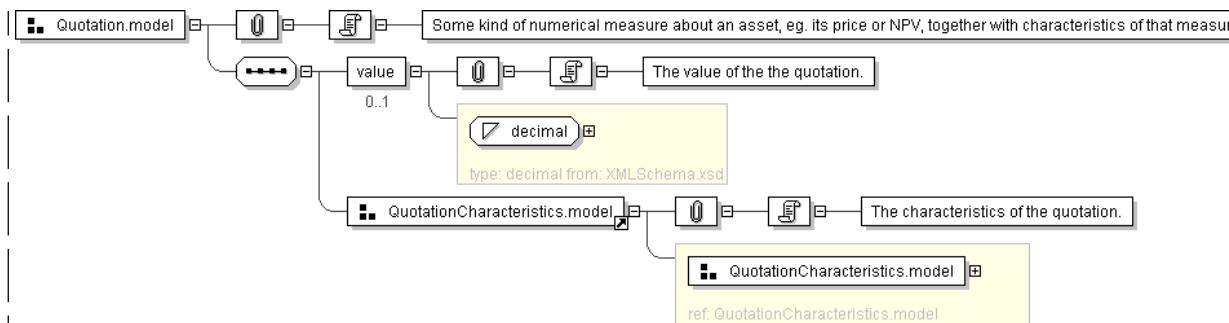
<maturity> xsd:date </maturity> [0..1]
'The date when the principal amount of a security becomes due and payable.'
```

**Diagram****Schema Component Representation**

```
<xsd:group name="FixedIncomeSecurityContent.model">
  <xsd:sequence>
    <xsd:choice minOccurs="0">
      <xsd:element name="issuerName" type="xsd:string" />
      <xsd:element name="issuerPartyReference" type="PartyReference" />
    </xsd:choice>
    <xsd:element name="seniority" type="CreditSeniority" minOccurs="0"/>
    <xsd:element name="couponType" type="CouponType" minOccurs="0"/>
    <xsd:element name="couponRate" type="xsd:decimal" minOccurs="0"/>
    <xsd:element name="maturity" type="xsd:date" minOccurs="0"/>
  </xsd:sequence>
</xsd:group>
```

<b>Name</b>	Quotation.model
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">BasicQuotation</a>
<b>Documentation</b>	Some kind of numerical measure about an asset, eg. its price or NPV, together with characteristics of that measure.
<b>XML Instance Representation</b>	
<pre>&lt;value&gt; xsd:decimal &lt;/value&gt; [0..1] 'The value of the the quotation.'</pre>	
<pre>&lt;measureType&gt; AssetMeasureType &lt;/measureType&gt; [0..1] 'The type of the value that is measured. This could be an NPV, a cash flow, a clean price, etc.'</pre>	
<pre>&lt;quoteUnits&gt; PriceQuoteUnits &lt;/quoteUnits&gt; [0..1] 'The optional units that the measure is expressed in. If not supplied, this is assumed to be a price/value in currency units.'</pre>	
<pre>&lt;side&gt; QuotationSideEnum &lt;/side&gt; [0..1] 'The side (bid/mid/ask) of the measure.'</pre>	
<pre>&lt;currency&gt; Currency &lt;/currency&gt; [0..1] 'The optional currency that the measure is expressed in. If not supplied, this is defaulted from the reportingCurrency in the valuationScenarioDefinition.'</pre>	
<pre>&lt;currencyType&gt; ReportingCurrencyType &lt;/currencyType&gt; [0..1] 'The optional currency that the measure is expressed in. If not supplied, this is defaulted from the reportingCurrency in the valuationScenarioDefinition.'</pre>	
<pre>&lt;timing&gt; QuoteTiming &lt;/timing&gt; [0..1] 'When during a day the quote is for. Typically, if this element is supplied, the QuoteLocation needs also to be supplied.'</pre>	
<p>Start Group: <a href="#">QuoteLocation.model</a> [0..1]</p> <p>'Where the quote is from.'</p>	
<p>Start <a href="#">Choice</a> [1]</p> <ul style="list-style-type: none"> <li>&lt;businessCenter&gt; BusinessCenter &lt;/businessCenter&gt; [1]           <p>'A city or other business center.'</p> </li> <li>&lt;exchangeId&gt; ExchangeId &lt;/exchangeId&gt; [1]           <p>'The exchange (e.g. stock or futures exchange) from which the quote is obtained.'</p> </li> </ul>	
<p>End Choice</p>	
<p>End Group: <a href="#">QuoteLocation.model</a></p>	
<pre>&lt;informationSource&gt; InformationSource &lt;/informationSource&gt; [0..*] 'The information source where a published or displayed market rate will be obtained, e.g. Telerate Page 3750.'</pre>	
<pre>&lt;time&gt; xsd:dateTime &lt;/time&gt; [0..1] 'When the quote was observed or derived.'</pre>	
<pre>&lt;valuationDate&gt; xsd:date &lt;/valuationDate&gt; [0..1] 'When the quote was computed.'</pre>	
<pre>&lt;expiryTime&gt; xsd:dateTime &lt;/expiryTime&gt; [0..1] 'When does the quote cease to be valid.'</pre>	
<pre>&lt;cashflowType&gt; CashflowType &lt;/cashflowType&gt; [0..1] 'For cash flows, the type of the cash flows. Examples include: Coupon payment, Premium Fee, Settlement Fee, Brokerage Fee, etc.'</pre>	

**Diagram**

**Schema Component Representation**

```

<xsd:group name="Quotation.model">
  <xsd:sequence>
    <xsd:element name="value" type="xsd:decimal" minOccurs="0"/>
    <xsd:group ref=" QuotationCharacteristics.model ">
    </xsd:sequence>
</xsd:group>

```

[top](#)**Model Group: QuotationCharacteristics.model**

Name	QuotationCharacteristics.model
Used by (from the same schema document)	Complex Type <a href="#">QuotationCharacteristics</a> , Model Group <a href="#">Quotation.model</a>
Documentation	A group collecting a set of characteristics that can be used to describe a quotation.

**XML Instance Representation**

```

<measureType> AssetMeasureType </measureType> [0..1]
'The type of the value that is measured. This could be an NPV, a cash flow, a clean price, etc.'

<quoteUnits> PriceQuoteUnits </quoteUnits> [0..1]
'The optional units that the measure is expressed in. If not supplied, this is assumed to be a price/value in currency units.'

<side> QuotationSideEnum </side> [0..1]
'The side (bid/mid/ask) of the measure.'

<currency> Currency </currency> [0..1]
'The optional currency that the measure is expressed in. If not supplied, this is defaulted from the reportingCurrency in the valuationScenarioDefinition.'

<currencyType> ReportingCurrencyType </currencyType> [0..1]
'The optional currency that the measure is expressed in. If not supplied, this is defaulted from the reportingCurrency in the valuationScenarioDefinition.'

<timing> QuoteTiming </timing> [0..1]
'When during a day the quote is for. Typically, if this element is supplied, the QuoteLocation needs also to be supplied.'

Start Group: QuoteLocation.model [0..1]
'Where the quote is from.'

Start Choice [1]
<businessCenter> BusinessCenter </businessCenter> [1]
'A city or other business center.'

<exchangeId> ExchangeId </exchangeId> [1]
'The exchange (e.g. stock or futures exchange) from which the quote is obtained.'

```

```

End Choice
End Group: QuoteLocation.model
<informationSource> InformationSource </informationSource> [0..*]
'The information source where a published or displayed market rate will be obtained, e.g. Telerate Page 3750.'

<time> xsd:dateTime </time> [0..1]
'When the quote was observed or derived.'

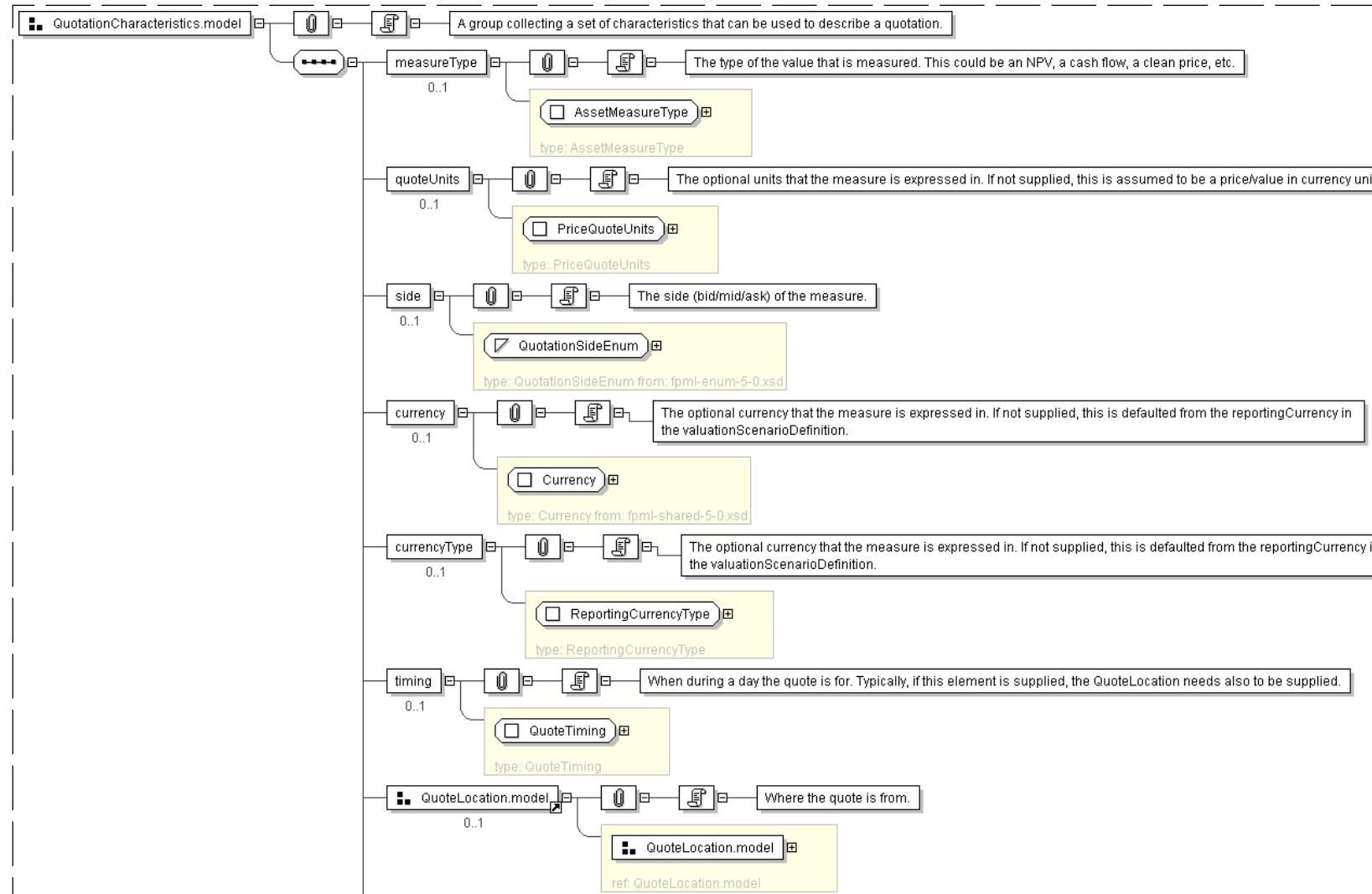
<valuationDate> xsd:date </valuationDate> [0..1]
'When the quote was computed.'

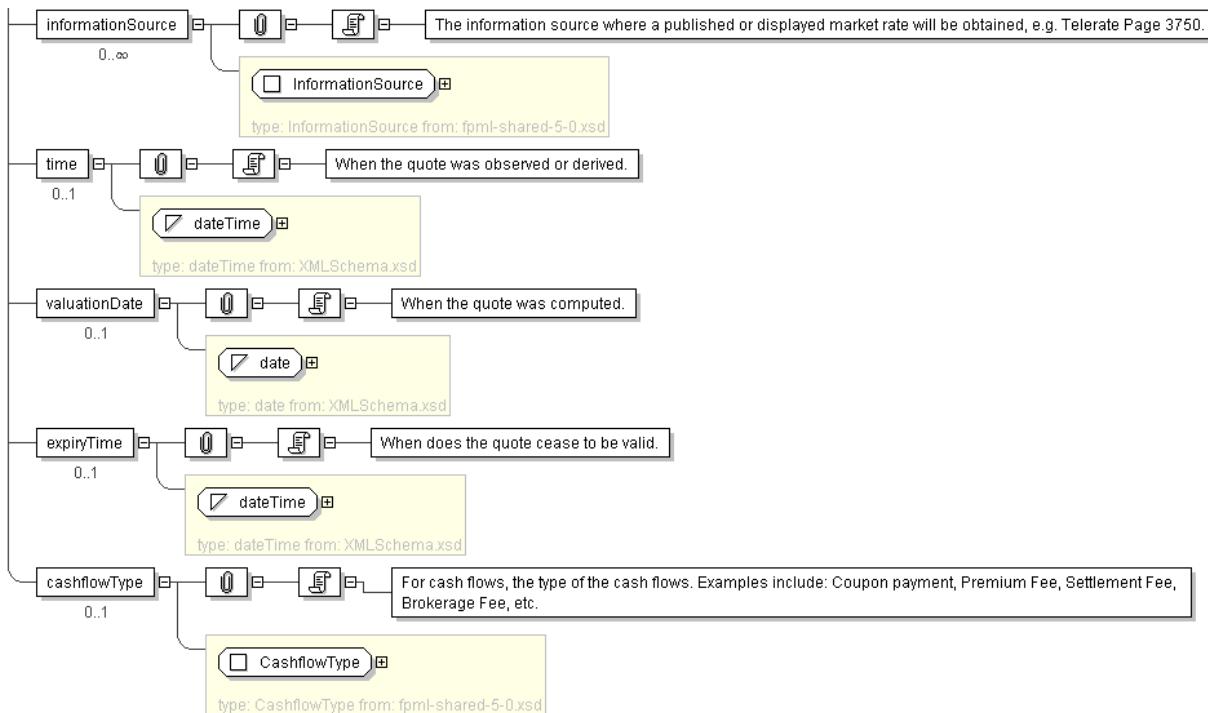
<expiryTime> xsd:dateTime </expiryTime> [0..1]
'When does the quote cease to be valid.'

<cashflowType> CashflowType </cashflowType> [0..1]
'For cash flows, the type of the cash flows. Examples include: Coupon payment, Premium Fee, Settlement Fee, Brokerage Fee, etc.'

```

## Diagram





#### Schema Component Representation

```

<xsd:group name="QuotationCharacteristics.model">
  <xsd:sequence>
    <xsd:element name="measureType" type=" AssetMeasureType " minOccurs="0"/>
    <xsd:element name="quoteUnits" type=" PriceQuoteUnits " minOccurs="0"/>
    <xsd:element name="side" type=" QuotationSideEnum " minOccurs="0"/>
    <xsd:element name="currency" type=" Currency " minOccurs="0"/>
    <xsd:element name="currencyType" type=" ReportingCurrencyType " minOccurs="0"/>
    <xsd:element name="timing" type=" QuoteTiming " minOccurs="0"/>
    <xsd:group ref=" QuoteLocation.model " minOccurs="0"/>
      <xsd:element name="informationSource" type=" InformationSource " minOccurs="0" maxOccurs="unbounded"/>
      <xsd:element name="time" type=" xsd:dateTime " minOccurs="0"/>
      <xsd:element name="valuationDate" type=" xsd:date " minOccurs="0"/>
      <xsd:element name="expiryTime" type=" xsd:dateTime " minOccurs="0"/>
      <xsd:element name="cashflowType" type=" CashflowType " minOccurs="0"/>
    </xsd:group>
  </xsd:sequence>
</xsd:group>

```

top

#### Model Group: `QuoteLocation.model`

Name	QuoteLocation.model
Used by (from the same schema document)	Model Group <code>QuotationCharacteristics.model</code>
Documentation	A group describing where a quote was or will be obtained, e.g. observed or calculated.

#### XML Instance Representation

```

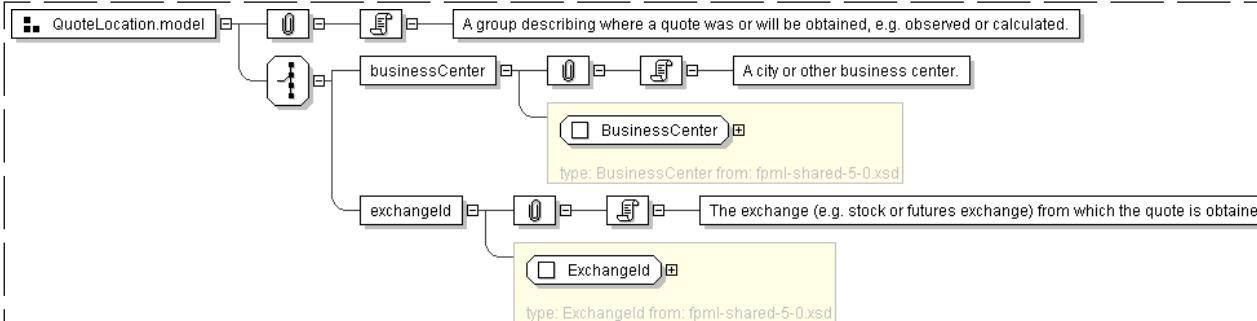
Start Choice [1]
  <businessCenter> BusinessCenter </businessCenter> [1]
  'A city or other business center.'

  <exchangeId> ExchangeId </exchangeId> [1]

```

'The exchange (e.g. stock or futures exchange) from which the quote is obtained.'

End Choice

**Diagram****Schema Component Representation**

```

<xsd:group name="QuoteLocation.model">
  <xsd:choice>
    <xsd:element name="businessCenter" type="BusinessCenter" />
    <xsd:element name="exchangeId" type="ExchangeId" />
  </xsd:choice>
</xsd:group>

```

top

**Legend**

**Complex Type:**  
Schema Component Type

**AusAddress**  
Schema Component Name

**Super-types:** [Address](#) < AusAddress (by extension)  
**Sub-types:** • [QLDAddress](#) (by restriction)

If this schema component is a type definition, its type hierarchy is shown in a gray-bordered box.

Name	AusAddress
Abstract	no

The table above displays the properties of this schema component.

**XML Instance Representation**

```

<... country="Australia" >
<unitNo> string </unitNo> [0..1]
<houseNo> string </houseNo> [1]
<street> string </street> [1]
Start Choice [1]
<city> string </city> [1]
<town> string </town> [1]
End Choice
<state> AusStates </state> [1]
<postcode> string <pattern = "[1-9][0-9]{3}>> </postcode> [1]
</...>

```

The XML Instance Representation table above shows the schema component's content as an XML instance.

- The minimum and maximum occurrence of elements and attributes are provided in square brackets, e.g. [0..1].
- Model group information are shown in gray, e.g. Start Choice ... End Choice.
- For type derivations, the elements and attributes that have been added to or changed from the base type's content are shown in **bold**.
- If an element/attribute has a fixed value, the fixed value is shown in green, e.g. country="Australia".
- Otherwise, the type of the element/attribute is displayed.

- › If the element/attribute's type is in the schema, a link is provided to it.
- › For local simple type definitions, the constraints are displayed in angle brackets, e.g. <>pattern = [1-9][0-9]{3}</>.

## Schema Component Representation

```
<complexType name="AusAddress">
<complexContent>
<extension base=" Address ">
<sequence>
<element name="state" type=" AusStates " />
<element name="postcode">
<simpleType>
<restriction base=" string ">
<pattern value="[1-9][0-9]{3}" />
</restriction>
</simpleType>
</element>
</sequence>
<attribute name="country" type=" string " fixed="Australia"/>
</extension>
</complexContent>
</complexType>
```

The Schema Component Representation table above displays the underlying XML representation of the schema component. (Annotations are not shown.)

[top](#)

## Glossary

**Abstract** (Applies to complex type definitions and element declarations). An abstract element or complex type cannot be used to validate an element instance. If there is a reference to an abstract element, only element declarations that can substitute the abstract element can be used to validate the instance. For references to abstract type definitions, only derived types can be used.

**All Model Group** Child elements can be provided *in any order* in instances. See: <http://www.w3.org/TR/xmlschema-1/#element-all>.

**Choice Model Group** Only one from the list of child elements and model groups can be provided in instances. See: <http://www.w3.org/TR/xmlschema-1/#element-choice>.

**Collapse Whitespace Policy** Replace tab, line feed, and carriage return characters with space character (Unicode character 32). Then, collapse contiguous sequences of space characters into single space character, and remove leading and trailing space characters.

**Disallowed Substitutions** (Applies to element declarations). If *substitution* is specified, then *substitution group* members cannot be used in place of the given element declaration to validate element instances. If *derivation methods*, e.g. extension, restriction, are specified, then the given element declaration will not validate element instances that have types derived from the element declaration's type using the specified derivation methods. Normally, element instances can override their declaration's type by specifying an *xsi:type* attribute.

**Key Constraint** Like [Uniqueness Constraint](#), but additionally requires that the specified value(s) must be provided. See: [http://www.w3.org/TR/xmlschema-1/#clentity-constraint\\_Definitions](http://www.w3.org/TR/xmlschema-1/#clentity-constraint_Definitions).

**Key Reference Constraint** Ensures that the specified value(s) must match value(s) from a [Key Constraint](#) or [Uniqueness Constraint](#). See: [http://www.w3.org/TR/xmlschema-1/#clentity-constraint\\_Definitions](http://www.w3.org/TR/xmlschema-1/#clentity-constraint_Definitions).

**Model Group** Groups together element content, specifying the order in which the element content can occur and the number of times the group of element content may be repeated. See: [http://www.w3.org/TR/xmlschema-1/#Model\\_Groups](http://www.w3.org/TR/xmlschema-1/#Model_Groups).

**Nillable** (Applies to element declarations). If an element declaration is nillable, instances can use the *xsi:nil* attribute. The *xsi:nil* attribute is the boolean attribute, *nil*, from the <http://www.w3.org/2001/XMLSchema-instance> namespace. If an element instance has an *xsi:nil* attribute set to true, it can be left empty, even though its element declaration may have required content.

**Notation** A notation is used to identify the format of a piece of data. Values of elements and attributes that are of type, NOTATION, must come from the names of declared notations. See: [http://www.w3.org/TR/xmlschema-1/#cNotation\\_Declarations](http://www.w3.org/TR/xmlschema-1/#cNotation_Declarations).

**Preserve Whitespace Policy** Preserve whitespaces exactly as they appear in instances.

**Prohibited Derivations** (Applies to type definitions). Derivation methods that cannot be used to create sub-types from a given type definition.

**Prohibited Substitutions** (Applies to complex type definitions). Prevents sub-types that have been derived using the specified derivation methods from validating element instances in place of the given type definition.

**Replace Whitespace Policy** Replace tab, line feed, and carriage return characters with space character (Unicode character 32).

**Sequence Model Group** Child elements and model groups must be provided *in the specified order* in instances. See: <http://www.w3.org/TR/xmlschema-1/#element-sequence>.

**Substitution Group** Elements that are *members* of a substitution group can be used wherever the *head* element of the substitution group is referenced.

**Substitution Group Exclusions** (Applies to element declarations). Prohibits element declarations from nominating themselves as being able to substitute a given element declaration, if they have types that are derived from the original element's type using the specified derivation methods.

**Target Namespace** The target namespace identifies the namespace that components in this schema belongs to. If no target namespace is provided, then the schema components do not belong to any namespace.

**Uniqueness Constraint** Ensures uniqueness of an element/attribute value, or a combination of values, within a specified scope. See: [http://www.w3.org/TR/xmlschema-1/#cIdentity-constraint\\_Definitions](http://www.w3.org/TR/xmlschema-1/#cIdentity-constraint_Definitions).

[top](#)

---

Generated by <oXygen/> XML Editor using a modified version of [xs3p](#) that adds schema diagrams and chunking support.

# XML Schema Documentation

## Table of Contents

- [Schema Document Properties](#)
- [Global Declarations](#)
  - [Element: bondOption](#)
- [Global Definitions](#)
  - [Complex Type: BondOption](#)
  - [Complex Type: BondOptionStrike](#)
  - [Complex Type: MakeWholeAmount](#)
  - [Complex Type: ReferenceSwapCurve](#)
  - [Complex Type: SwapCurveValuation](#)
- [Legend](#)
- [Glossary](#)

[top](#)

## Schema Document Properties

<b>Target Namespace</b>	<a href="http://www.fpml.org/FpML-5/confirmation">http://www.fpml.org/FpML-5/confirmation</a>
<b>Version</b>	\$Revision: 2350 \$
<b>Element and Attribute Namespaces</b>	<ul style="list-style-type: none"> <li>• Global element and attribute declarations belong to this schema's target namespace.</li> <li>• By default, local element declarations belong to this schema's target namespace.</li> <li>• By default, local attribute declarations have no namespace.</li> </ul>
<b>Schema Composition</b>	<ul style="list-style-type: none"> <li>• This schema includes components from the following schema document(s):           <ul style="list-style-type: none"> <li>◦ <a href="#">fpml-option-shared-5-0.xsd</a></li> </ul> </li> </ul>

## Declared Namespaces

Prefix	Namespace
Default namespace	<a href="http://www.fpml.org/FpML-5/confirmation">http://www.fpml.org/FpML-5/confirmation</a>
xml	http://www.w3.org/XML/1998/namespace
xsd	http://www.w3.org/2001/XMLSchema
ecore	http://www.eclipse.org/emf/2002/Ecore

## Schema Component Representation

```

<xsd:schema targetNamespace="http://www.fpml.org/FpML-5/confirmation"
documentRoot="FpML" nsPrefix="conf" package="org.fpml.confirmation" version="$Revision: 2350
$" elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xsd:include schemaLocation="fpml-option-shared-5-0.xsd"/>
  ...
</xsd:schema>

```

[top](#)

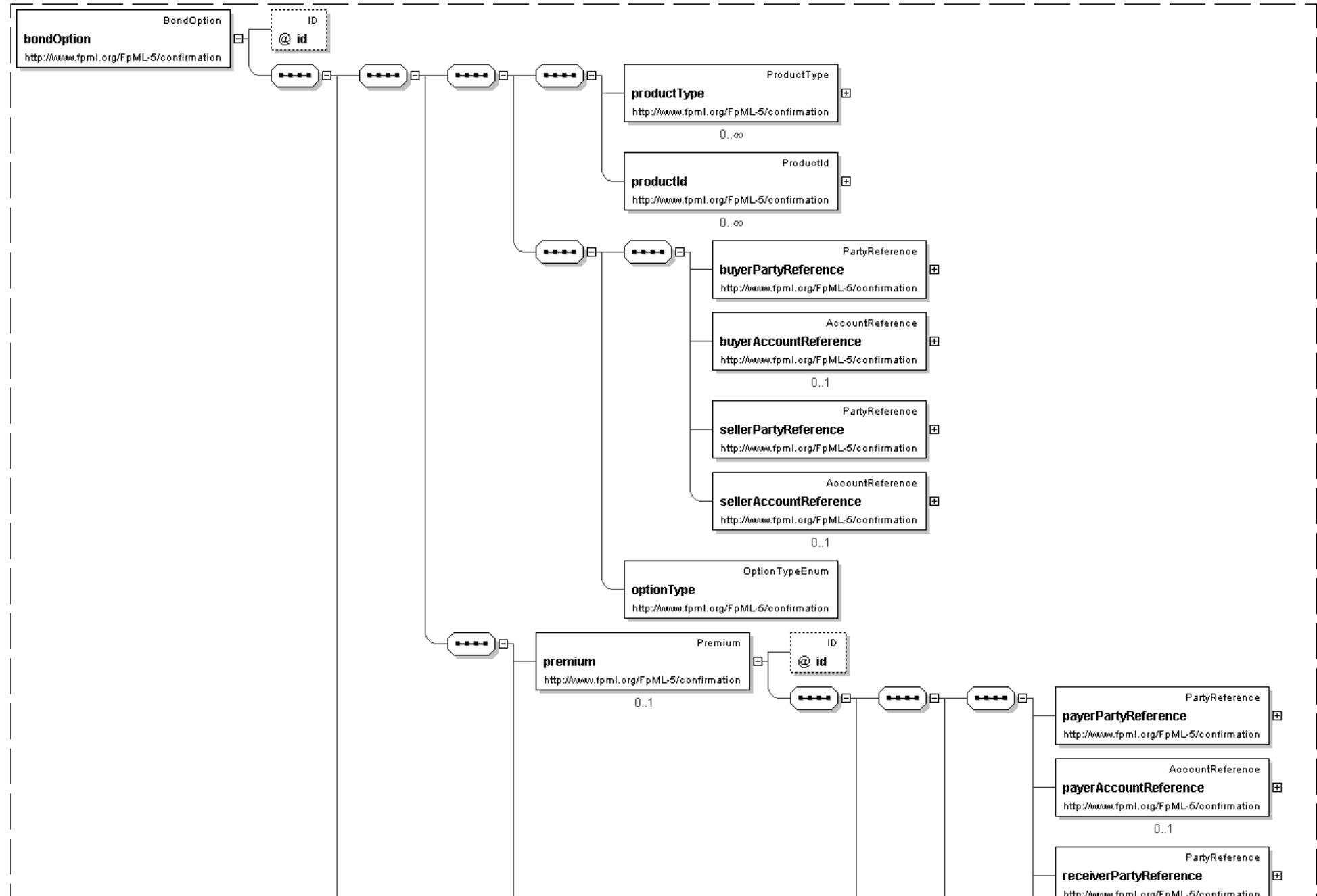
## Global Declarations

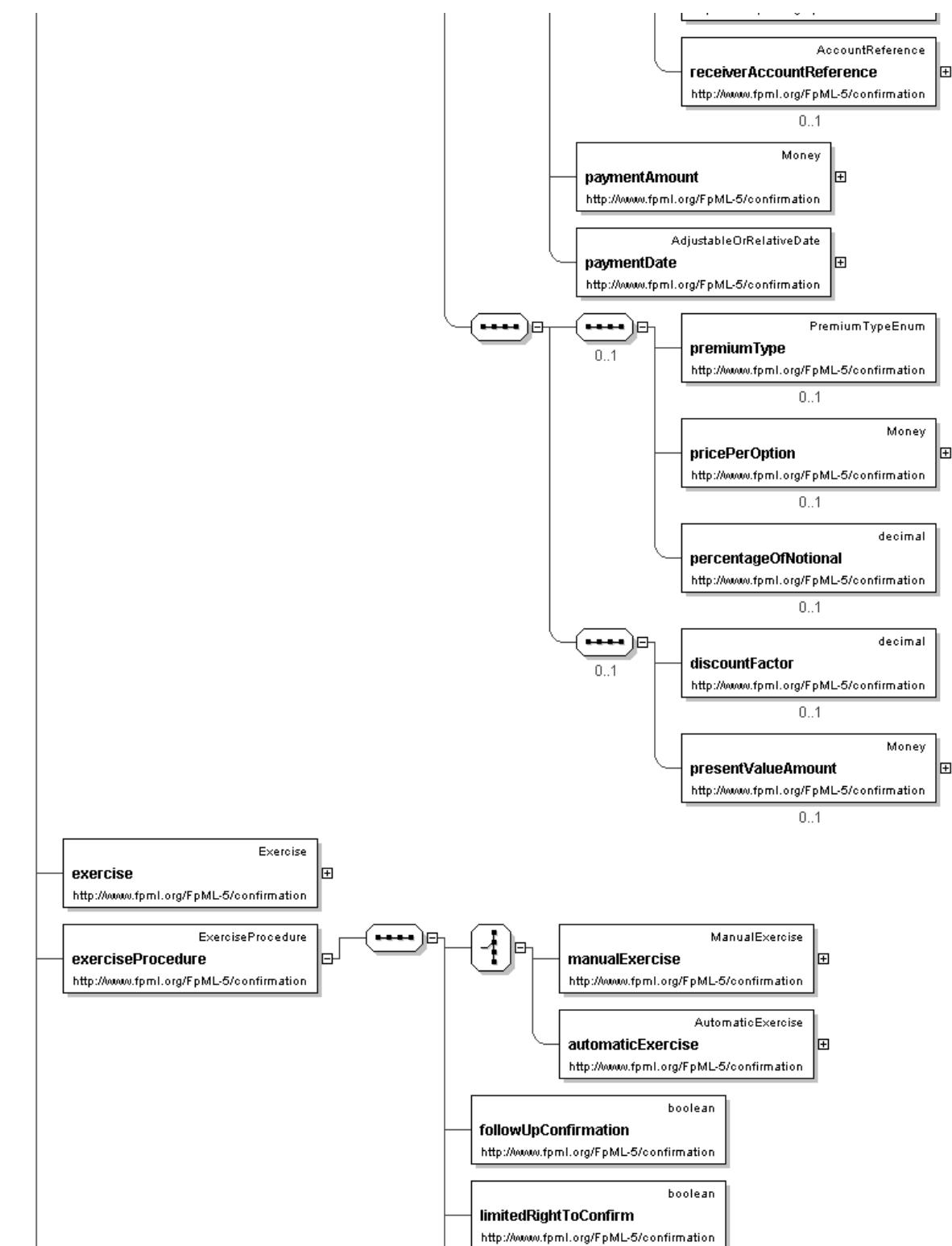
### Element: bondOption

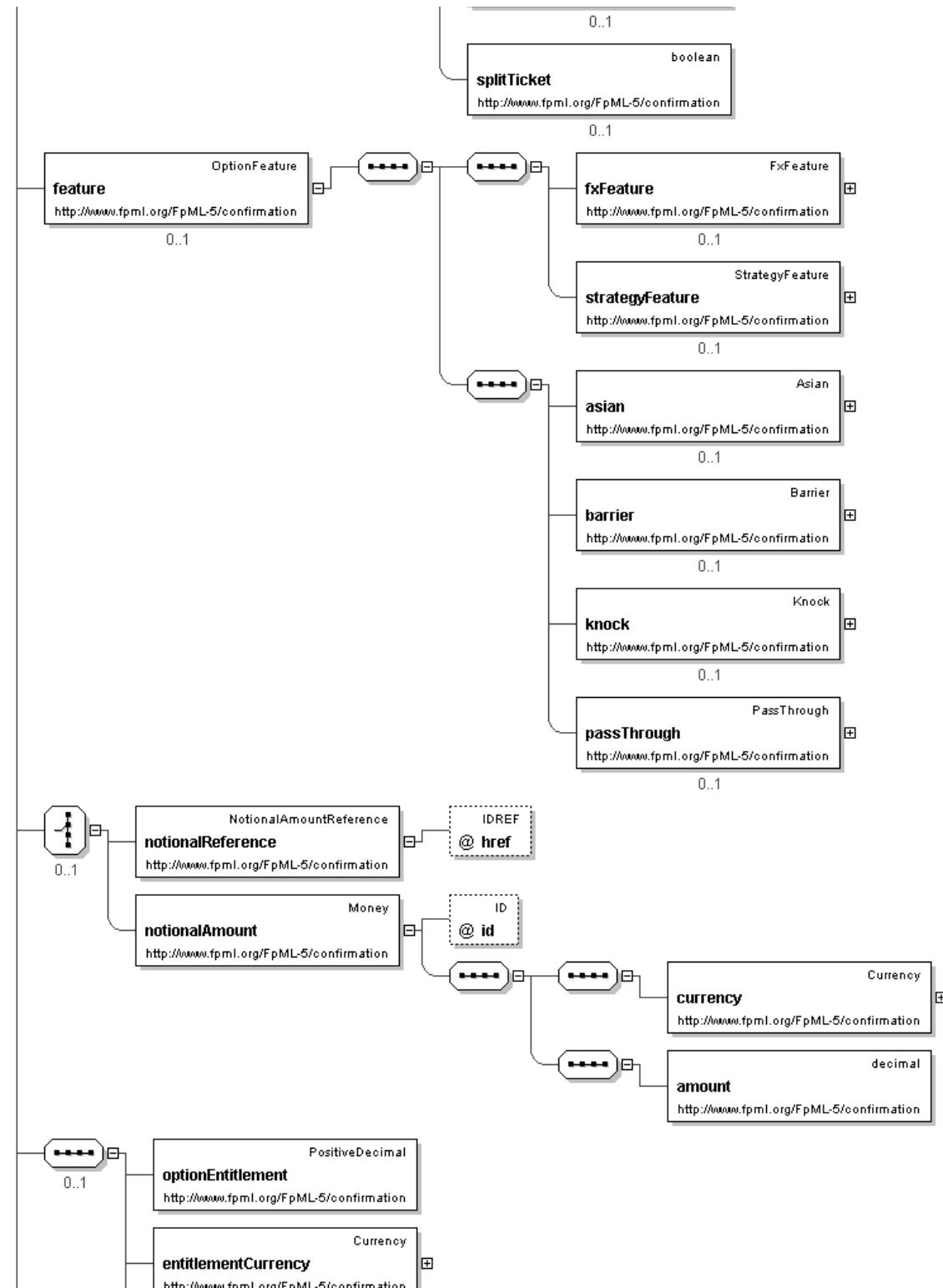
- This element can be used wherever the following element is referenced:
  - [product](#)

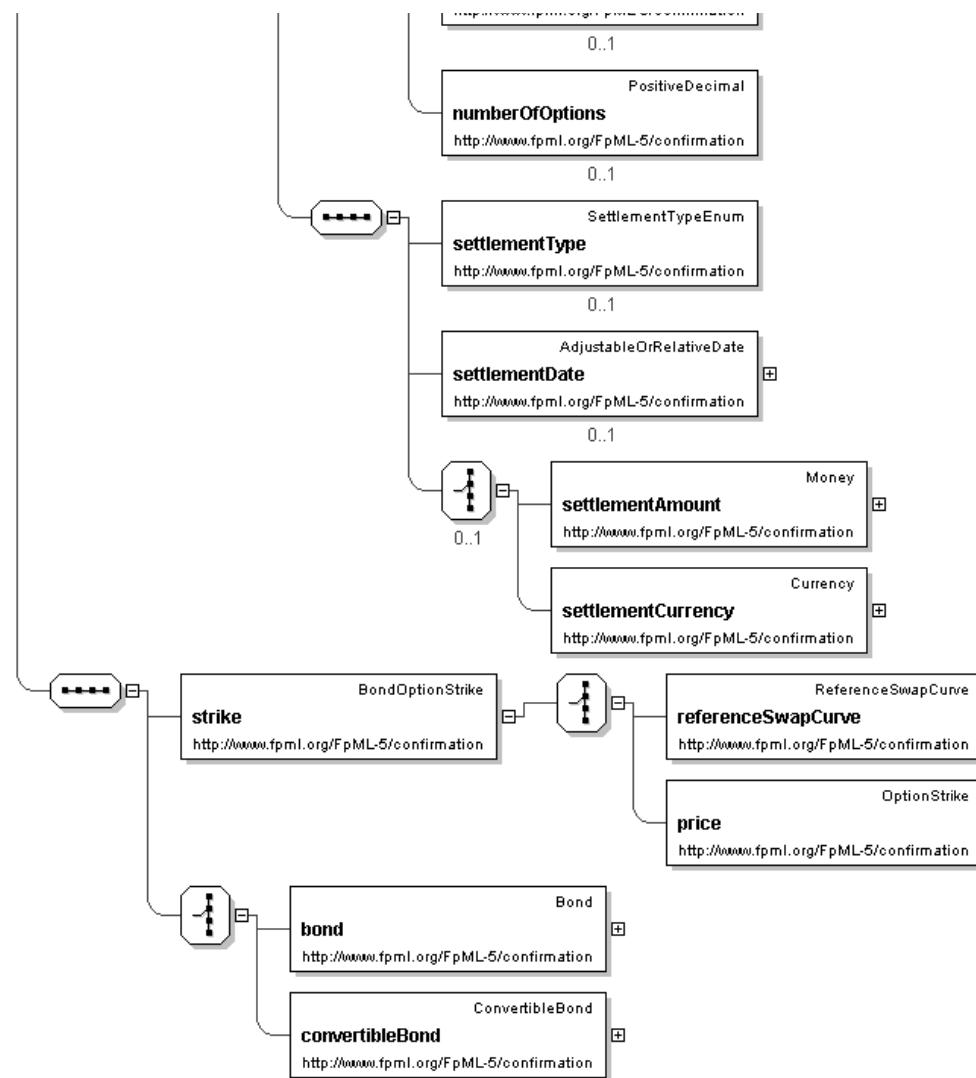
Name	bondOption
Type	BondOption
Nillable	no
Abstract	no
Documentation	A component describing a Bond Option product.

## Logical Diagram







**XML Instance Representation**

```

<bondOption
id=" xsd:ID [0..1]">
<productType> ProductType </productType> [0..*]

```

'A classification of the type of product. FpML defines a simple product categorization using a coding scheme.'

```

<productId> ProductId </productId> [0..*]

```

'A product 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.'

```

<buyerPartyReference> PartyReference </buyerPartyReference> [1]

```

'A reference to the party that buys this instrument, ie. pays for this instrument and receives the rights defined by it. See 2000 ISDA definitions Article 11.1 (b). In the case of FRAs this the fixed rate payer.'

```

<buyerAccountReference> AccountReference </buyerAccountReference> [0..1]
'A reference to the account that buys this instrument.'

<sellerPartyReference> PartyReference </sellerPartyReference> [1]
'A reference to the party that sells ("writes\") this instrument, i.e. that grants the
rights defined by this instrument and in return receives a payment for it. See 2000
ISDA definitions Article 11.1 (a). In the case of FRAs this is the floating rate payer.'

<sellerAccountReference> AccountReference </sellerAccountReference> [0..1]
'A reference to the account that sells this instrument.'

<optionType> OptionTypeEnum </optionType> [1]
'The type of option transaction. From a usage standpoint, put/call is the default option
type, while payer/receiver indicator is used for options index credit default
swaps, consistently with the industry practice. Straddle is used for the case of
straddle strategy, that combine a call and a put with the same strike.'

<premium> Premium </premium> [0..1]
'The option premium payable by the buyer to the seller.'

<exercise> ... </exercise> [1]
<exerciseProcedure> ExerciseProcedure </exerciseProcedure> [1]
'A set of parameters defining procedures associated with the exercise.'

<feature> OptionFeature </feature> [0..1]
'An Option feature such as quanto, asian, barrier, knock.'

Start Choice [0..1]
'A choice between an explicit representation of the notional amount, or a reference to
a notional amount defined elsewhere in this document.'

<notionalReference> NotionalAmountReference </notionalReference> [1]
<notionalAmount> Money </notionalAmount> [1]

End Choice
Start Group: OptionDenomination.model [0..1]
<optionEntitlement> PositiveDecimal </optionEntitlement> [1]
'The number of units of underlyer per option comprised in the option transaction.'

<entitlementCurrency> Currency </entitlementCurrency> [0..1]
'TODO'

<numberOfOptions> PositiveDecimal </numberOfOptions> [0..1]
'The number of options comprised in the option transaction.'

End Group: OptionDenomination.model
<settlementType> SettlementTypeEnum </settlementType> [0..1]
<settlementDate> AdjustableOrRelativeDate </settlementDate> [0..1]
Start Group: SettlementAmountOrCurrency.model [0..1]
Start Choice [1]
<settlementAmount> Money </settlementAmount> [1]
'Settlement Amount'

<settlementCurrency> Currency </settlementCurrency> [1]
'Settlement Currency for use where the Settlement Amount cannot be known in advance'

```

```

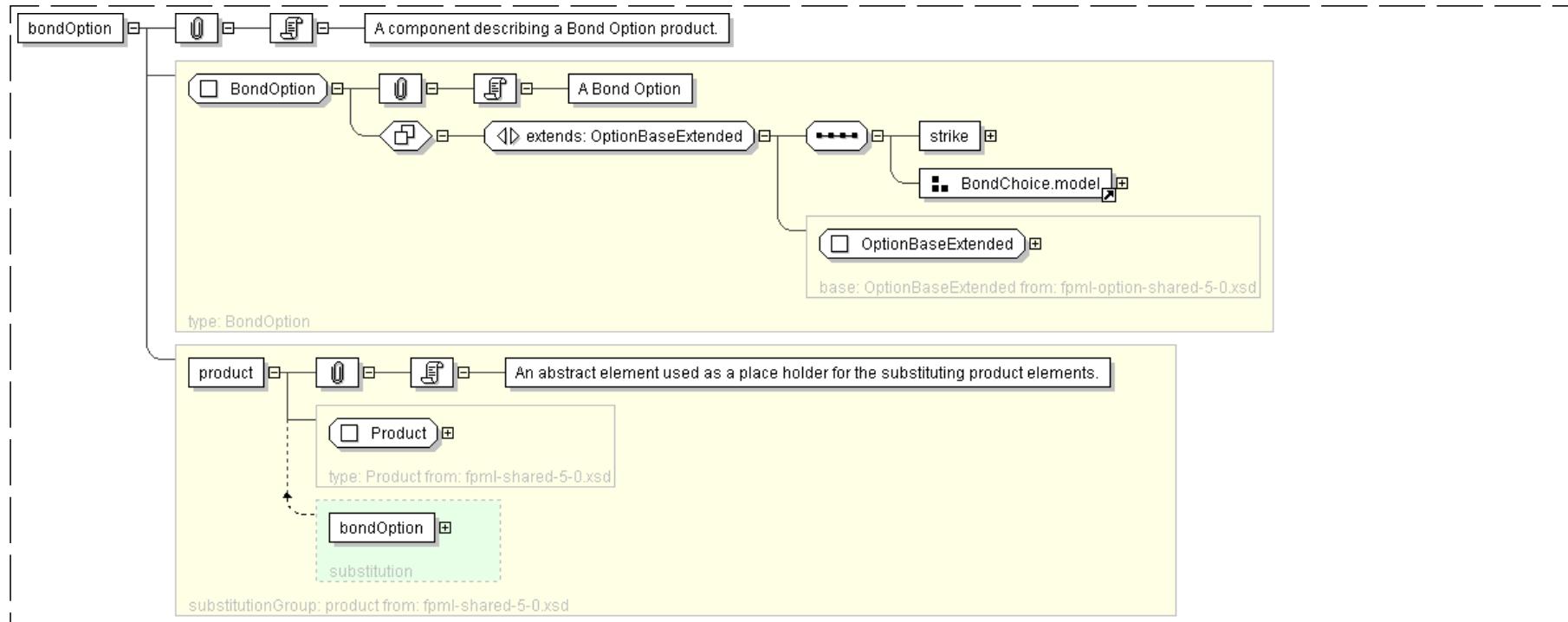
End Choice
End Group: SettlementAmountOrCurrency.model
<strike> BondOptionStrike </strike> [1]
'Strike of the the Bond Option.'

Start Choice [1]
<bond> ... </bond> [1]
'A bond instrument referenced by a contract'

<convertibleBond> ... </convertibleBond> [1]
'A convertible bond instrument referenced by a contract.'

End Choice
</bondOption>

```

**Diagram****Schema Component Representation**

```
<xsd:element name="bondOption" type=" BondOption " substitutionGroup="product"/>
```

top

**Global Definitions****Complex Type: BondOption**

Super-types:

[OptionBaseExtended](#) < **BondOption** (by extension)

## Sub-types:

None

Name	BondOption
Used by (from the same schema document)	Element <a href="#">bondOption</a>
Abstract	no
Documentation	A Bond Option

## XML Instance Representation

```
<...>
<id=" xsd:ID [0..1]">
<productType> ProductType </productType> [0..*]
'A classification of the type of product. FpML defines a simple product categorization using
a coding scheme.'<br/>

<productId> ProductId </productId> [0..*]
'A product 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.'<br/>

<buyerPartyReference> PartyReference </buyerPartyReference> [1]
'A reference to the party that buys this instrument, ie. pays for this instrument and
receives the rights defined by it. See 2000 ISDA definitions Article 11.1 (b). In the case
of FRAs this is the fixed rate payer.'<br/>

<buyerAccountReference> AccountReference </buyerAccountReference> [0..1]
'A reference to the account that buys this instrument.'<br/>

<sellerPartyReference> PartyReference </sellerPartyReference> [1]
'A reference to the party that sells ("writes") this instrument, i.e. that grants the
rights defined by this instrument and in return receives a payment for it. See 2000
ISDA definitions Article 11.1 (a). In the case of FRAs this is the floating rate payer.'<br/>

<sellerAccountReference> AccountReference </sellerAccountReference> [0..1]
'A reference to the account that sells this instrument.'<br/>

<optionType> OptionTypeEnum </optionType> [1]
'The type of option transaction. From a usage standpoint, put/call is the default option
type, while payer/receiver indicator is used for options index credit default
swaps, consistently with the industry practice. Straddle is used for the case of
straddle strategy, that combine a call and a put with the same strike.'<br/>

<premium> Premium </premium> [0..1]
'The option premium payable by the buyer to the seller.'<br/>

<exercise> ... </exercise> [1]
<exerciseProcedure> ExerciseProcedure </exerciseProcedure> [1]
'A set of parameters defining procedures associated with the exercise.'<br/>

<feature> OptionFeature </feature> [0..1]
'An Option feature such as quanto, asian, barrier, knock.'<br/>

Start Choice [0..1]
'A choice between an explicit representation of the notional amount, or a reference to
a notional amount defined elsewhere in this document.'<br/>

<notionalReference> NotionalAmountReference </notionalReference> [1]
```

```

<notionalAmount> Money </notionalAmount> [1]
End Choice
Start Group: OptionDenomination.model [0..1]
<optionEntitlement> PositiveDecimal </optionEntitlement> [1]
'The number of units of underlyer per option comprised in the option transaction.'

<entitlementCurrency> Currency </entitlementCurrency> [0..1]
'TODO'

<numberOfOptions> PositiveDecimal </numberOfOptions> [0..1]
'The number of options comprised in the option transaction.'

End Group: OptionDenomination.model
<settlementType> SettlementTypeEnum </settlementType> [0..1]
<settlementDate> AdjustableOrRelativeDate </settlementDate> [0..1]
Start Group: SettlementAmountOrCurrency.model [0..1]
Start Choice [1]
    <settlementAmount> Money </settlementAmount> [1]
    'Settlement Amount'

    <settlementCurrency> Currency </settlementCurrency> [1]
    'Settlement Currency for use where the Settlement Amount cannot be known in advance'

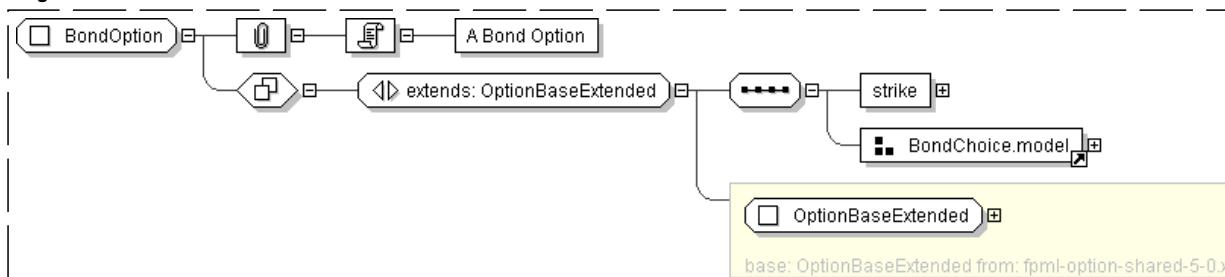
End Choice
End Group: SettlementAmountOrCurrency.model
<strike> BondOptionStrike </strike> [1]
'Strike of the the Bond Option.'

Start Choice [1]
    <bond> ... </bond> [1]
    'A bond instrument referenced by a contract'

    <convertibleBond> ... </convertibleBond> [1]
    'A convertible bond instrument referenced by a contract.'

End Choice
</...>

```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="BondOption">
    <xsd:complexContent>
        <xsd:extension base=" OptionBaseExtended ">
            <xsd:sequence>
                <xsd:element name="strike" type=" BondOptionStrike "/>

```

```

<xsd:group ref=" BondChoice.model " />
</xsd:sequence>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>

```

[top](#)

## Complex Type: BondOptionStrike

Super-types:	None
Sub-types:	None

Name	BondOptionStrike
Used by (from the same schema document)	Complex Type <a href="#">BondOption</a>
Abstract	no
Documentation	A complex type to specify the strike of a bond or convertible bond option.

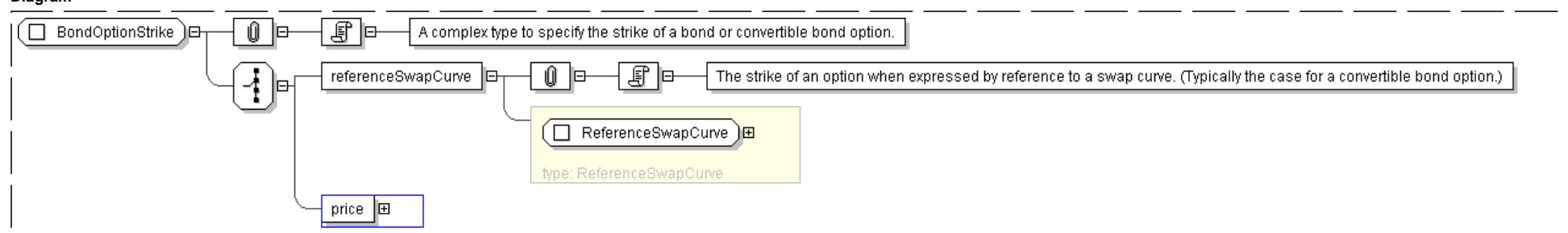
### XML Instance Representation

```

<...>
Start Choice [1]
  <referenceSwapCurve> ReferenceSwapCurve </referenceSwapCurve> [1]
  'The strike of an option when expressed by reference to a swap curve. (Typically the case for
  a convertible bond option.)'
  <price> OptionStrike </price> [1]
End Choice
</...>

```

### Diagram



### Schema Component Representation

```

<xsd:complexType name="BondOptionStrike">
  <xsd:choice>
    <xsd:element name="referenceSwapCurve" type="ReferenceSwapCurve" />
    <xsd:element name="price" type="OptionStrike" />
  </xsd:choice>
</xsd:complexType>

```

[top](#)

## Complex Type: MakeWholeAmount

**Super-types:** [SwapCurveValuation](#) < **MakeWholeAmount** (by extension)

**Sub-types:** None

**Name** MakeWholeAmount

**Used by (from the same schema document)** Complex Type [ReferenceSwapCurve](#)

**Abstract** no

**Documentation** A complex type to specify the amount to be paid by the buyer of the option if the option is exercised prior to the Early Call Date (Typically applicable to the convertible bond options).

#### XML Instance Representation

```
<...>
<floatingRateIndex> FloatingRateIndex </floatingRateIndex> [1]
<indexTenor> Period </indexTenor> [0..1]
'The ISDA Designated Maturity, i.e. the tenor of the floating rate.'

<spread> xsd:decimal </spread> [1]
'Spread in basis points over the floating rate index.'

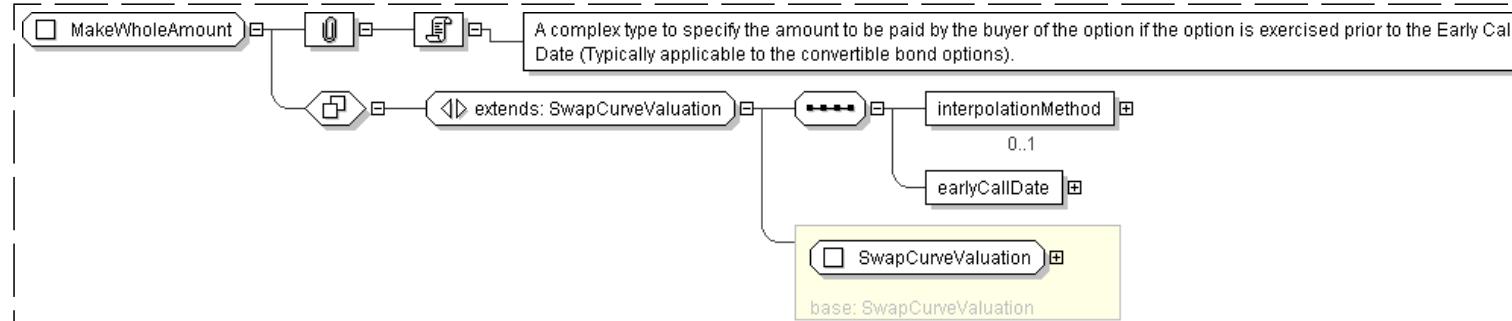
<side> QuotationSideEnum </side> [0..1]
'The side (bid/mid/ask) of the measure.'

<interpolationMethod> InterpolationMethod </interpolationMethod> [0..1]
'The type of interpolation method that the calculation agent reserves the right to use.'

<earlyCallDate> IdentifiedDate </earlyCallDate> [1]
'Date prior to which the option buyer will have to pay a Make Whole Amount to the option seller if he/she exercises the option.'

</...>
```

#### Diagram



#### Schema Component Representation

```

<xsd:complexType name="MakeWholeAmount">
  <xsd:complexContent>
    <xsd:extension base=" SwapCurveValuation ">
      <xsd:sequence>
        <xsd:element name="interpolationMethod" type=" InterpolationMethod" minOccurs="0" />
        <xsd:element name="earlyCallDate" type=" IdentifiedDate" />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
  
```

## Complex Type: ReferenceSwapCurve

<b>Super-types:</b>	None
<b>Sub-types:</b>	None

<b>Name</b>	ReferenceSwapCurve
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">BondOptionStrike</a>
<b>Abstract</b>	no
<b>Documentation</b>	A complex type used to specify the option and convertible bond option strike when expressed in reference to a swap curve.

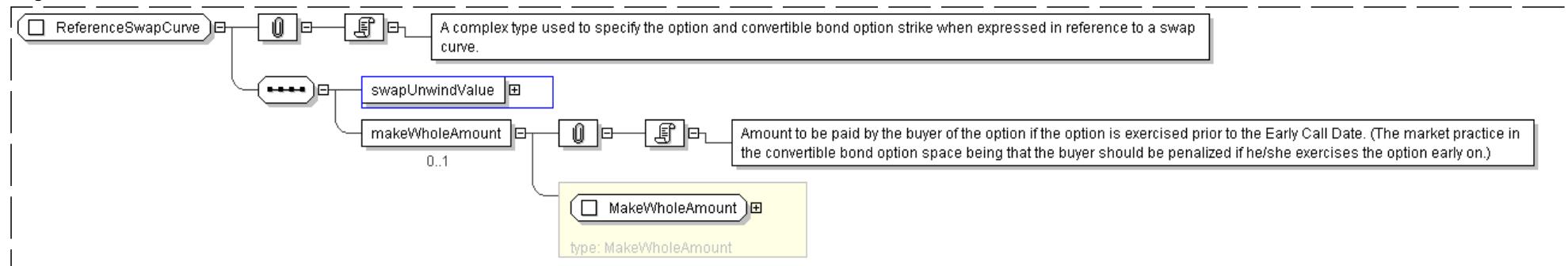
### XML Instance Representation

```
<...>
<swapUnwindValue> SwapCurveValuation </swapUnwindValue> [1]
<makeWholeAmount> MakeWholeAmount </makeWholeAmount> [0..1]

'Amount to be paid by the buyer of the option if the option is exercised prior to the
Early Call Date. (The market practice in the convertible bond option space being that the
buyer should be penalized if he/she exercises the option early on.)'

</...>
```

### Diagram



### Schema Component Representation

```

<xsd:complexType name="ReferenceSwapCurve">
  <xsd:sequence>
    <xsd:element name="swapUnwindValue" type=" SwapCurveValuation " />
    <xsd:element name="makeWholeAmount" type=" MakeWholeAmount " minOccurs="0" />
  </xsd:sequence>
</xsd:complexType>

```

## Complex Type: SwapCurveValuation

<b>Super-types:</b>	None
<b>Sub-types:</b>	• <a href="#">MakeWholeAmount</a> (by extension)

<b>Name</b>	SwapCurveValuation
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">ReferenceSwapCurve</a>

**Abstract**

no

**Documentation**

A complex type to specify a valuation swap curve, which is used as part of the strike construct for the bond and convertible bond options.

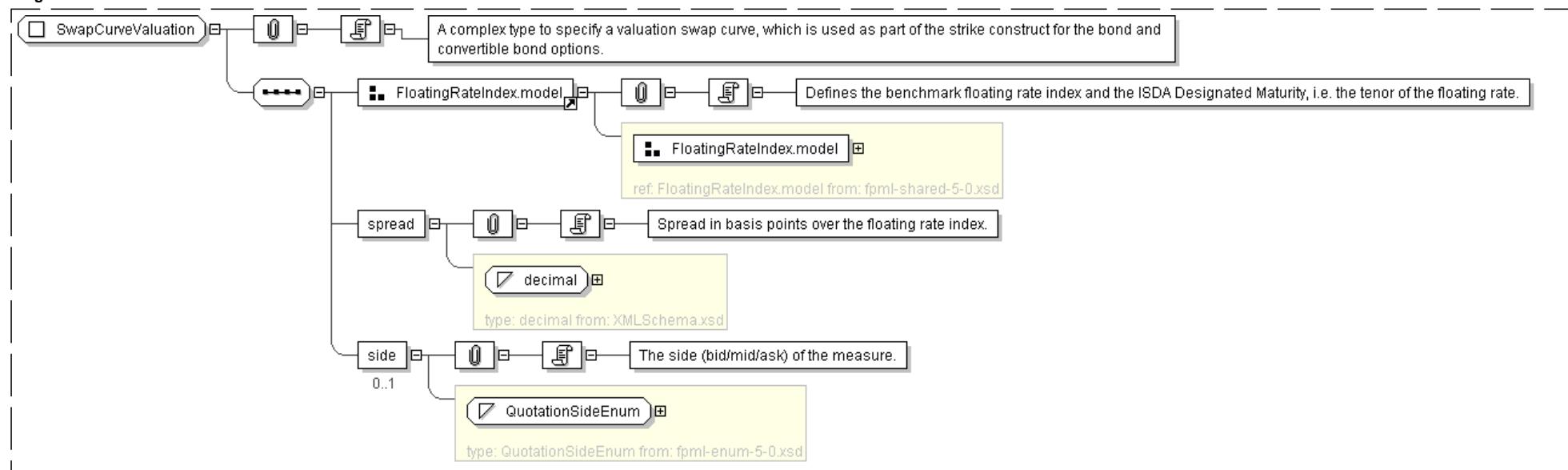
**XML Instance Representation**

```

<...>
  <floatingRateIndex> FloatingRateIndex </floatingRateIndex> [1]
  <indexTenor> Period </indexTenor> [0..1]
    'The ISDA Designated Maturity, i.e. the tenor of the floating rate.'
  <spread> xsd:decimal </spread> [1]
    'Spread in basis points over the floating rate index.'
  <side> QuotationSideEnum </side> [0..1]
    'The side (bid/mid/ask) of the measure.'

</...>

```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="SwapCurveValuation">
  <xsd:sequence>
    <xsd:group ref=" FloatingRateIndex.model " />
    <xsd:element name="spread" type=" xsd:decimal " />
    <xsd:element name="side" type=" QuotationSideEnum " minOccurs="0" />
  </xsd:sequence>
</xsd:complexType>

```

top

**Legend****Complex Type:****AusAddress**

**Super-types:**[Address](#) < AusAddress (by extension)**Sub-types:**

- [QLDAddress](#) (by restriction)

If this schema component is a type definition, its type hierarchy is shown in a gray-bordered box.

<b>Name</b>	AusAddress
<b>Abstract</b>	no

The table above displays the properties of this schema component.

**XML Instance Representation**

```
<.... country="Australia" >
<unitNo> string </unitNo> [0..1]
<houseNo> string </houseNo> [1]
<street> string </street> [1]
Start Choice [1]
<city> string </city> [1]
<town> string </town> [1]
End Choice
<state> AusStates </state> [1]
<postcode> string <>pattern = [1-9][0-9]{3}</> </postcode> [1]
</....>
```

The XML Instance Representation table above shows the schema component's content as an XML instance.

- The minimum and maximum occurrence of elements and attributes are provided in square brackets, e.g. [0..1].
- Model group information are shown in gray, e.g. Start Choice ... End Choice.
- For type derivations, the elements and attributes that have been added to or changed from the base type's content are shown in **bold**.
- If an element/attribute has a fixed value, the fixed value is shown in green, e.g. country="Australia".
- Otherwise, the type of the element/attribute is displayed.
  - If the element/attribute's type is in the schema, a link is provided to it.
  - For local simple type definitions, the constraints are displayed in angle brackets, e.g. <>pattern = [1-9][0-9]{3}</>.

**Schema Component Representation**

```
<complexType name="AusAddress">
<complexContent>
<extension base=" Address ">
<sequence>
<element name="state" type=" AusStates " />
<element name="postcode">
<simpleType>
<restriction base=" string ">
<pattern value="[1-9][0-9]{3}" />
</restriction>
</simpleType>
</element>
</sequence>
<attribute name="country" type=" string " fixed="Australia" />
</extension>
</complexContent>
</complexType>
```

The Schema Component Representation table above displays the underlying XML representation of the schema component. (Annotations are not shown.)

## Glossary

**Abstract** (Applies to complex type definitions and element declarations). An abstract element or complex type cannot be used to validate an element instance. If there is a reference to an abstract element, only element declarations that can substitute the abstract element can be used to validate the instance. For references to abstract type definitions, only derived types can be used.

**All Model Group** Child elements can be provided *in any order* in instances. See: <http://www.w3.org/TR/xmlschema-1/#element-all>.

**Choice Model Group** Only one from the list of child elements and model groups can be provided in instances. See: <http://www.w3.org/TR/xmlschema-1/#element-choice>.

**Collapse Whitespace Policy** Replace tab, line feed, and carriage return characters with space character (Unicode character 32). Then, collapse contiguous sequences of space characters into single space character, and remove leading and trailing space characters.

**Disallowed Substitutions** (Applies to element declarations). If *substitution* is specified, then *substitution group* members cannot be used in place of the given element declaration to validate element instances. If *derivation methods*, e.g. extension, restriction, are specified, then the given element declaration will not validate element instances that have types derived from the element declaration's type using the specified derivation methods. Normally, element instances can override their declaration's type by specifying an *xsi:type* attribute.

**Key Constraint** Like *Uniqueness Constraint*, but additionally requires that the specified value(s) must be provided. See: [http://www.w3.org/TR/xmlschema-1/#cIdentity-constraint\\_Definitions](http://www.w3.org/TR/xmlschema-1/#cIdentity-constraint_Definitions).

**Key Reference Constraint** Ensures that the specified value(s) must match value(s) from a *Key Constraint* or *Uniqueness Constraint*. See: [http://www.w3.org/TR/xmlschema-1/#cIdentity-constraint\\_Definitions](http://www.w3.org/TR/xmlschema-1/#cIdentity-constraint_Definitions).

**Model Group** Groups together element content, specifying the order in which the element content can occur and the number of times the group of element content may be repeated. See: [http://www.w3.org/TR/xmlschema-1/#Model\\_Groups](http://www.w3.org/TR/xmlschema-1/#Model_Groups).

**Nillable** (Applies to element declarations). If an element declaration is nillable, instances can use the *xsi:nil* attribute. The *xsi:nil* attribute is the boolean attribute, *nil*, from the <http://www.w3.org/2001/XMLSchema-instance> namespace. If an element instance has an *xsi:nil* attribute set to true, it can be left empty, even though its element declaration may have required content.

**Notation** A notation is used to identify the format of a piece of data. Values of elements and attributes that are of type, NOTATION, must come from the names of declared notations. See: [http://www.w3.org/TR/xmlschema-1/#cNotation\\_Declarations](http://www.w3.org/TR/xmlschema-1/#cNotation_Declarations).

**Preserve Whitespace Policy** Preserve whitespaces exactly as they appear in instances.

**Prohibited Derivations** (Applies to type definitions). Derivation methods that cannot be used to create sub-types from a given type definition.

**Prohibited Substitutions** (Applies to complex type definitions). Prevents sub-types that have been derived using the specified derivation methods from validating element instances in place of the given type definition.

**Replace Whitespace Policy** Replace tab, line feed, and carriage return characters with space character (Unicode character 32).

**Sequence Model Group** Child elements and model groups must be provided *in the specified order* in instances. See: <http://www.w3.org/TR/xmlschema-1/#element-sequence>.

**Substitution Group** Elements that are *members* of a substitution group can be used wherever the *head* element of the substitution group is referenced.

**Substitution Group Exclusions** (Applies to element declarations). Prohibits element declarations from nominating themselves as being able to substitute a given element declaration, if they have types that are derived from the original element's type using the specified derivation methods.

**Target Namespace** The target namespace identifies the namespace that components in this schema belongs to. If no target namespace is provided, then the schema components do not belong to any namespace.

**Uniqueness Constraint** Ensures uniqueness of an element/attribute value, or a combination of values, within a specified scope. See: [http://www.w3.org/TR/xmlschema-1/#cIdentity-constraint\\_Definitions](http://www.w3.org/TR/xmlschema-1/#cIdentity-constraint_Definitions).

---

Generated by [oXygen XML Editor](#) using a modified version of [xs3p](#) that adds schema diagrams and chunking support.

# XML Schema Documentation

## Table of Contents

- [Schema Document Properties](#)
- [Global Declarations](#)
  - Element: [additionalEvent](#)
  - Element: [changeEvent](#)
  - Element: [indexChange](#)
- [Global Definitions](#)
  - Complex Type: [ChangeEvent](#)
  - Complex Type: [EventProposedMatch](#)
  - Complex Type: [EventsChoice](#)
  - Complex Type: [IndexChange](#)
  - Complex Type: [TradeAmendmentContent](#)
  - Complex Type: [TradeChangeBase](#)
  - Complex Type: [TradeChangeContent](#)
  - Complex Type: [TradeNotionalChange](#)
  - Complex Type: [TradeNovationContent](#)
  - Model Group: [AgreementAndEffectiveDates.model](#)
  - Model Group: [Events.model](#)
  - Model Group: [ProposedMatch.model](#)
  - Model Group: [TradeAlterationPayment.model](#)
- [Legend](#)
- [Glossary](#)

[top](#)

## Schema Document Properties

<b>Target Namespace</b>	<a href="http://www.fpml.org/FpML-5/confirmation">http://www.fpml.org/FpML-5/confirmation</a>
<b>Version</b>	\$Revision: 2864 \$
<b>Element and Attribute Namespaces</b>	<ul style="list-style-type: none"> <li>• Global element and attribute declarations belong to this schema's target namespace.</li> <li>• By default, local element declarations belong to this schema's target namespace.</li> <li>• By default, local attribute declarations have no namespace.</li> </ul>
<b>Schema Composition</b>	<ul style="list-style-type: none"> <li>• This schema includes components from the following schema document(s):           <ul style="list-style-type: none"> <li>◦ <a href="#">fpml-msg-5-0.xsd</a></li> </ul> </li> </ul>

## Declared Namespaces

Prefix	Namespace
Default namespace	<a href="http://www.fpml.org/FpML-5/confirmation">http://www.fpml.org/FpML-5/confirmation</a>
xml	http://www.w3.org/1998/namespace
dsig	http://www.w3.org/2000/09/xmldsig#
fpml-annotation	http://www.fpml.org/annotation
xsd	http://www.w3.org/2001/XMLSchema
ecore	http://www.eclipse.org/emf/2002/Ecore

## Schema Component Representation

```

<xsd:schema targetNamespace="http://www.fpml.org/FpML-5/confirmation"
  documentRoot="FpML" nsPrefix="conf" package="org.fpml.confirmation" version="$Revision: 2864 "
  $" elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xsd:include schemaLocation="fpml-msg-5-0.xsd"/>
  ...
</xsd:schema>

```

## Global Declarations

### Element: additionalEvent

Name	additionalEvent
Used by (from the same schema document)	Model Group <a href="#">Events.model</a>
Type	anyType
Nillable	no
Abstract	yes

#### Logical Diagram



#### XML Instance Representation

```
<additionalEvent> ... </additionalEvent>
```

#### Diagram



#### Schema Component Representation

```
<xsd:element name="additionalEvent" abstract="true" />
```

### Element: changeEvent

- The following elements can be used wherever this element is referenced:
  - [indexChange](#)

Name	changeEvent
Used by (from the same schema document)	Complex Type <a href="#">TradeChangeContent</a>
Type	<a href="#">ChangeEvent</a>
Nillable	no
Abstract	yes
Documentation	Abstract substitutable place holder for specific change details.

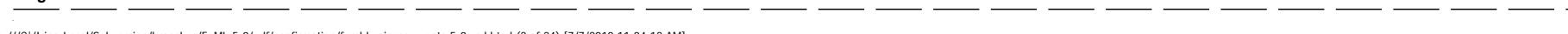
#### Logical Diagram

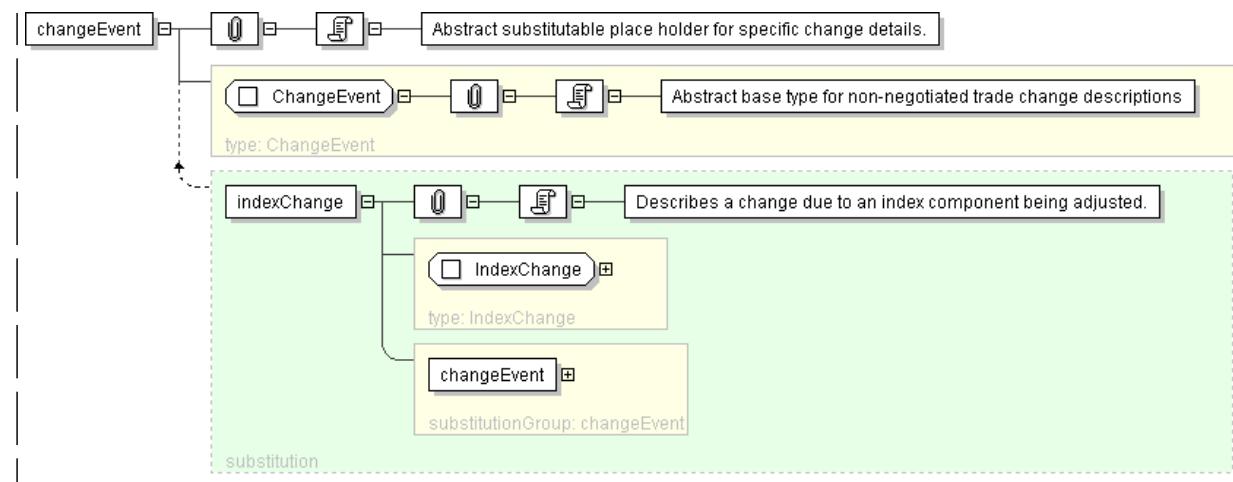


#### XML Instance Representation

```
<changeEvent />
```

#### Diagram



**Schema Component Representation**

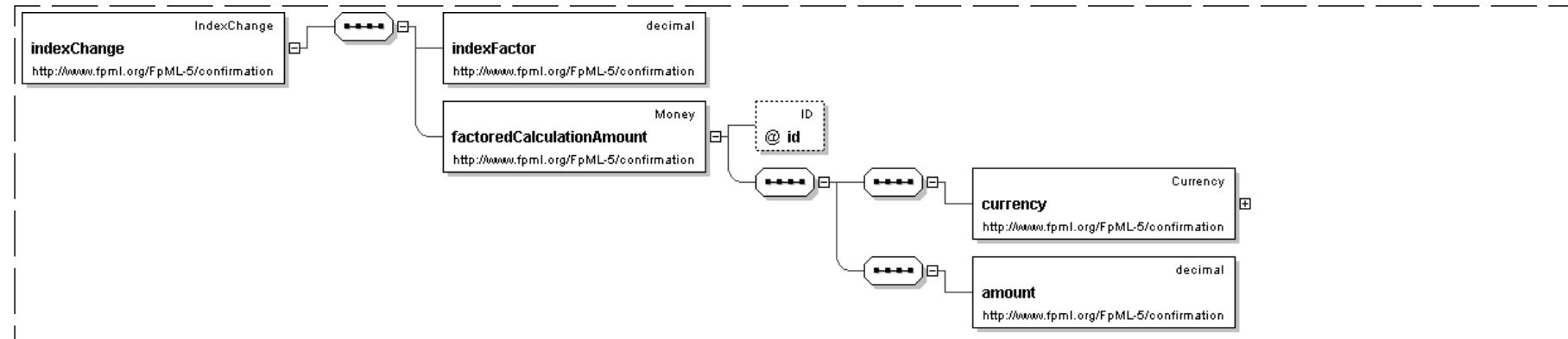
```
<xsd:element name="changeEvent" type=" ChangeEvent " abstract="true" />
```

top

**Element: indexChange**

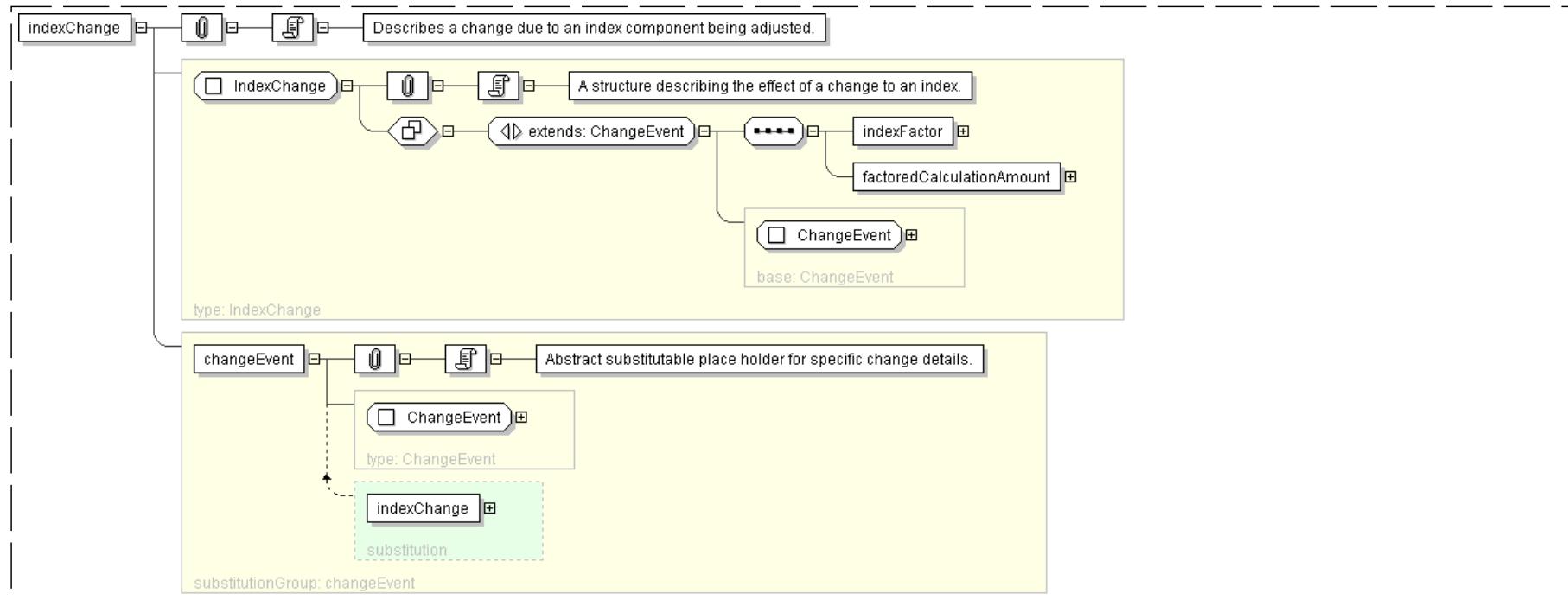
- This element can be used wherever the following element is referenced:
  - [changeEvent](#)

Name	indexChange
Type	<a href="#">IndexChange</a>
Nillable	no
Abstract	no
Documentation	Describes a change due to an index component being adjusted.

**Logical Diagram****XML Instance Representation**

```
<indexChange>
  <indexFactor> xsd:decimal </indexFactor> [1]
  <factoredCalculationAmount> Money </factoredCalculationAmount> [1]
```

&lt;/indexChange&gt;

**Diagram****Schema Component Representation**

```
<xsd:element name="indexChange" type=" IndexChange " substitutionGroup="changeEvent" />
```

top

**Global Definitions****Complex Type: ChangeEvent**

<b>Super-types:</b>	None
<b>Sub-types:</b>	<ul style="list-style-type: none"> <li>• <a href="#">IndexChange</a> (by extension)</li> </ul>

<b>Name</b>	ChangeEvent
<b>Used by (from the same schema document)</b>	Element <a href="#">changeEvent</a>
<b>Abstract</b>	yes
<b>Documentation</b>	Abstract base type for non-negotiated trade change descriptions

**XML Instance Representation**

&lt; . . . /&gt;

**Diagram****Schema Component Representation**

```
<xsd:complexType name="ChangeEvent" abstract="true"/>
```

[top](#)

## Complex Type: EventProposedMatch

Super-types:	None
Sub-types:	None

Name	EventProposedMatch
Abstract	no

### XML Instance Representation

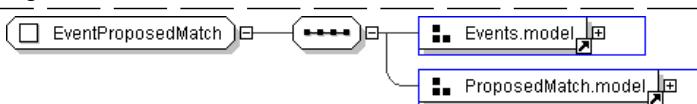
```
<...>
Start Choice [1]
  <trade> Trade </trade> [1]
  <amendment> TradeAmendmentContent </amendment> [1]
  <increase> TradeNotionalChange </increase> [1]
  <termination> TradeNotionalChange </termination> [1]
  <novation> TradeNovationContent </novation> [1]
  <additionalEvent> ... </additionalEvent> [1]
End Choice
<matchId> MatchId </matchId> [0..1]
'A unique identifier assigned by the matching service to each set of matched positions.'

<difference> Difference </difference> [0..*]
'A type used to record the details of a difference between two sides of a business event.'

<matchScore> xsd:decimal </matchScore> [0..1]
'Numeric score to represent the quality of the match.'

</...>
```

### Diagram



### Schema Component Representation

```
<xsd:complexType name="EventProposedMatch">
  <xsd:sequence>
    <xsd:group ref=" Events.model " />
    <xsd:group ref=" ProposedMatch.model " />
  </xsd:sequence>
</xsd:complexType>
```

[top](#)

## Complex Type: EventsChoice

Super-types:	None
Sub-types:	None

Name	EventsChoice
------	--------------

**Abstract**

no

**XML Instance Representation**

```
<...>
Start Choice [1]
  <trade> Trade </trade> [1]
  <amendment> TradeAmendmentContent </amendment> [1]
  <increase> TradeNotionalChange </increase> [1]
  <termination> TradeNotionalChange </termination> [1]
  <novation> TradeNovationContent </novation> [1]
  <additionalEvent> ... </additionalEvent> [1]
End Choice
</...>
```

**Diagram****Schema Component Representation**

```
<xsd:complexType name="EventsChoice">
  <xsd:sequence>
    <xsd:group ref=" Events.model " />
  </xsd:sequence>
</xsd:complexType>
```

[top](#)**Complex Type: IndexChange**

**Super-types:** [ChangeEvent](#) < **IndexChange** (by extension)

**Sub-types:** None

**Name** IndexChange

**Used by (from the same schema document)** Element [indexChange](#)

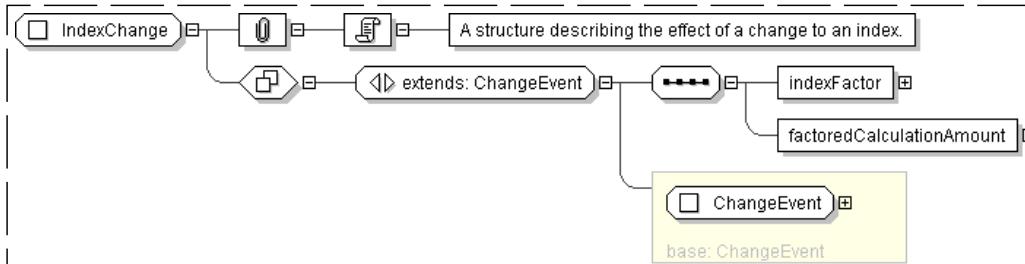
**Abstract**

no

**Documentation** A structure describing the effect of a change to an index.

**XML Instance Representation**

```
<...>
  <indexFactor> xsd:decimal </indexFactor> [1]
  <factoredCalculationAmount> Money </factoredCalculationAmount> [1]
</...>
```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="IndexChange">
  <xsd:complexContent>
    <xsd:extension base=" ChangeEvent ">
      <xsd:sequence>
        <xsd:element name="indexFactor" type=" xsd:decimal "/>
        <xsd:element name="factoredCalculationAmount" type=" Money "/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

[top](#)

## Complex Type: TradeAmendmentContent

Super-types:	None
Sub-types:	None

Name	TradeAmendmentContent
Used by (from the same schema document)	Model Group <a href="#">Events.model</a>
Abstract	no
Documentation	A structure describing a negotiated amendment.

### XML Instance Representation

```

<...>
  <trade> Trade </trade> [1]
  'A full description of the amended trade.'

  <agreementDate> xsd:date </agreementDate> [1]
  'The date on which the change was agreed.'

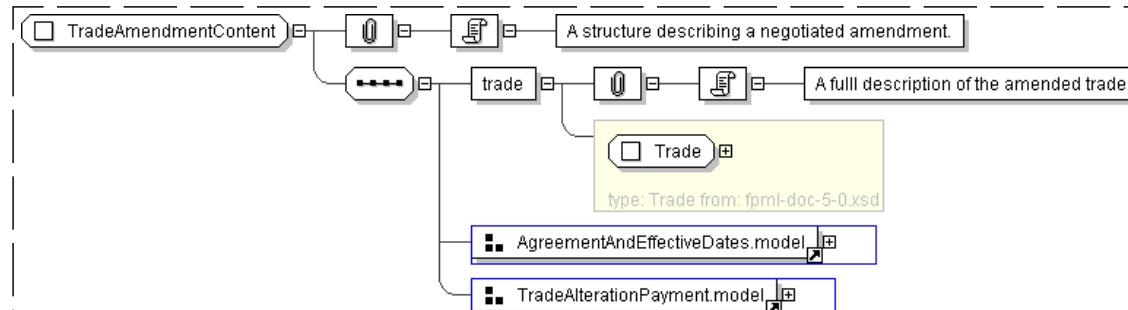
  <effectiveDate> xsd:date </effectiveDate> [1]
  'The date on which the change become effective.'

  <payment> Payment </payment> [0..1]
  'Describes a payment made in settlement of the change.'

</...>

```

### Diagram



### Schema Component Representation

```

<xsd:complexType name="TradeAmendmentContent">
  <xsd:sequence>
    <xsd:element name="trade" type=" Trade "/>

```

```

<xsd:group ref=" AgreementAndEffectiveDates.model " />
<xsd:group ref=" TradeAlterationPayment.model " />
</xsd:sequence>
</xsd:complexType>

```

[top](#)

## Complex Type: TradeChangeBase

Super-types:

None

Sub-types:

• [TradeNotionalChange](#) (by extension)**Name**

TradeChangeBase

**Abstract**

no

**Documentation**

A structure describing a trade change.

### XML Instance Representation

```

<...>
<tradeIdentifier> PartyTradeIdentifier </tradeIdentifier> [1..*]
<agreementDate> xsd:date </agreementDate> [1]
'The date on which the change was agreed.'

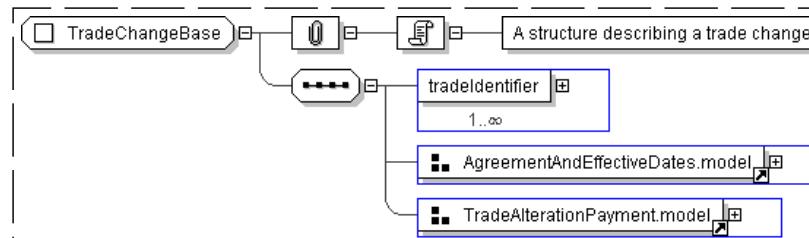
<effectiveDate> xsd:date </effectiveDate> [1]
'The date on which the change became effective.'

<payment> Payment </payment> [0..1]
'Describes a payment made in settlement of the change.'

</...>

```

### Diagram



### Schema Component Representation

```

<xsd:complexType name="TradeChangeBase">
  <xsd:sequence>
    <xsd:element name="tradeIdentifier" type=" PartyTradeIdentifier " maxOccurs="unbounded" />
    <xsd:group ref=" AgreementAndEffectiveDates.model " />
    <xsd:group ref=" TradeAlterationPayment.model " />
  </xsd:sequence>
</xsd:complexType>

```

[top](#)

## Complex Type: TradeChangeContent

Super-types:

None

Sub-types:

None

**Name** TradeChangeContent**Abstract** no**Documentation** A structure describing a non-negotiated trade resulting from a market event.**XML Instance Representation**

```
<...>
Start Choice [0..1]
<oldTradeIdentifier> PartyTradeIdentifier </oldTradeIdentifier> [1]
'The original qualified trade identifier.'

<oldTrade> Trade </oldTrade> [1]
'The original trade details.'

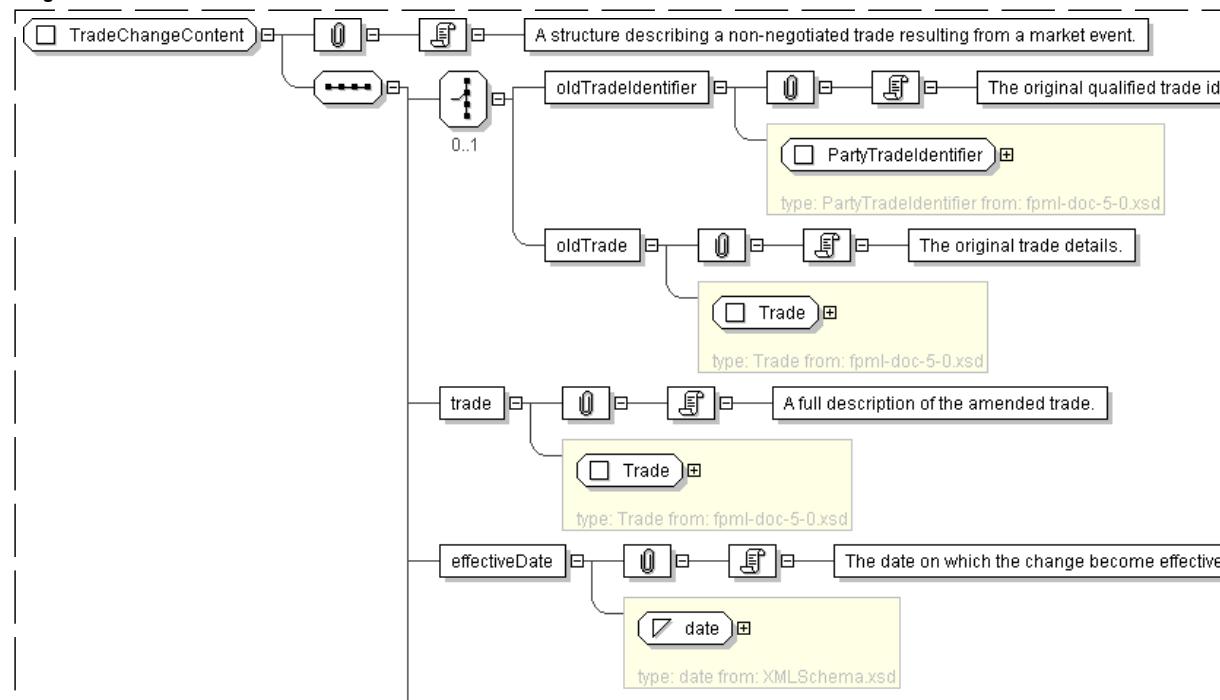
End Choice
<trade> Trade </trade> [1]
'A full description of the amended trade.'

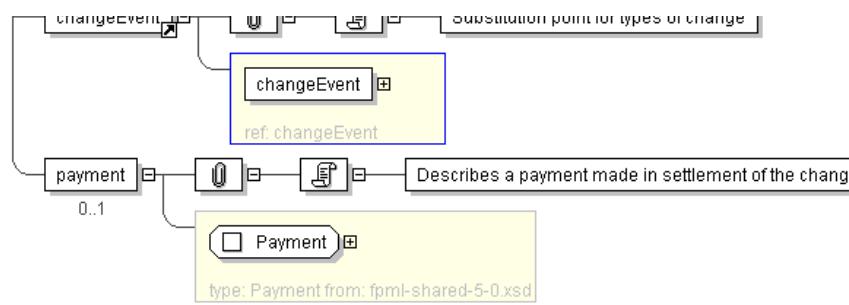
<effectiveDate> xsd:date </effectiveDate> [1]
'The date on which the change become effective'

<changeEvent> ... </changeEvent> [1]
'Substitution point for types of change'

<payment> Payment </payment> [0..1]
'Describes a payment made in settlement of the change.'

</...>
```

**Diagram**

**Schema Component Representation**

```

<xsd:complexType name="TradeChangeContent">
  <xsd:sequence>
    <xsd:choice minOccurs="0">
      <xsd:element name="oldTradeIdentifier" type="PartyTradeIdentifier" />
      <xsd:element name="oldTrade" type="Trade" />
    </xsd:choice>
    <xsd:element name="trade" type="Trade" />
    <xsd:element name="effectiveDate" type="xsd:date" />
    <xsd:element ref="changeEvent" />
    <xsd:element name="payment" type="Payment" minOccurs="0" />
  </xsd:sequence>
</xsd:complexType>

```

top

**Complex Type: TradeNotionalChange**

<b>Super-types:</b>	<a href="#">TradeChangeBase</a> < <b>TradeNotionalChange</b> (by extension)
<b>Sub-types:</b>	None

<b>Name</b>	TradeNotionalChange
<b>Used by (from the same schema document)</b>	Model Group <a href="#">Events.model</a> , Model Group <a href="#">Events.model</a>
<b>Abstract</b>	no
<b>Documentation</b>	A structure describing a change to the trade notional.

**XML Instance Representation**

```

<...>
<tradeIdentifier> PartyTradeIdentifier </tradeIdentifier> [1..*]
<agreementDate> xsd:date </agreementDate> [1]
'The date on which the change was agreed.'

<effectiveDate> xsd:date </effectiveDate> [1]
'The date on which the change become effective.'

<payment> Payment </payment> [0..1]
'Describes a payment made in settlement of the change.'

Start Choice [1]
<changeInNotionalAmount> Money </changeInNotionalAmount> [1]
'Specifies the fixed amount by which the Notional Amount changes'

<outstandingNotionalAmount> Money </outstandingNotionalAmount> [1]
'Specifies the Notional amount after the Change'

```

```

<changeInNumberOfOptions> xsd:decimal </changeInNumberOfOptions> [1]
'Specifies the fixed amount by which the Number of Options changes'

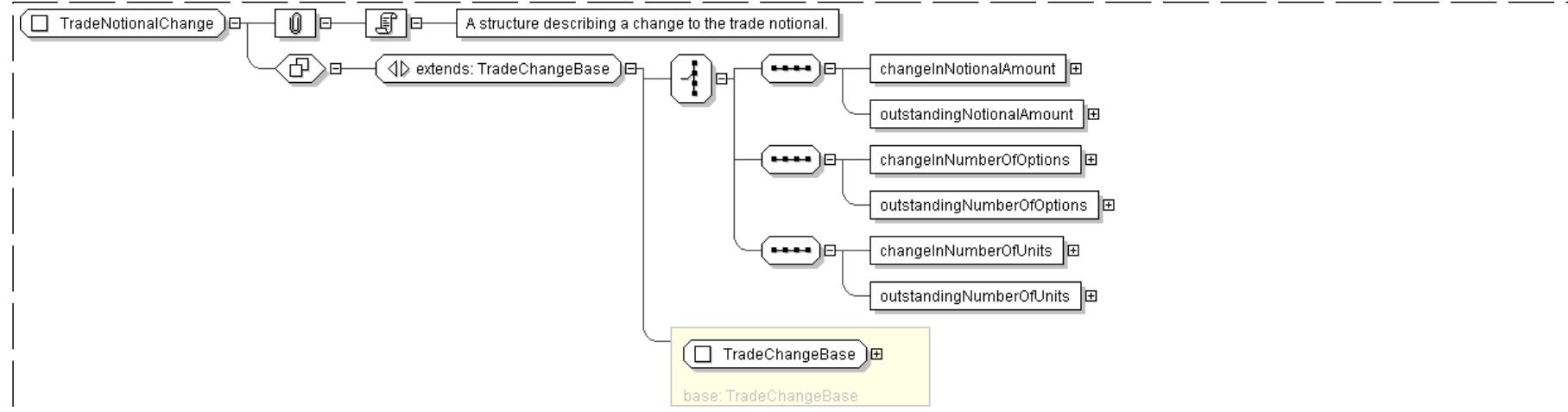
<outstandingNumberOfOptions> xsd:decimal </outstandingNumberOfOptions> [1]
'Specifies the Number of Options after the Change.'

<changeInNumberOfUnits> xsd:decimal </changeInNumberOfUnits> [1]
'Specifies the fixed amount by which the Number of Units changes'

<outstandingNumberOfUnits> xsd:decimal </outstandingNumberOfUnits> [1]
'Specifies the Number of Units'

End Choice
</...>

```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="TradeNotionalChange">
  <xsd:complexContent>
    <xsd:extension base=" TradeChangeBase ">
      <xsd:choice>
        <xsd:sequence>
          <xsd:element name="changeInNotionalAmount" type=" Money "/>
          <xsd:element name="outstandingNotionalAmount" type=" Money "/>
        </xsd:sequence>
        <xsd:sequence>
          <xsd:element name="changeInNumberOfOptions" type=" xsd:decimal "/>
          <xsd:element name="outstandingNumberOfOptions" type=" xsd:decimal "/>
        </xsd:sequence>
        <xsd:sequence>
          <xsd:element name="changeInNumberOfUnits" type=" xsd:decimal "/>
          <xsd:element name="outstandingNumberOfUnits" type=" xsd:decimal "/>
        </xsd:sequence>
      </xsd:choice>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

**Complex Type: TradeNovationContent**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None

<b>Name</b>	TradeNovationContent
<b>Used by (from the same schema document)</b>	Model Group <a href="#">Events.model</a>
<b>Abstract</b>	no
<b>Documentation</b>	A structure describing a novation.

**XML Instance Representation**

```

<...>
Start Choice [1]
Start Choice [1]
'Choice between identification and representation of the new contract.'

<newTradeIdentifier> PartyTradeIdentifier </newTradeIdentifier> [1..*]
'Indicates a reference to the new trade between the transferee and the remaining party.'

<newTrade> Trade </newTrade> [1]
'Indicates the new trade between the transferee and the remaining party.'

End Choice
Start Choice [1]
<oldTradeIdentifier> PartyTradeIdentifier </oldTradeIdentifier> [1..*]
'Indicates a reference to the original trade between the transferor and the remaining party.'

<oldTrade> Trade </oldTrade> [1]
'Indicates the original trade between the transferor and the remaining party.'

End Choice
Start Choice [0..1]
'Choice between identification and representation of the new contract.'

<newTradeIdentifier> PartyTradeIdentifier </newTradeIdentifier> [1..*]
'Indicates a reference to the new trade between the transferee and the remaining party.'

<newTrade> Trade </newTrade> [1]
'Indicates the original trade between the transferor and the remaining party.'

End Choice
End Choice
<transferor> PartyReference </transferor> [1]
'A pointer style reference to a party identifier defined elsewhere in the document. In a
three-way novation the party referenced is the Transferor (outgoing party) in the novation.
The Transferor means a party which transfers by novation to a Transferee all of its
rights, liabilities, duties and obligations with respect to a Remaining Party. In a four-
way novation the party referenced is Transferor 1 which transfers by novation to Transferee
1 all of its rights, liabilities, duties and obligations with respect to Transferor 2.
ISDA 2004 Novation Term: Transferor (three-way novation) or Transferor 1 (four-way novation).'

<transferorAccount> AccountReference </transferorAccount> [0..1]
<transferee> PartyReference </transferee> [1]
'A pointer style reference to a party identifier defined elsewhere in the document. In a
three-way novation the party referenced is the Transferee (incoming party) in the
novation. Transferee means a party which accepts by way of novation all rights,
liabilities, duties and obligations of a Transferor with respect to a Remaining Party. In
a four-way novation the party referenced is Transferee 1 which accepts by way of novation

```

*the rights, liabilities, duties and obligations of Transferor 1. ISDA 2004 Novation Term: Transferee (three-way novation) or Transferee 1 (four-way novation).'*

<transfereeAccount> AccountReference </transfereeAccount> [0..1]  
<remainingParty> PartyReference </remainingParty> [1]

'A pointer style reference to a party identifier defined elsewhere in the document. In a three-way novation the party referenced is the Remaining Party in the novation. Remaining Party means a party which consents to a Transferor's transfer by novation and the acceptance thereof by the Transferee of all of the Transferor's rights, liabilities, duties and obligations with respect to such Remaining Party under and with respect of the Novated Amount of a transaction. In a four-way novation the party referenced is Transferor 2 per the ISDA definition and acts in the role of a Transferor. Transferor 2 transfers by novation to Transferee 2 all of its rights, liabilities, duties and obligations with respect to Transferor 1. ISDA 2004 Novation Term: Remaining Party (three-way novation) or Transferor 2 (four-way novation).'

<remainingPartyAccount> AccountReference </remainingPartyAccount> [0..1]  
<otherRemainingParty> PartyReference </otherRemainingParty> [0..1]

'A pointer style reference to a party identifier defined elsewhere in the document. This element is not applicable in a three-way novation and should be omitted. In a four-way novation the party referenced is Transferee 2. Transferee 2 means a party which accepts by way of novation the rights, liabilities, duties and obligations of Transferor 2. ISDA 2004 Novation Term: Transferee 2 (four-way novation).'

<otherRemainingPartyAccount> AccountReference </otherRemainingPartyAccount> [0..1]  
<novationDate> xsd:date </novationDate> [1]

'Specifies the date that one party's legal obligations with regard to a trade are transferred to another party. It corresponds to the Novation Date section of the 2004 ISDA Novation Definitions, section 1.16.'

<novationTradeDate> xsd:date </novationTradeDate> [0..1]

'Specifies the date the parties agree to assign or novate a Contract. If this element is not specified, the novationContractDate will be deemed to be the novationDate. It corresponds to the Novation Trade Date section of the 2004 ISDA Novation Definitions, section 1.17.'

#### Start Choice [1]

'Choice for expressing the novated amount as either a money amount, number of options, or number of units, according the the financial product which is being novated.'

<novatedAmount> Money </novatedAmount> [1]

'The amount which represents the portion of the Old Contract being novated.'

<remainingAmount> Money </remainingAmount> [0..1]

'The amount which represents the portion of the Old Contract not being novated.'

<novatedNumberOfOptions> xsd:decimal </novatedNumberOfOptions> [1]

'The number of options which represent the portion of the Old Contract being novated.'

<remainingNumberOfOptions> xsd:decimal </remainingNumberOfOptions> [0..1]

'The number of options which represent the portion of the Old Contract not being novated.'

<novatedNumberofUnits> xsd:decimal </novatedNumberofUnits> [1]

'The number of options which represent the portion of the Old Contract being novated.'

<remainingNumberofUnits> xsd:decimal </remainingNumberofUnits> [0..1]

'The number of options which represent the portion of the Old Contract not being novated.'

#### End Choice

<fullFirstCalculationPeriod> xsd:boolean </fullFirstCalculationPeriod> [0..1]

'This element corresponds to the applicability of the Full First Calculation Period as

defined in the 2004 ISDA Novation Definitions, section 1.20.'

<firstPeriodStartDate> FirstPeriodStartDate </firstPeriodStartDate> [0..2]

'Element that is used to be able to make sense of the "new transaction" without requiring reference back to the "old transaction". In the case of interest rate products there are potentially 2 "first period start dates" to reference - one with respect to each party to the new transaction. For Credit Default Swaps there is just the one with respect to the party that is the fixed rate payer.'

<nonReliance> Empty </nonReliance> [0..1]

'This element corresponds to the non-Reliance section in the 2004 ISDA Novation Definitions, section 2.1 (c) (i). The element appears in the instance document when non-Reliance is applicable.'

<creditDerivativesNotices> CreditDerivativesNotices </creditDerivativesNotices> [0..1]

'This element should be specified if one or more of either a Credit Event Notice, Notice of Publicly Available Information, Notice of Physical Settlement or Notice of Intended Physical Settlement, as applicable, has been delivered by or to the Transferor or the Remaining Party. The type of notice or notices that have been delivered should be indicated by setting the relevant boolean element value(s) to true. The absence of the element means that no Credit Event Notice, Notice of Publicly Available Information, Notice of Physical Settlement or Notice of Intended Physical Settlement, as applicable, has been delivered by or to the Transferor or the Remaining Party.'

<contractualDefinitions> ContractualDefinitions </contractualDefinitions> [0..\*]

'The definitions (such as those published by ISDA) that will define the terms of the novation transaction.'

<contractualTermsSupplement> ContractualTermsSupplement </contractualTermsSupplement> [0..\*]

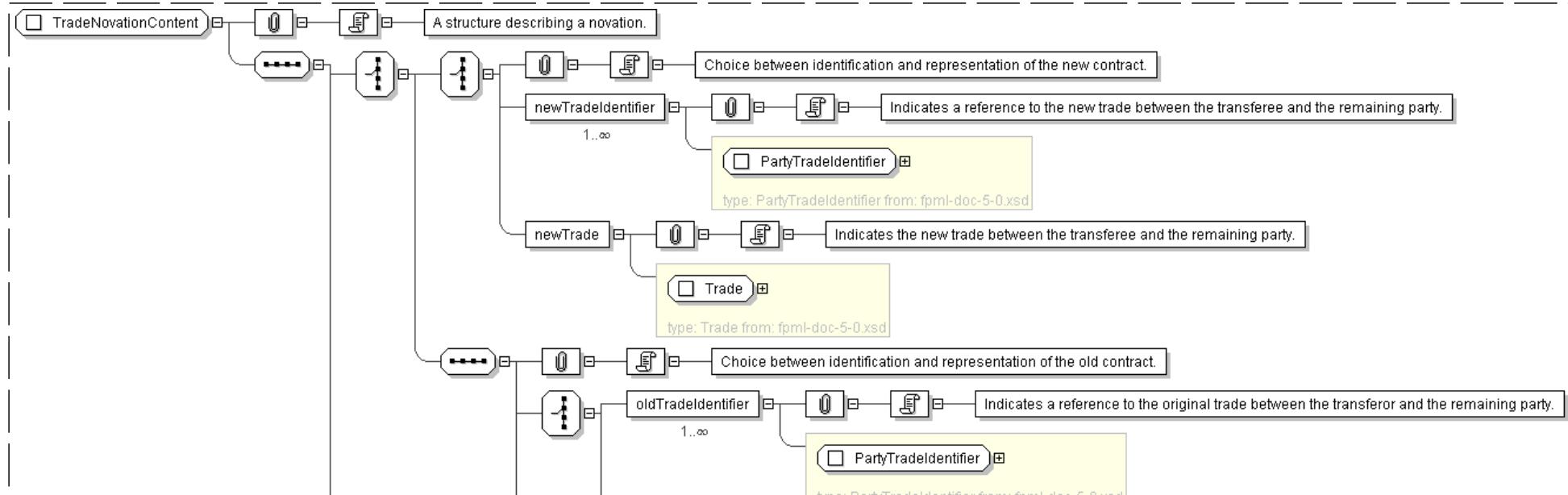
'A contractual supplement (such as those published by ISDA) that will apply to the trade.'

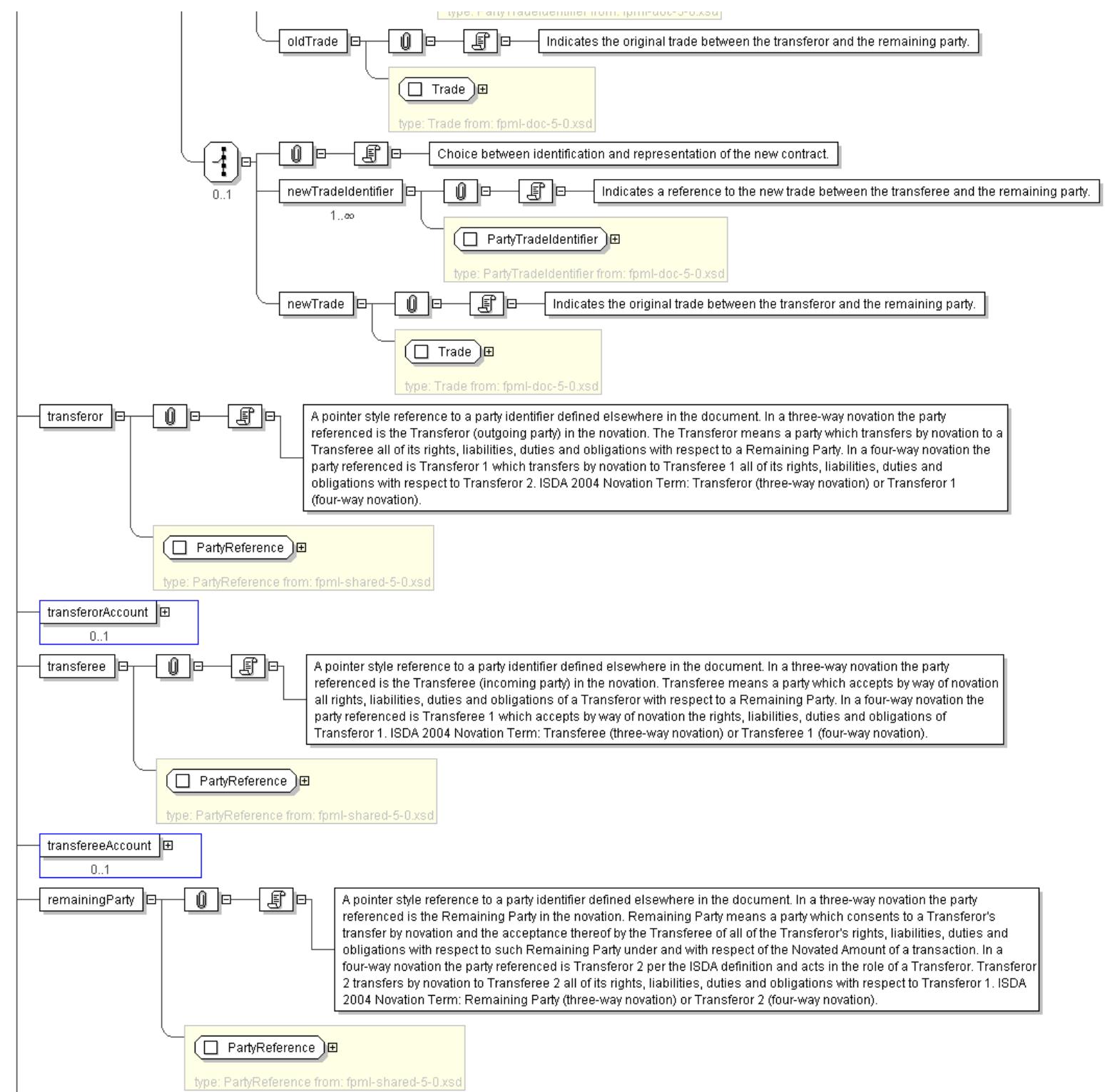
<payment> Payment </payment> [0..1]

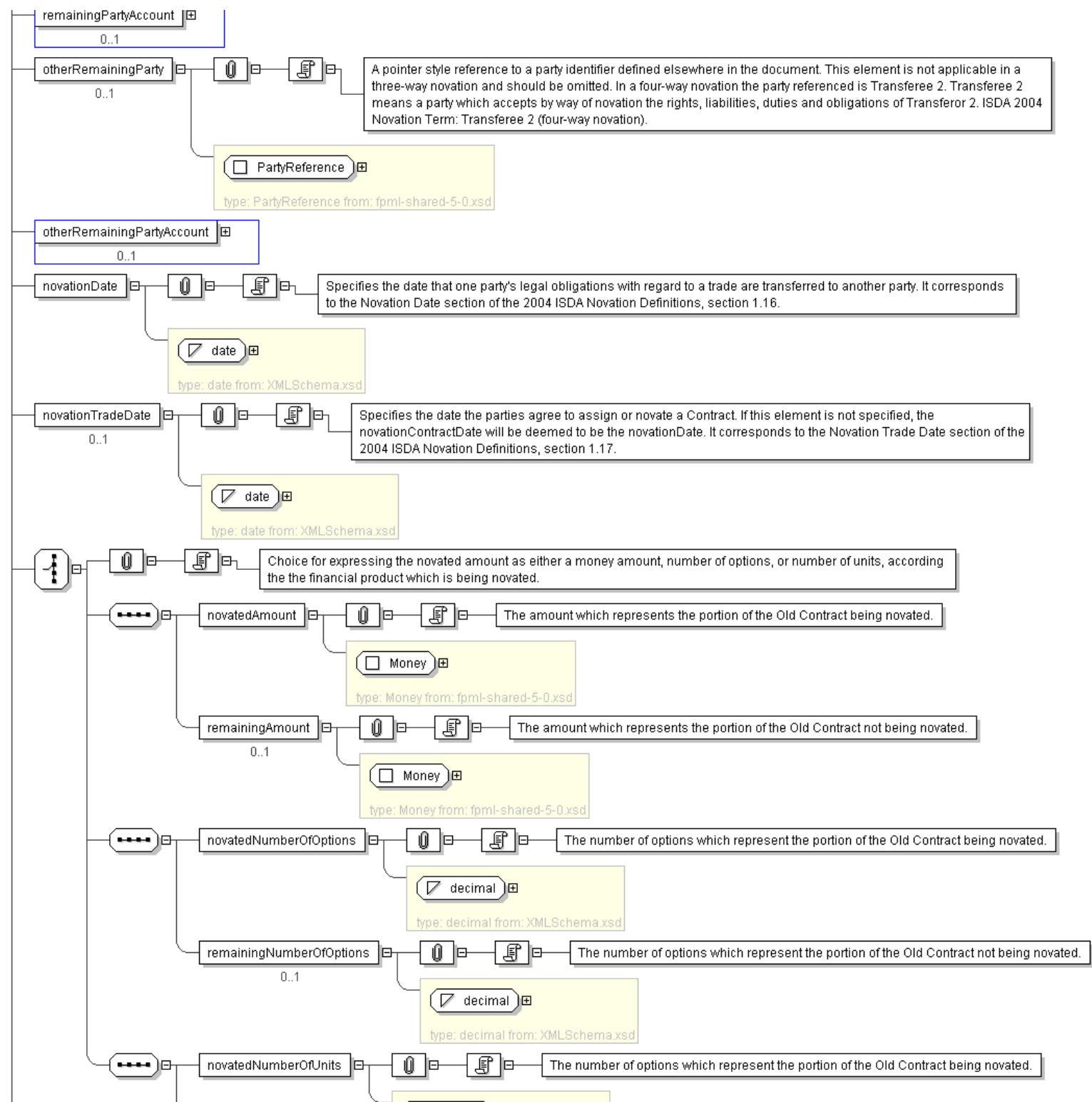
'Describes a payment made in settlement of the novation.'

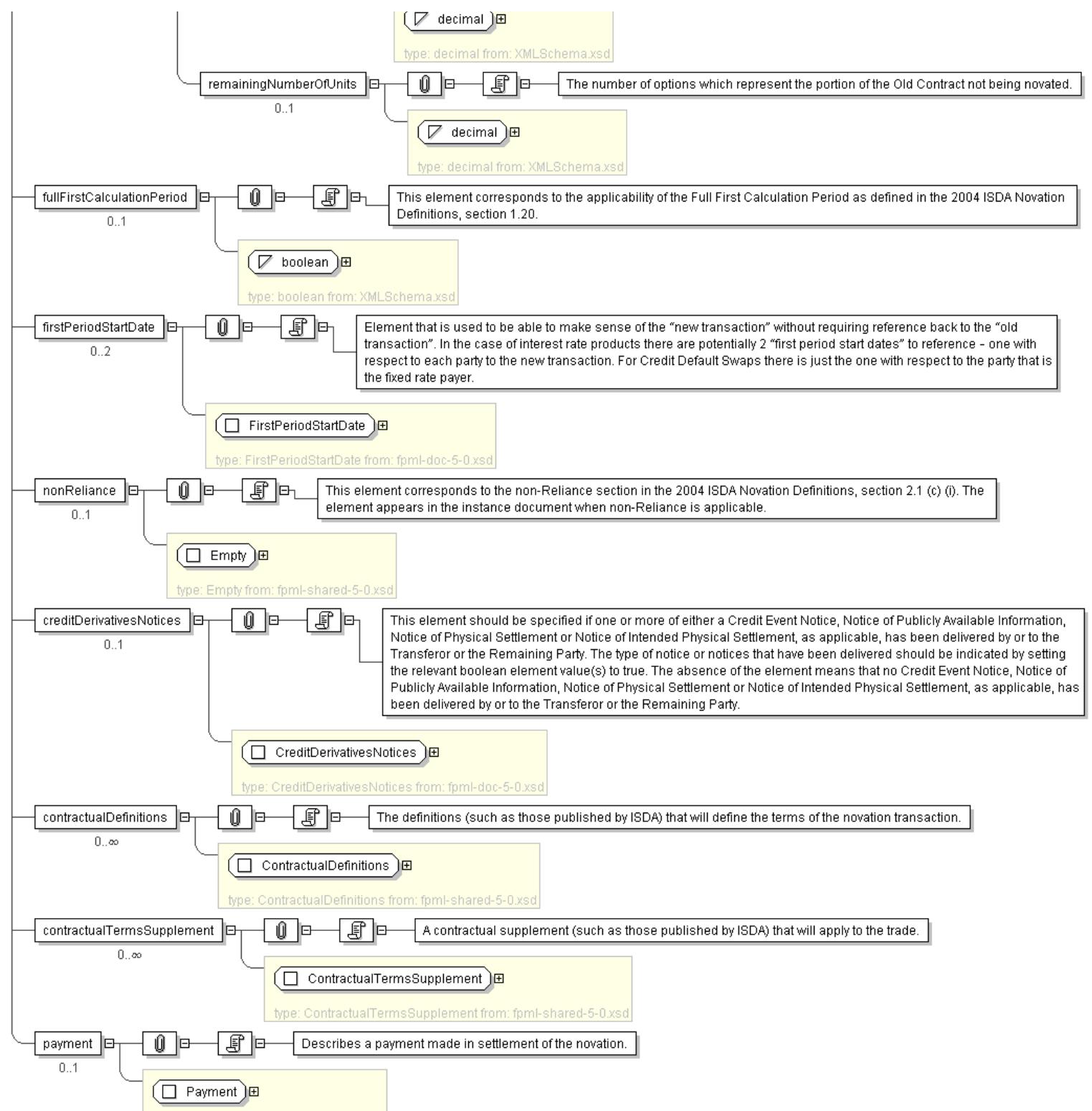
<!-->

#### Diagram









type: Payment from: fpml-shared-5-0.xsd

## Schema Component Representation

```

<xsd:complexType name="TradeNovationContent">
  <xsd:sequence>
    <xsd:choice>
      <xsd:choice>
        <xsd:element name="newTradeIdentifier" type=" PartyTradeIdentifier " maxOccurs="unbounded" />
        <xsd:element name="newTrade" type=" Trade " />
      </xsd:choice>
    <xsd:sequence>
      <xsd:choice>
        <xsd:element name="oldTradeIdentifier" type=" PartyTradeIdentifier " maxOccurs="unbounded" />
        <xsd:element name="oldTrade" type=" Trade " />
      </xsd:choice>
    <xsd:choice minOccurs="0">
      <xsd:element name="newTradeIdentifier" type=" PartyTradeIdentifier " maxOccurs="unbounded" />
      <xsd:element name="newTrade" type=" Trade " />
    </xsd:choice>
  </xsd:sequence>
</xsd:choice>
<xsd:element name="transferor" type=" PartyReference " />
<xsd:element name="transferorAccount" type=" AccountReference " minOccurs="0" />
<xsd:element name="transferee" type=" PartyReference " />
<xsd:element name="transfereeAccount" type=" AccountReference " minOccurs="0" />
<xsd:element name="remainingParty" type=" PartyReference " />
<xsd:element name="remainingPartyAccount" type=" AccountReference " minOccurs="0" />
<xsd:element name="otherRemainingParty" type=" PartyReference " minOccurs="0" />
<xsd:element name="otherRemainingPartyAccount" type=" AccountReference " minOccurs="0" />
<xsd:element name="novationDate" type=" xsd:date " />
<xsd:element name="novationTradeDate" type=" xsd:date " minOccurs="0" />
<xsd:choice>
  <xsd:sequence>
    <xsd:element name="novatedAmount" type=" Money " />
    <xsd:element name="remainingAmount" type=" Money " minOccurs="0" />
  </xsd:sequence>
  <xsd:sequence>
    <xsd:element name="novatedNumberOfOptions" type=" xsd:decimal " />
    <xsd:element name="remainingNumberOfOptions" type=" xsd:decimal " minOccurs="0" />
  </xsd:sequence>
  <xsd:sequence>
    <xsd:element name="novatedNumberOfUnits" type=" xsd:decimal " />
    <xsd:element name="remainingNumberOfUnits" type=" xsd:decimal " minOccurs="0" />
  </xsd:sequence>
</xsd:choice>
<xsd:element name="fullFirstCalculationPeriod" type=" xsd:boolean " minOccurs="0" />
<xsd:element name="firstPeriodStartDate" type=" FirstPeriodStartDate "
minOccurs="0" maxOccurs="2" />
<xsd:element name="nonReliance" type=" Empty " minOccurs="0" />
<xsd:element name="creditDerivativesNotices" type=" CreditDerivativesNotices " minOccurs="0" />
<xsd:element name="contractualDefinitions" type=" ContractualDefinitions "
minOccurs="0" maxOccurs="unbounded" />
<xsd:element name="contractualTermsSupplement" type=" ContractualTermsSupplement
" minOccurs="0" maxOccurs="unbounded" />
<xsd:element name="payment" type=" Payment " minOccurs="0" />
</xsd:sequence>
</xsd:complexType>

```

top

Name	AgreementAndEffectiveDates.model
Used by (from the same schema document)	Complex Type <a href="#">TradeAmendmentContent</a> , Complex Type <a href="#">TradeChangeBase</a>
Documentation	A model group defining agreement and effective dates.

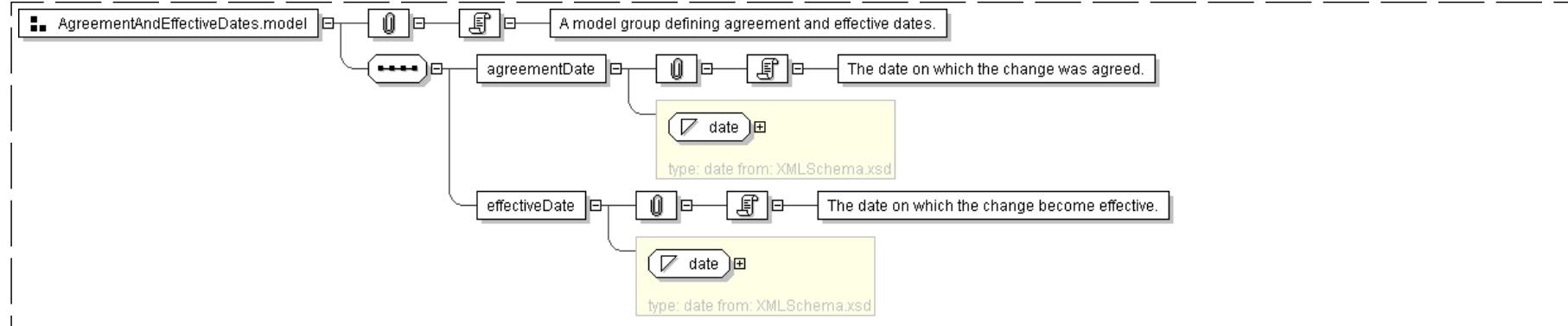
**XML Instance Representation**

```
<agreementDate> xsd:date </agreementDate> [1]
```

'The date on which the change was agreed.'

```
<effectiveDate> xsd:date </effectiveDate> [1]
```

'The date on which the change become effective.'

**Diagram****Schema Component Representation**

```

<xsd:group name="AgreementAndEffectiveDates.model">
  <xsd:sequence>
    <xsd:element name="agreementDate" type="xsd:date" />
    <xsd:element name="effectiveDate" type="xsd:date" />
  </xsd:sequence>
</xsd:group>
  
```

top

**Model Group: Events.model**

Name	Events.model
Used by (from the same schema document)	Complex Type <a href="#">EventProposedMatch</a> , Complex Type <a href="#">EventsChoice</a>
Documentation	Choice between a trade and a post-trade event.

**XML Instance Representation**

```
Start Choice [1]
```

```
  <trade> Trade </trade> [1]
```

```
  <amendment> TradeAmendmentContent </amendment> [1]
```

```
  <increase> TradeNotionalChange </increase> [1]
```

```
  <termination> TradeNotionalChange </termination> [1]
```

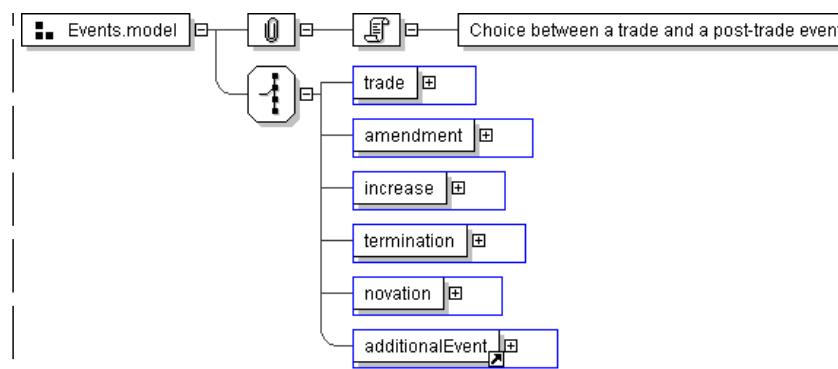
```
  <novation> TradeNovationContent </novation> [1]
```

```
  <additionalEvent> ... </additionalEvent> [1]
```

```
End Choice
```

**Diagram**

|

**Schema Component Representation**

```

<xsd:group name="Events.model">
  <xsd:choice>
    <xsd:element name="trade" type=" Trade " />
    <xsd:element name="amendment" type=" TradeAmendmentContent " />
    <xsd:element name="increase" type=" TradeNotionalChange " />
    <xsd:element name="termination" type=" TradeNotionalChange " />
    <xsd:element name="novation" type=" TradeNovationContent " />
    <xsd:element ref=" additionalEvent " />
  </xsd:choice>
</xsd:group>
  
```

[top](#)**Model Group: ProposedMatch.model**

Name	ProposedMatch.model
Used by (from the same schema document)	Complex Type <a href="#">EventProposedMatch</a>

**XML Instance Representation**

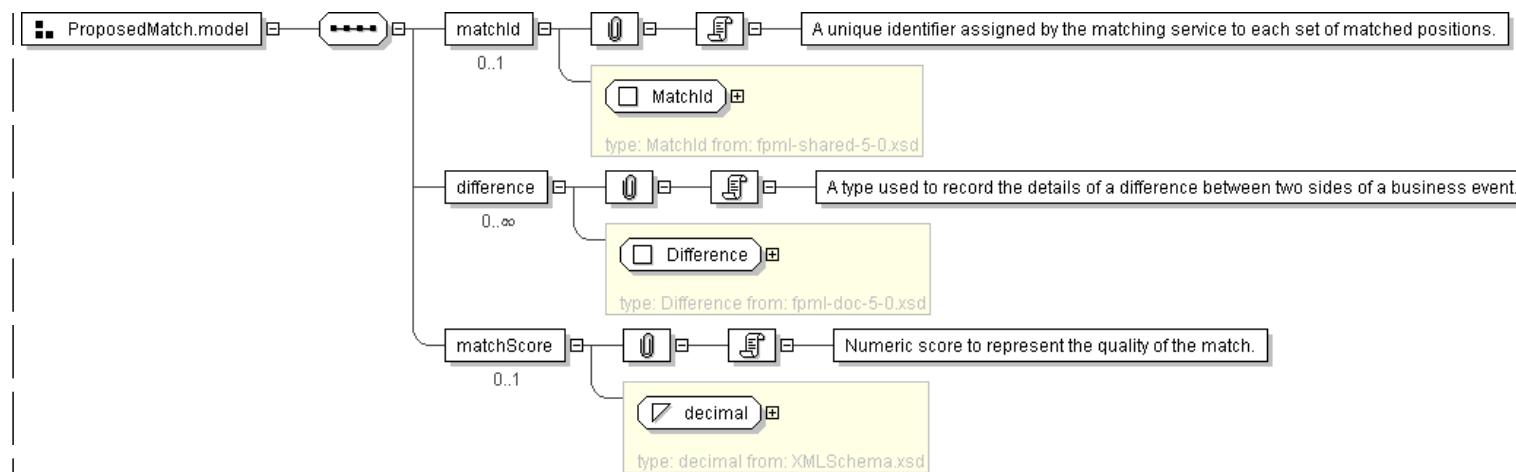
```

<matchId> MatchId </matchId> [0..1]
'A unique identifier assigned by the matching service to each set of matched positions.'

<difference> Difference </difference> [0..*]
'A type used to record the details of a difference between two sides of a business event.'

<matchScore> xsd:decimal </matchScore> [0..1]
'Numeric score to represent the quality of the match.'
  
```

**Diagram**

**Schema Component Representation**

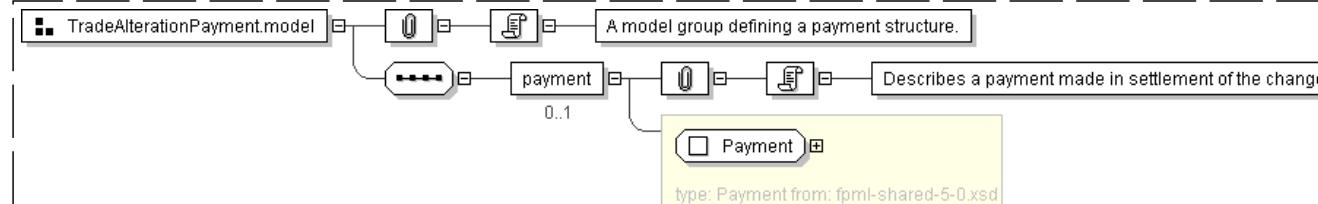
```
<xsd:group name="ProposedMatch.model">
  <xsd:sequence>
    <xsd:element name="matchId" type=" MatchId " minOccurs="0" />
    <xsd:element name="difference" type=" Difference " minOccurs="0" maxOccurs="unbounded" />
    <xsd:element name="matchScore" type=" xsd:decimal " minOccurs="0" />
  </xsd:sequence>
</xsd:group>
```

[top](#)**Model Group: TradeAlterationPayment.model**

<b>Name</b>	TradeAlterationPayment.model
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">TradeAmendmentContent</a> , Complex Type <a href="#">TradeChangeBase</a>
<b>Documentation</b>	A model group defining a payment structure.

**XML Instance Representation**

```
<payment> Payment </payment> [0..1]
'Describes a payment made in settlement of the change.'
```

**Diagram****Schema Component Representation**

```
<xsd:group name="TradeAlterationPayment.model">
  <xsd:sequence>
    <xsd:element name="payment" type=" Payment " minOccurs="0" />
  </xsd:sequence>
</xsd:group>
```

## Legend

**Complex Type:**  
Schema Component Type

**AusAddress**  
Schema Component Name

<b>Super-types:</b>	<a href="#">Address</a> < AusAddress (by extension)
<b>Sub-types:</b>	• <a href="#">QLDAddress</a> (by restriction)

If this schema component is a type definition, its type hierarchy is shown in a gray-bordered box.

<b>Name</b>	AusAddress
<b>Abstract</b>	no

The table above displays the properties of this schema component.

### XML Instance Representation

```
<... country="Australia" >
<unitNo> string </unitNo> [0..1]
<houseNo> string </houseNo> [1]
<street> string </street> [1]
Start Choice [1]
<city> string </city> [1]
<town> string </town> [1]
End Choice
<state> AusStates </state> [1]
<postcode> string <><pattern = [1-9][0-9]{3}></postcode> [1]
</...>
```

The XML Instance Representation table above shows the schema component's content as an XML instance.

- The minimum and maximum occurrence of elements and attributes are provided in square brackets, e.g. [0..1].
- Model group information are shown in gray, e.g. Start Choice ... End Choice.
- For type derivations, the elements and attributes that have been added to or changed from the base type's content are shown in **bold**.
- If an element/attribute has a fixed value, the fixed value is shown in green, e.g. country="Australia".
- Otherwise, the type of the element/attribute is displayed.
  - If the element/attribute's type is in the schema, a link is provided to it.
  - For local simple type definitions, the constraints are displayed in angle brackets, e.g. <><pattern = [1-9][0-9]{3}></>.

### Schema Component Representation

```
<complexType name="AusAddress">
<complexContent>
<extension base=" Address ">
<sequence>
<element name="state" type=" AusStates " />
<element name="postcode">
<simpleType>
<restriction base=" string ">
<pattern value="[1-9][0-9]{3}" />
</restriction>
</simpleType>
</element>
</sequence>
<attribute name="country" type=" string " fixed="Australia" />
</extension>
</complexContent>
</complexType>
```

The Schema Component Representation table above displays the underlying XML representation of the schema component. (Annotations are not shown.)

## Glossary

**Abstract** (Applies to complex type definitions and element declarations). An abstract element or complex type cannot be used to validate an element instance. If there is a reference to an abstract element, only element declarations that can substitute the abstract element can be used to validate the instance. For references to abstract type definitions, only derived types can be used.

**All Model Group** Child elements can be provided *in any order* in instances. See: <http://www.w3.org/TR/xmlschema-1/#element-all>.

**Choice Model Group** *Only one* from the list of child elements and model groups can be provided in instances. See: <http://www.w3.org/TR/xmlschema-1/#element-choice>.

**Collapse Whitespace Policy** Replace tab, line feed, and carriage return characters with space character (Unicode character 32). Then, collapse contiguous sequences of space characters into single space character, and remove leading and trailing space characters.

**Disallowed Substitutions** (Applies to element declarations). If *substitution* is specified, then *substitution group* members cannot be used in place of the given element declaration to validate element instances. If *derivation methods*, e.g. extension, restriction, are specified, then the given element declaration will not validate element instances that have types derived from the element declaration's type using the specified derivation methods. Normally, element instances can override their declaration's type by specifying an *xsi:type* attribute.

**Key Constraint** Like *Uniqueness Constraint*, but additionally requires that the specified value(s) must be provided. See: [http://www.w3.org/TR/xmlschema-1/#clentity-constraint\\_Definitions](http://www.w3.org/TR/xmlschema-1/#clentity-constraint_Definitions).

**Key Reference Constraint** Ensures that the specified value(s) must match value(s) from a *Key Constraint* or *Uniqueness Constraint*. See: [http://www.w3.org/TR/xmlschema-1/#clentity-constraint\\_Definitions](http://www.w3.org/TR/xmlschema-1/#clentity-constraint_Definitions).

**Model Group** Groups together element content, specifying the order in which the element content can occur and the number of times the group of element content may be repeated. See: [http://www.w3.org/TR/xmlschema-1/#Model\\_Groups](http://www.w3.org/TR/xmlschema-1/#Model_Groups).

**Nillable** (Applies to element declarations). If an element declaration is nillable, instances can use the *xsi:nil* attribute. The *xsi:nil* attribute is the boolean attribute, *nil*, from the <http://www.w3.org/2001/XMLSchema-instance> namespace. If an element instance has an *xsi:nil* attribute set to true, it can be left empty, even though its element declaration may have required content.

**Notation** A notation is used to identify the format of a piece of data. Values of elements and attributes that are of type, NOTATION, must come from the names of declared notations. See: [http://www.w3.org/TR/xmlschema-1/#cNotation\\_Declarations](http://www.w3.org/TR/xmlschema-1/#cNotation_Declarations).

**Preserve Whitespace Policy** Preserve whitespaces exactly as they appear in instances.

**Prohibited Derivations** (Applies to type definitions). Derivation methods that cannot be used to create sub-types from a given type definition.

**Prohibited Substitutions** (Applies to complex type definitions). Prevents sub-types that have been derived using the specified derivation methods from validating element instances in place of the given type definition.

**Replace Whitespace Policy** Replace tab, line feed, and carriage return characters with space character (Unicode character 32).

**Sequence Model Group** Child elements and model groups must be provided *in the specified order* in instances. See: <http://www.w3.org/TR/xmlschema-1/#element-sequence>.

**Substitution Group** Elements that are *members* of a substitution group can be used wherever the *head* element of the substitution group is referenced.

**Substitution Group Exclusions** (Applies to element declarations). Prohibits element declarations from nominating themselves as being able to substitute a given element declaration, if they have types that are derived from the original element's type using the specified derivation methods.

**Target Namespace** The target namespace identifies the namespace that components in this schema belongs to. If no target namespace is provided, then the schema components do not belong to any namespace.

**Uniqueness Constraint** Ensures uniqueness of an element/attribute value, or a combination of values, within a specified scope. See: [http://www.w3.org/TR/xmlschema-1/#cldentity-constraint\\_Definitions](http://www.w3.org/TR/xmlschema-1/#cldentity-constraint_Definitions).

[top](#)

---

Generated by [coOxygen](#)/[XML Editor](#) using a modified version of [xs3p](#) that adds schema diagrams and chunking support.

# XML Schema Documentation

## Table of Contents

- [Schema Document Properties](#)
- [Global Declarations](#)
  - [Element: creditDefaultSwap](#)
  - [Element: creditDefaultSwapOption](#)
- [Global Definitions](#)
  - Complex Type: [AdditionalFixedPayments](#)
  - Complex Type: [AdditionalTerm](#)
  - Complex Type: [AdjustedPaymentDates](#)
  - Complex Type: [BasketReferenceInformation](#)
  - Complex Type: [CalculationAmount](#)
  - Complex Type: [CashSettlementTerms](#)
  - Complex Type: [CreditDefaultSwap](#)
  - Complex Type: [CreditDefaultSwapOption](#)
  - Complex Type: [CreditOptionStrike](#)
  - Complex Type: [DeliverableObligations](#)
  - Complex Type: [EntityType](#)
  - Complex Type: [FeeLeg](#)
  - Complex Type: [FixedAmountCalculation](#)
  - Complex Type: [FixedRate](#)
  - Complex Type: [FixedRateReference](#)
  - Complex Type: [FloatingAmountEvents](#)
  - Complex Type: [FloatingAmountProvisions](#)
  - Complex Type: [GeneralTerms](#)
  - Complex Type: [IndexAnnexSource](#)
  - Complex Type: [IndexId](#)
  - Complex Type: [IndexName](#)
  - Complex Type: [IndexReferenceInformation](#)
  - Complex Type: [InitialPayment](#)
  - Complex Type: [InterestShortFall](#)
  - Complex Type: [LoanParticipation](#)
  - Complex Type: [MatrixSource](#)
  - Complex Type: [MultipleValuationDates](#)
  - Complex Type: [NotDomesticCurrency](#)
  - Complex Type: [Obligations](#)
  - Complex Type: [PCDeliverableObligationCharac](#)
  - Complex Type: [PeriodicPayment](#)
  - Complex Type: [PhysicalSettlementPeriod](#)
  - Complex Type: [PhysicalSettlementTerms](#)
  - Complex Type: [ProtectionTerms](#)
  - Complex Type: [ProtectionTermsReference](#)
  - Complex Type: [ReferenceInformation](#)
  - Complex Type: [ReferenceObligation](#)
  - Complex Type: [ReferencePair](#)
  - Complex Type: [ReferencePool](#)
  - Complex Type: [ReferencePoolItem](#)
  - Complex Type: [ScheduledTerminationDate](#)
  - Complex Type: [SettledEntityMatrix](#)
  - Complex Type: [SettlementTerms](#)
  - Complex Type: [SettlementTermsReference](#)
  - Complex Type: [SinglePayment](#)
  - Complex Type: [SingleValuationDate](#)
  - Complex Type: [SpecifiedCurrency](#)
  - Complex Type: [Tranche](#)
  - Complex Type: [ValuationDate](#)
  - Model Group: [FixedRecovery.model](#)
- [Legend](#)

## Schema Document Properties

<b>Target Namespace</b>	<a href="http://www.fpml.org/FpML-5/confirmation">http://www.fpml.org/FpML-5/confirmation</a>
<b>Version</b>	\$Revision: 2350 \$
<b>Element and Attribute Namespaces</b>	<ul style="list-style-type: none"> <li>Global element and attribute declarations belong to this schema's target namespace.</li> <li>By default, local element declarations belong to this schema's target namespace.</li> <li>By default, local attribute declarations have no namespace.</li> </ul>
<b>Schema Composition</b>	<ul style="list-style-type: none"> <li>This schema includes components from the following schema document(s):           <ul style="list-style-type: none"> <li><a href="#">fpml-option-shared-5-0.xsd</a></li> </ul> </li> </ul>

## Declared Namespaces

Prefix	Namespace
Default namespace	<a href="http://www.fpml.org/FpML-5/confirmation">http://www.fpml.org/FpML-5/confirmation</a>
xml	<a href="http://www.w3.org/XML/1998/namespace">http://www.w3.org/XML/1998/namespace</a>
fpml-annotation	<a href="http://www.fpml.org/annotation">http://www.fpml.org/annotation</a>
xsd	<a href="http://www.w3.org/2001/XMLSchema">http://www.w3.org/2001/XMLSchema</a>
ecore	<a href="http://www.eclipse.org/emf/2002/Ecore">http://www.eclipse.org/emf/2002/Ecore</a>

## Schema Component Representation

```

<xsd:schema targetNamespace="http://www.fpml.org/FpML-5/confirmation"
  documentRoot="FpML" nsPrefix="conf" package="org.fpml.confirmation" version="$Revision: 2350 "
  $" elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xsd:include schemaLocation="fpml-option-shared-5-0.xsd"/>
  ...
</xsd:schema>
  
```

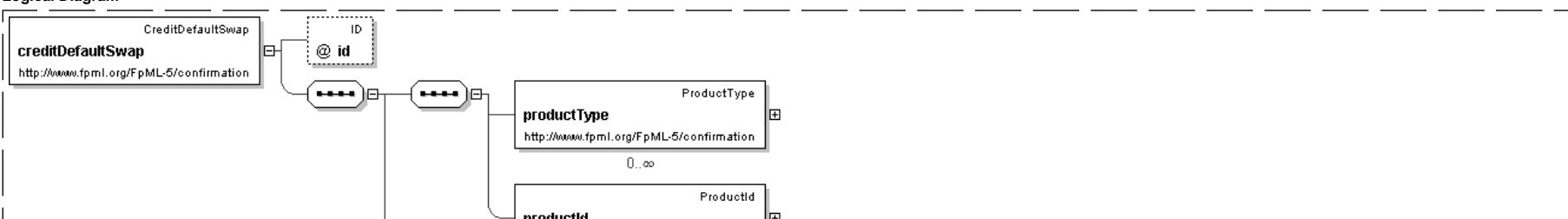
## Global Declarations

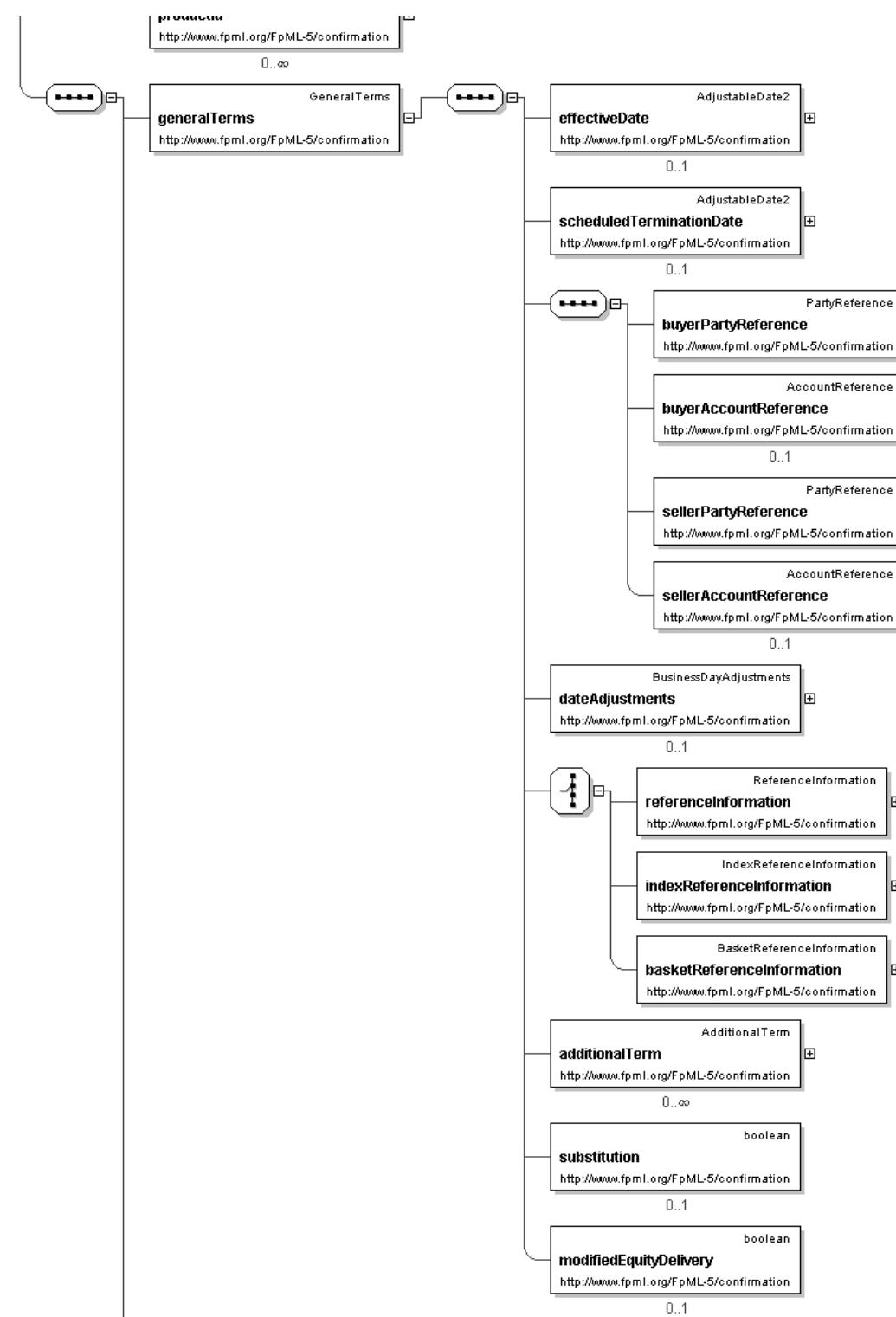
### Element: creditDefaultSwap

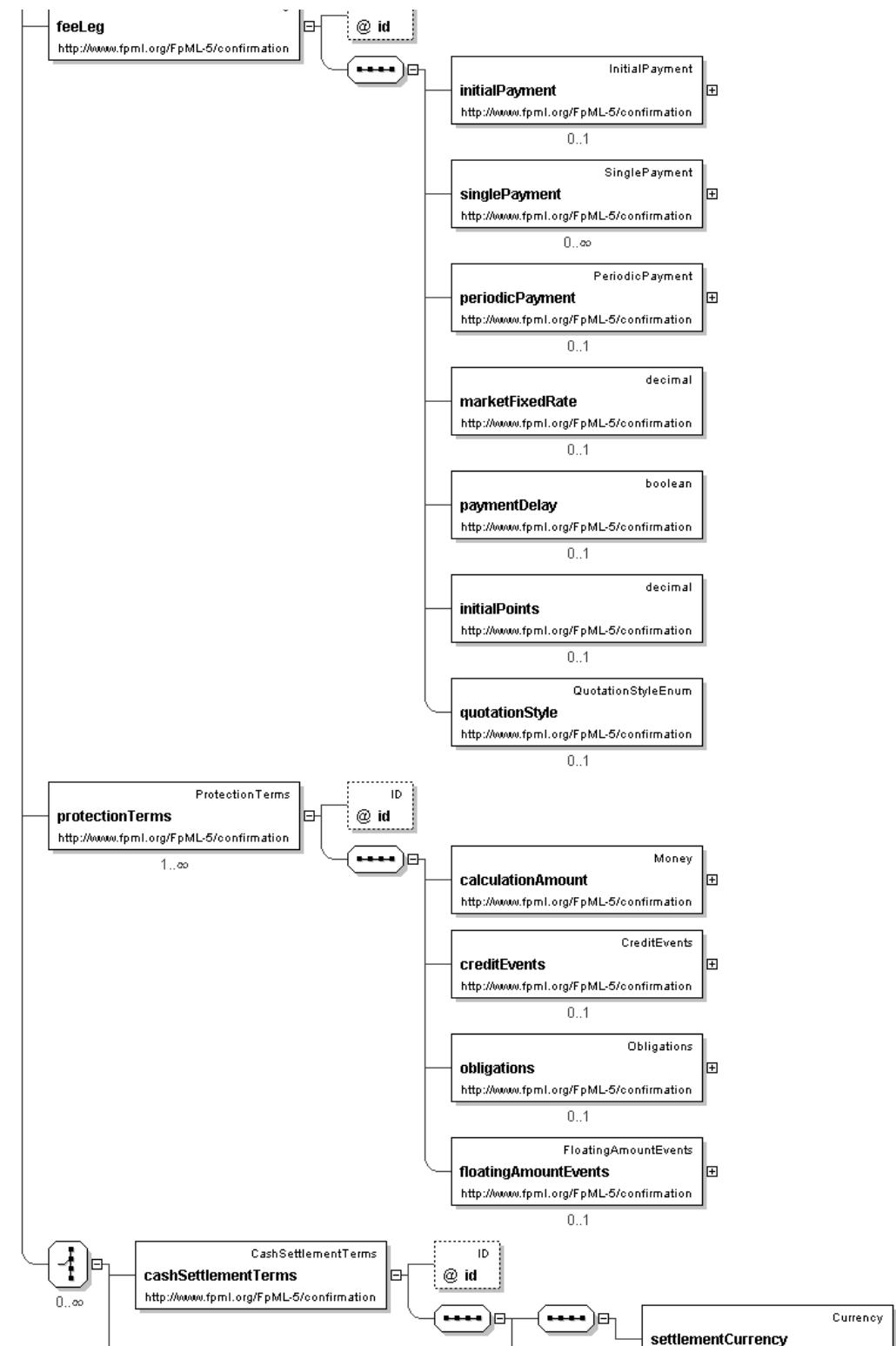
- This element can be used wherever the following element is referenced:
  - [product](#)

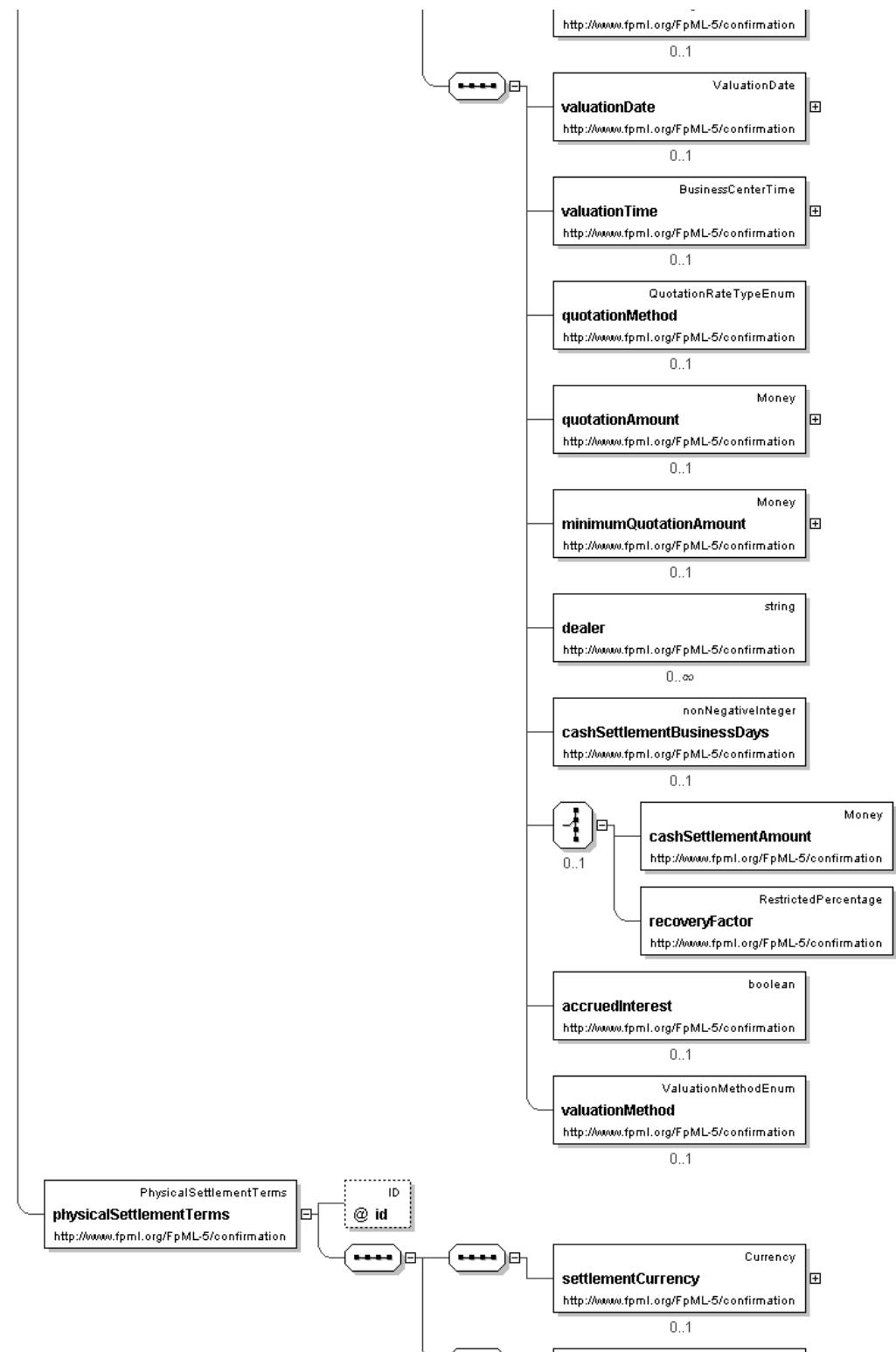
<b>Name</b>	creditDefaultSwap
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">CreditDefaultSwapOption</a>
<b>Type</b>	<a href="#">CreditDefaultSwap</a>
<b>Nillable</b>	no
<b>Abstract</b>	no
<b>Documentation</b>	In a credit default swap one party (the protection seller) agrees to compensate another party (the protection buyer) if a specified company or Sovereign (the reference entity) experiences a credit event, indicating it is or may be unable to service its debts. The protection seller is typically paid a fee and/ or premium, expressed as an annualized percent of the notional in basis points, regularly over the life of the transaction or otherwise as agreed by the parties.

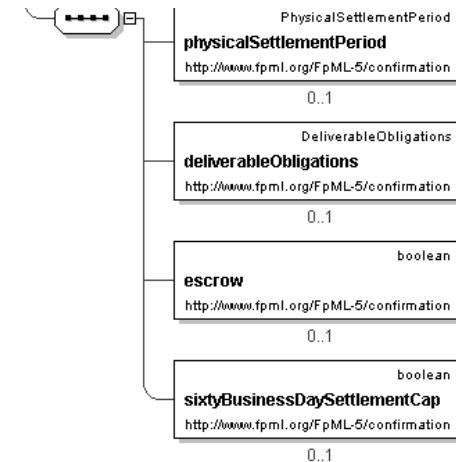
## Logical Diagram









**XML Instance Representation**

```

<creditDefaultSwap
id=" xsd:ID [0..1]">
  <productType> ProductType </productType> [0..*]
  'A classification of the type of product. FpML defines a simple product categorization using
  a coding scheme.'

  <productId> ProductId </productId> [0..*]
  'A product 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.'

  <generalTerms> GeneralTerms </generalTerms> [1]
  'This element contains all the data that appears in the section entitled \"1. General Terms
  \" in the 2003 ISDA Credit Derivatives Confirmation.'

  <feeLeg> FeeLeg </feeLeg> [1]
  'This element contains all the terms relevant to defining the fixed amounts/payments per
  the applicable ISDA definitions.'

  <protectionTerms> ProtectionTerms </protectionTerms> [1..*]
  'This element contains all the terms relevant to defining the applicable floating rate
  payer calculation amount, credit events and associated conditions to settlement, and
  reference obligations.'

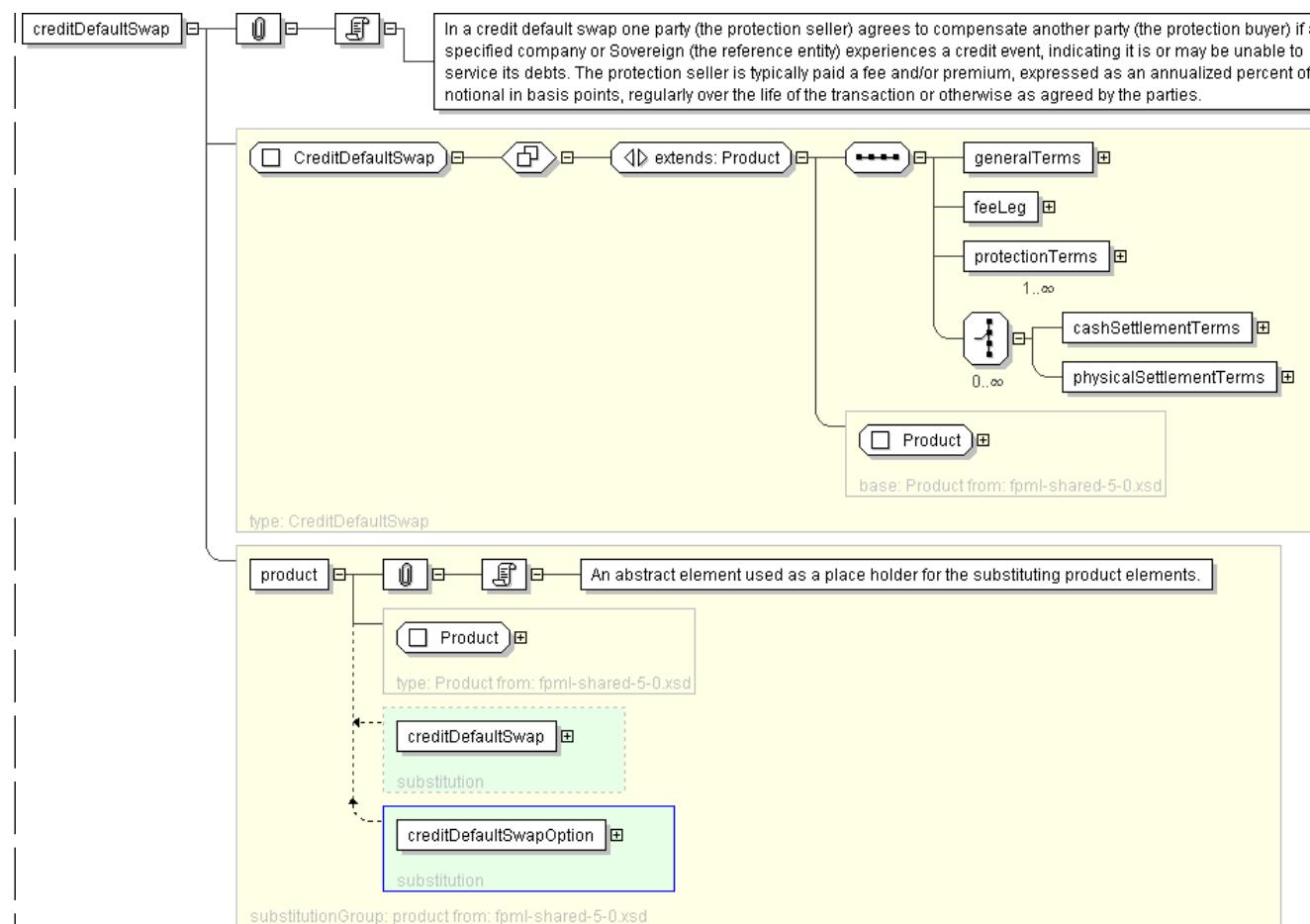
Start Choice [0..*]
  <cashSettlementTerms> CashSettlementTerms </cashSettlementTerms> [1]
  'This element contains all the ISDA terms relevant to cash settlement for when cash
  settlement is applicable. ISDA 2003 Term: Cash Settlement'

  <physicalSettlementTerms> PhysicalSettlementTerms </physicalSettlementTerms> [1]
  'This element contains all the ISDA terms relevant to physical settlement for when
  physical settlement is applicable. ISDA 2003 Term: Physical Settlement'

End Choice
</creditDefaultSwap>

```

**Diagram**

**Schema Component Representation**

```
<xsd:element name="creditDefaultSwap" type=" CreditDefaultSwap " substitutionGroup="product" />
```

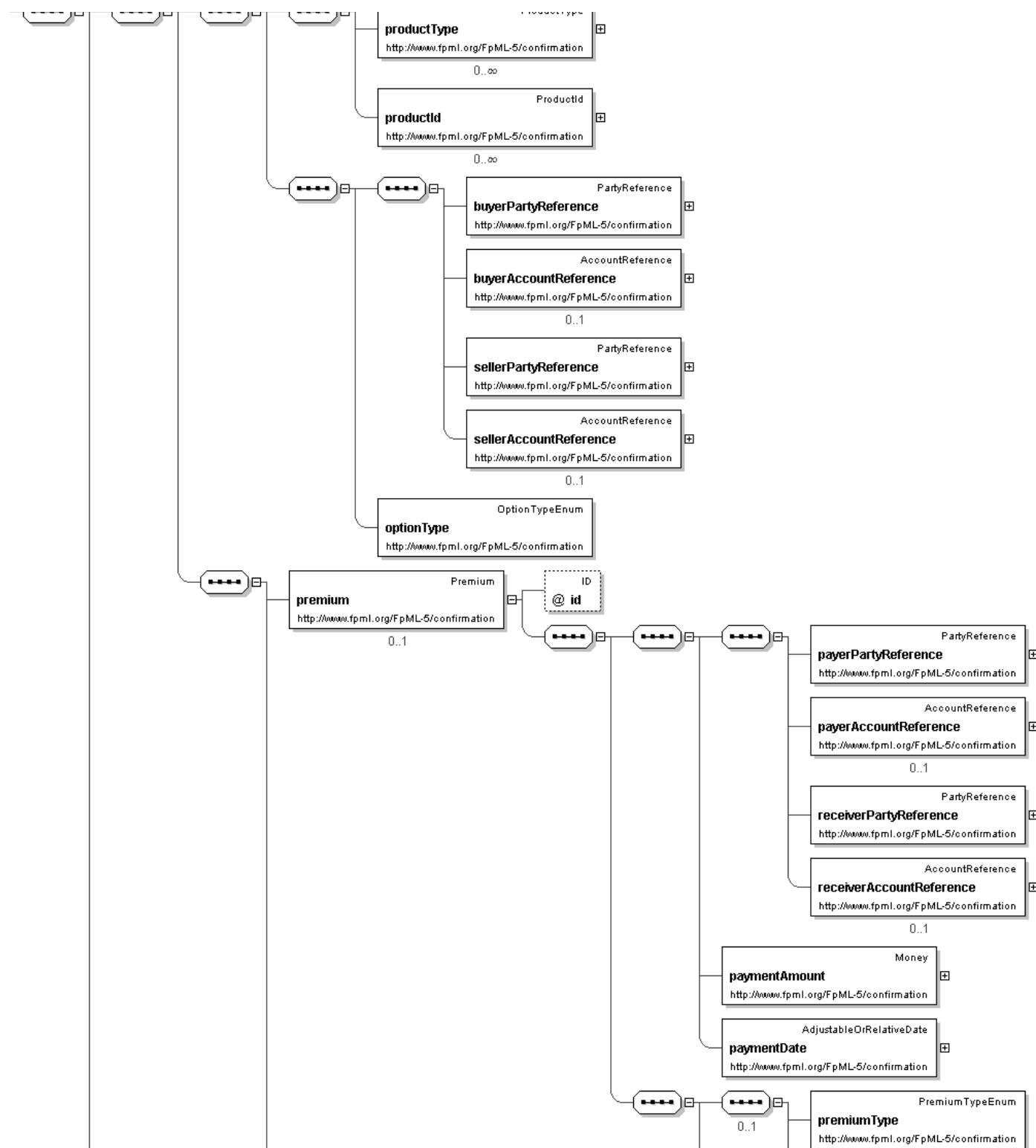
top

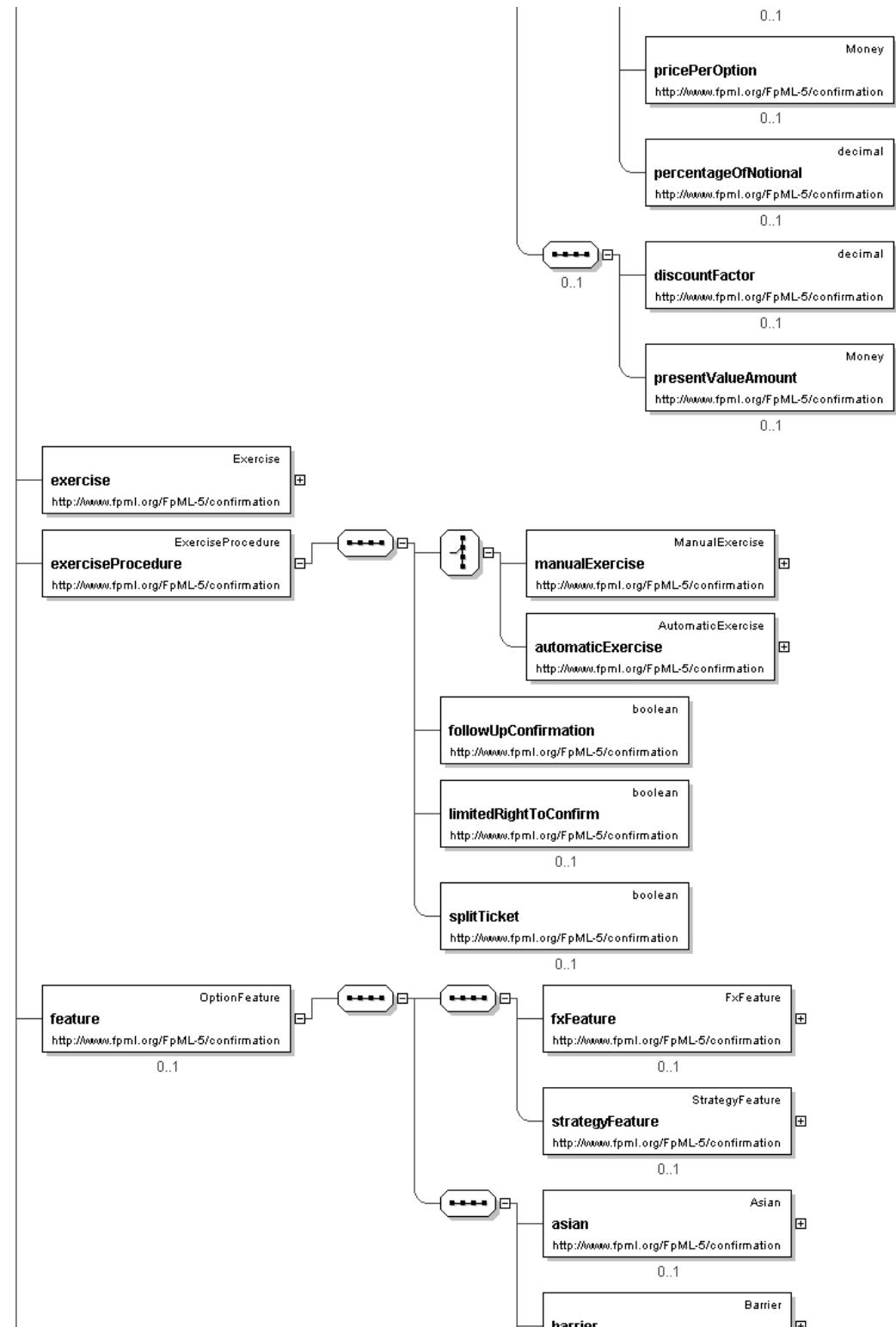
**Element: creditDefaultSwapOption**

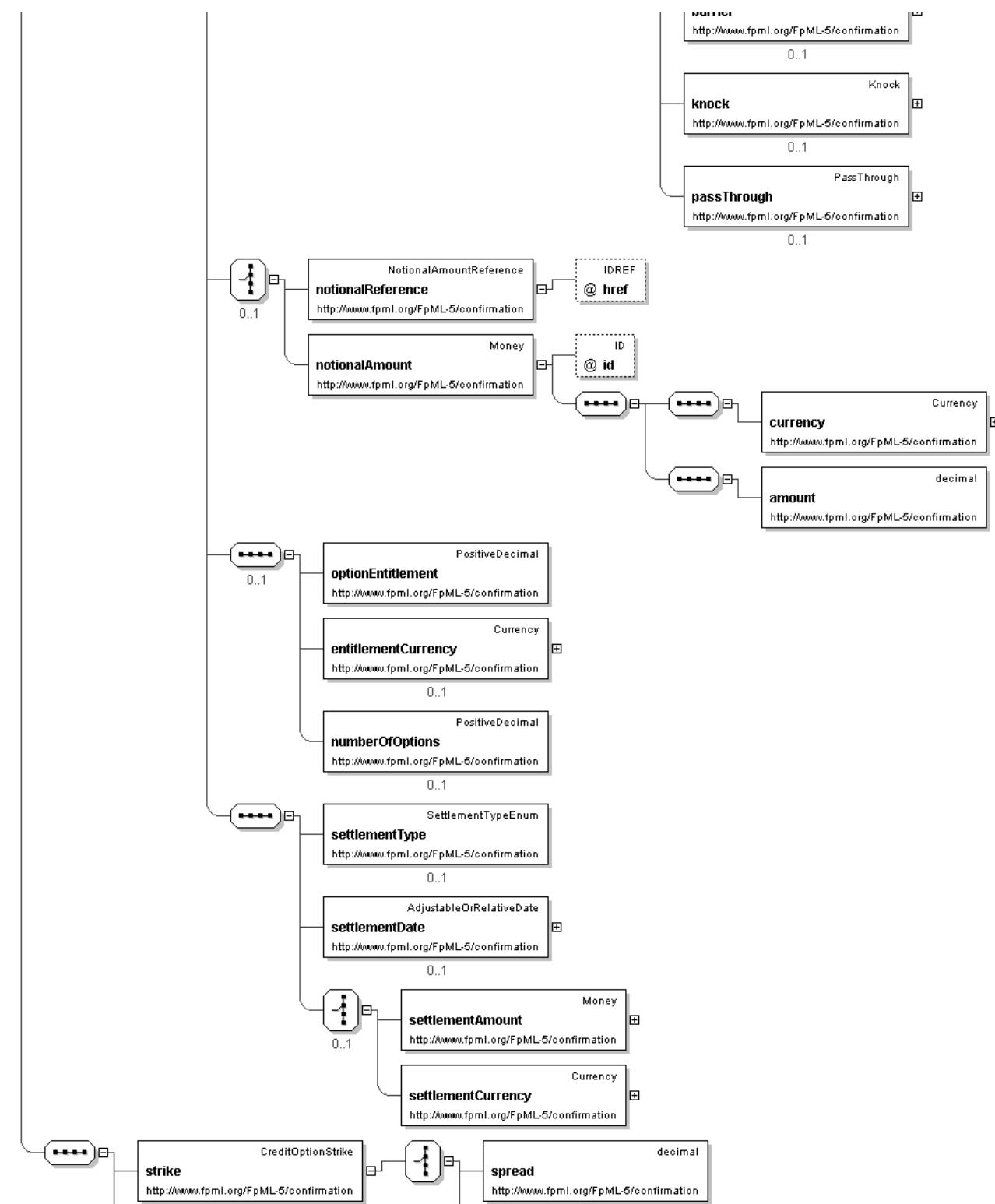
- This element can be used wherever the following element is referenced:
  - `product`

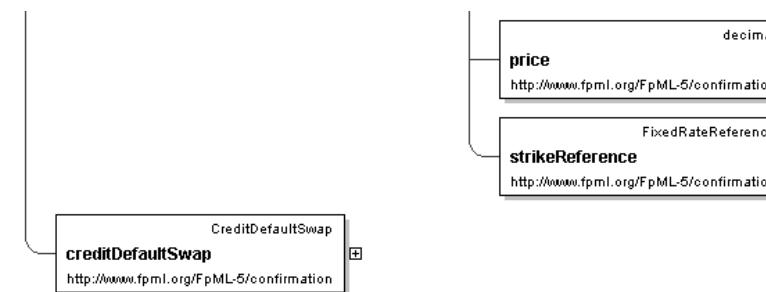
<b>Name</b>	creditDefaultSwapOption
<b>Type</b>	<a href="#">CreditDefaultSwapOption</a>
<b>Nullable</b>	no
<b>Abstract</b>	no
<b>Documentation</b>	An option on a credit default swap.

**Logical Diagram**







**XML Instance Representation**

```

<creditDefaultSwapOption
  id=" xsd:ID [0..1]">
  <productType> ProductType </productType> [0..*]
  'A classification of the type of product. FpML defines a simple product categorization using
  a coding scheme.'

  <productId> ProductId </productId> [0..*]
  'A product 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.'

  <buyerPartyReference> PartyReference </buyerPartyReference> [1]
  'A reference to the party that buys this instrument, i.e. pays for this instrument and
  receives the rights defined by it. See 2000 ISDA definitions Article 11.1 (b). In the case
  of FRAs this is the fixed rate payer.'

  <buyerAccountReference> AccountReference </buyerAccountReference> [0..1]
  'A reference to the account that buys this instrument.'

  <sellerPartyReference> PartyReference </sellerPartyReference> [1]
  'A reference to the party that sells ("writes") this instrument, i.e. that grants the
  rights defined by this instrument and in return receives a payment for it. See 2000
  ISDA definitions Article 11.1 (a). In the case of FRAs this is the floating rate payer.'

  <sellerAccountReference> AccountReference </sellerAccountReference> [0..1]
  'A reference to the account that sells this instrument.'

  <optionType> OptionTypeEnum </optionType> [1]
  'The type of option transaction. From a usage standpoint, put/call is the default option
  type, while payer/receiver indicator is used for options index credit default
  swaps, consistently with the industry practice. Straddle is used for the case of
  straddle strategy, that combine a call and a put with the same strike.'

  <premium> Premium </premium> [0..1]
  'The option premium payable by the buyer to the seller.'

  <exercise> ... </exercise> [1]
  <exerciseProcedure> ExerciseProcedure </exerciseProcedure> [1]
  'A set of parameters defining procedures associated with the exercise.'

  <feature> OptionFeature </feature> [0..1]
  'An Option feature such as quanto, asian, barrier, knock.'

Start Choice [0..1]
'A choice between an explicit representation of the notional amount, or a reference to
a notional amount defined elsewhere in this document.'

  <notionalReference> NotionalAmountReference </notionalReference> [1]

```

```

<notionalAmount> Money </notionalAmount> [1]
End Choice
Start Group: OptionDenomination.model [0..1]
<optionEntitlement> PositiveDecimal </optionEntitlement> [1]
'The number of units of underlyer per option comprised in the option transaction.'

<entitlementCurrency> Currency </entitlementCurrency> [0..1]
'TODO'

<numberOfOptions> PositiveDecimal </numberOfOptions> [0..1]
'The number of options comprised in the option transaction.'

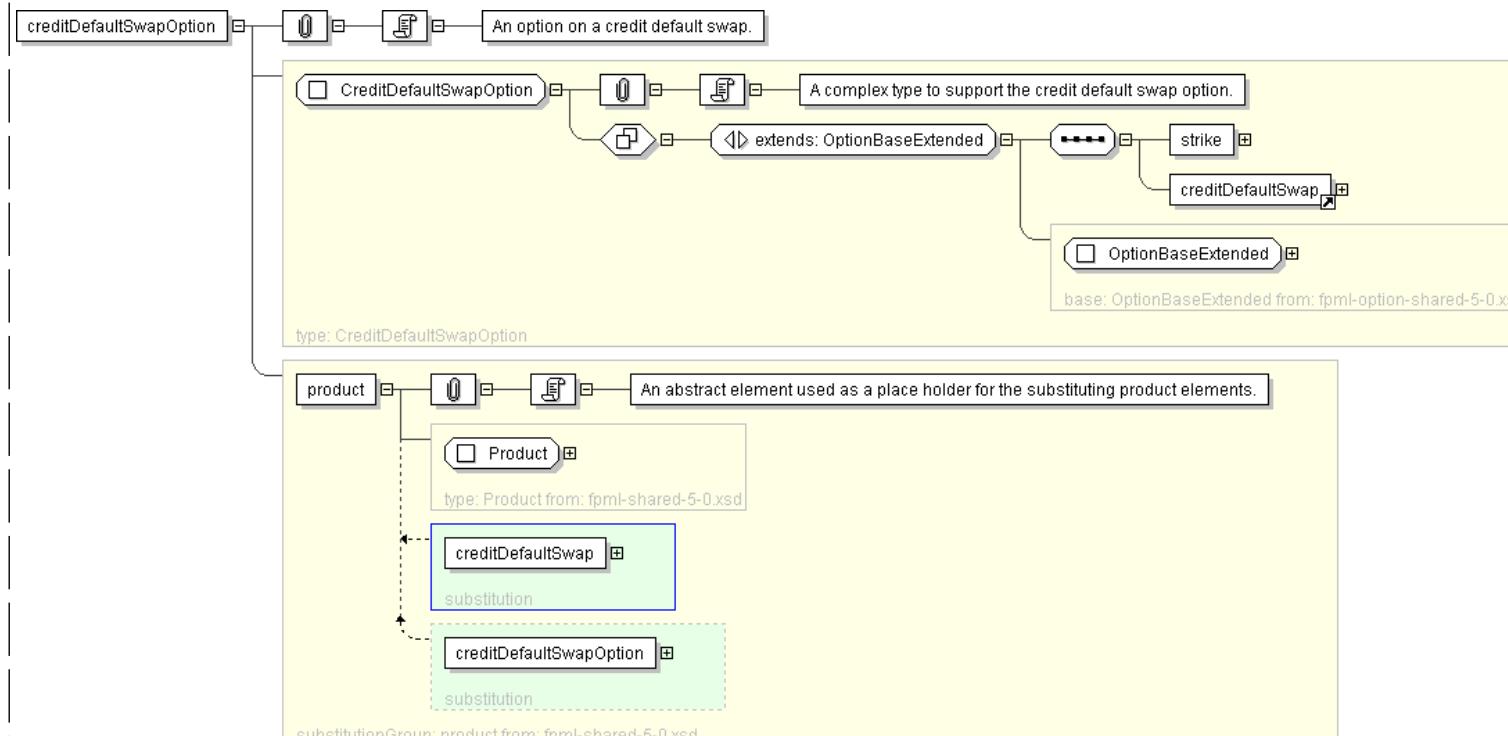
End Group: OptionDenomination.model
<settlementType> SettlementTypeEnum </settlementType> [0..1]
<settlementDate> AdjustableOrRelativeDate </settlementDate> [0..1]
Start Group: SettlementAmountOrCurrency.model [0..1]
Start Choice [1]
    <settlementAmount> Money </settlementAmount> [1]
    'Settlement Amount'

    <settlementCurrency> Currency </settlementCurrency> [1]
    'Settlement Currency for use where the Settlement Amount cannot be known in advance'

End Choice
End Group: SettlementAmountOrCurrency.model
<strike> CreditOptionStrike </strike> [1]
'Specifies the strike of the option on credit default swap.'

<creditDefaultSwap> ... </creditDefaultSwap> [1]
</creditDefaultSwapOption>

```

**Diagram**

**Schema Component Representation**

```
<xsd:element name="creditDefaultSwapOption" type="CreditDefaultSwapOption"
  "substitutionGroup="product"/>
```

**Global Definitions****Complex Type: AdditionalFixedPayments**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None

<b>Name</b>	AdditionalFixedPayments
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">FloatingAmountEvents</a>
<b>Abstract</b>	no

**XML Instance Representation**

```
<...>
<interestShortfallReimbursement> xsd:boolean </interestShortfallReimbursement> [0..1]
'An additional Fixed Payment Event. Corresponds to the payment by or on behalf of the Issuer
of an actual interest amount in respect to the reference obligation that is greater than
the expected interest amount. ISDA 2003 Term: Interest Shortfall Reimbursement.'
```

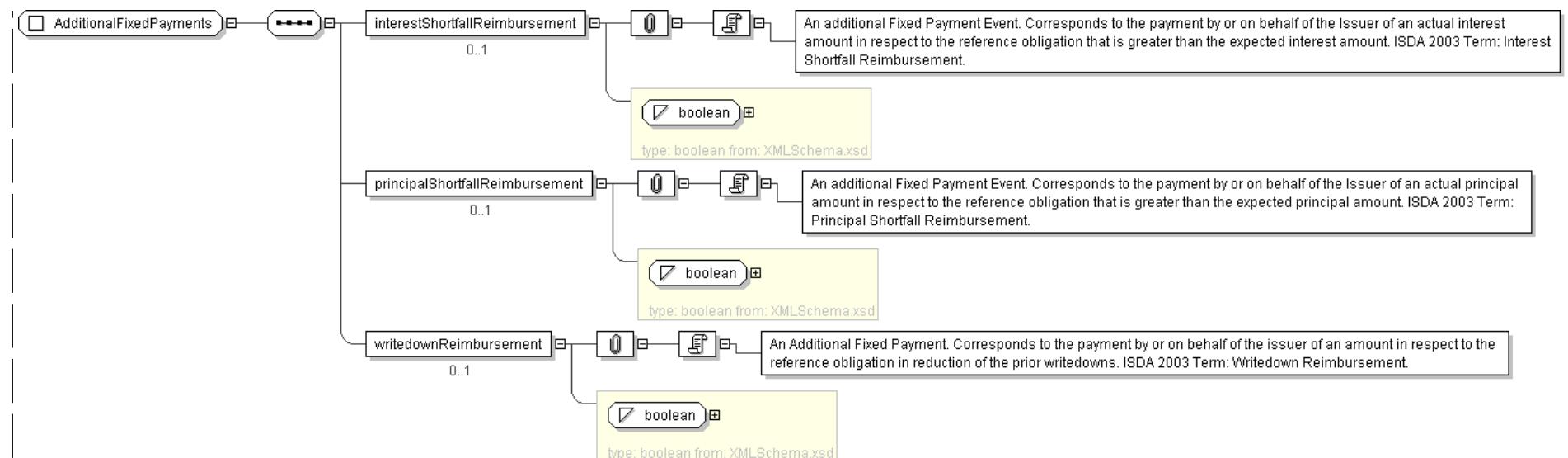
  

```
<principalShortfallReimbursement> xsd:boolean </principalShortfallReimbursement> [0..1]
'An additional Fixed Payment Event. Corresponds to the payment by or on behalf of the Issuer
of an actual principal amount in respect to the reference obligation that is greater than
the expected principal amount. ISDA 2003 Term: Principal Shortfall Reimbursement.'
```

```
<writedownReimbursement> xsd:boolean </writedownReimbursement> [0..1]
'An Additional Fixed Payment. Corresponds to the payment by or on behalf of the issuer of
an amount in respect to the reference obligation in reduction of the prior writedowns.
ISDA 2003 Term: Writedown Reimbursement.'
```

**Diagram**

**Schema Component Representation**

```
<xsd:complexType name="AdditionalFixedPayments">
  <xsd:sequence>
    <xsd:element name="interestShortfallReimbursement" type="xsd:boolean" minOccurs="0"/>
    <xsd:element name="principalShortfallReimbursement" type="xsd:boolean" minOccurs="0"/>
    <xsd:element name="writedownReimbursement" type="xsd:boolean" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>
```

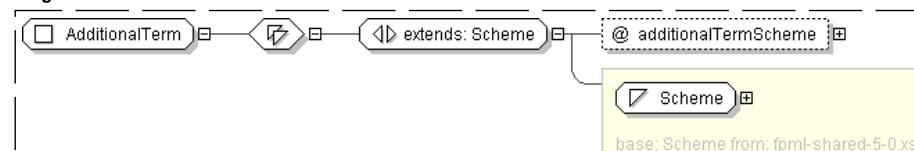
[top](#)**Complex Type: AdditionalTerm**

<b>Super-types:</b>	<a href="#">Scheme</a> < <b>AdditionalTerm</b> (by extension)
<b>Sub-types:</b>	None

<b>Name</b>	AdditionalTerm
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">GeneralTerms</a>
<b>Abstract</b>	no

**XML Instance Representation**

```
<...
  additionalTermScheme="xsd:anyURI [0..1]">
  Scheme
</...>
```

**Diagram****Schema Component Representation**

```
<xsd:complexType name="AdditionalTerm">
```

```

<xsd:simpleContent>
  <xsd:extension base=" Scheme ">
    <xsd:attribute name="additionalTermScheme" type=" xsd:anyURI " />
  </xsd:extension>
</xsd:simpleContent>
</xsd:complexType>

```

[top](#)

## Complex Type: **AdjustedPaymentDates**

**Super-types:** None  
**Sub-types:** None

<b>Name</b>	AdjustedPaymentDates
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">PeriodicPayment</a>
<b>Abstract</b>	no

### XML Instance Representation

```

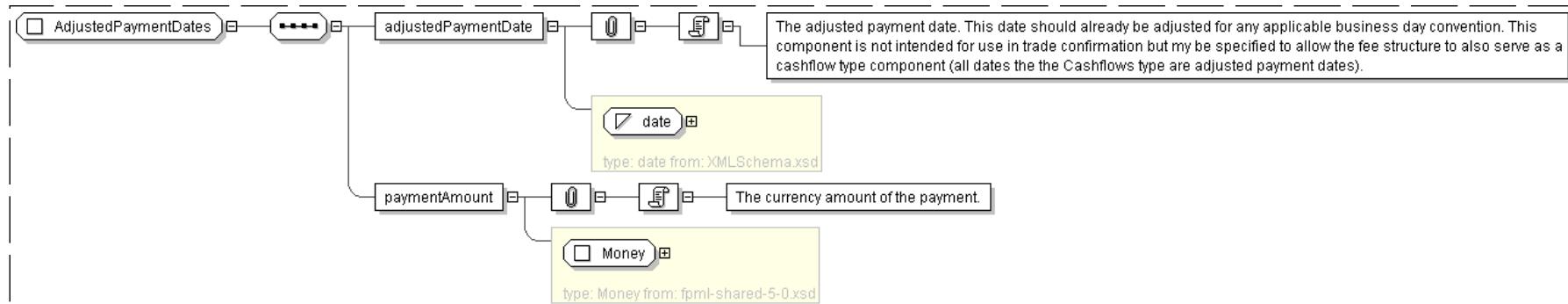
<...>
<adjustedPaymentDate> xsd:date </adjustedPaymentDate> [1]
'The adjusted payment date. This date should already be adjusted for any applicable
business day convention. This component is not intended for use in trade confirmation but may
be specified to allow the fee structure to also serve as a cashflow type component (all
dates in the Cashflows type are adjusted payment dates).'

<paymentAmount> Money </paymentAmount> [1]
'The currency amount of the payment.'

</...>

```

### Diagram



### Schema Component Representation

```

<xsd:complexType name="AdjustedPaymentDates">
  <xsd:sequence>
    <xsd:element name="adjustedPaymentDate" type=" xsd:date " />
    <xsd:element name="paymentAmount" type=" Money " />
  </xsd:sequence>
</xsd:complexType>

```

[top](#)

## Complex Type: **BasketReferenceInformation**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None
<b>Name</b>	BasketReferenceInformation
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">GeneralTerms</a>
<b>Abstract</b>	no
<b>Documentation</b>	CDS Basket Reference Information

**XML Instance Representation**

```

<...>
Start Group: BasketIdentifier.model [0..1]
'Reuses the group that specifies a name and an identifier for a given basket.'

Start Choice [1]
  <basketName> BasketName </basketName> [1]
    'The name of the basket expressed as a free format string. FpML does not define usage rules
    for this element.'

  <basketId> BasketId </basketId> [0..*]
    'A CDS basket identifier'

  <basketId> BasketId </basketId> [1..*]
    'A CDS basket identifier'

End Choice
End Group: BasketIdentifier.model
<referencePool> ReferencePool </referencePool> [1]
'This element contains all the reference pool items to define the reference entity
and reference obligation(s) in the basket'

Start Choice [0..1]
  <nthToDefault> xsd:positiveInteger </nthToDefault> [1]
    'N th reference obligation to default triggers payout.'

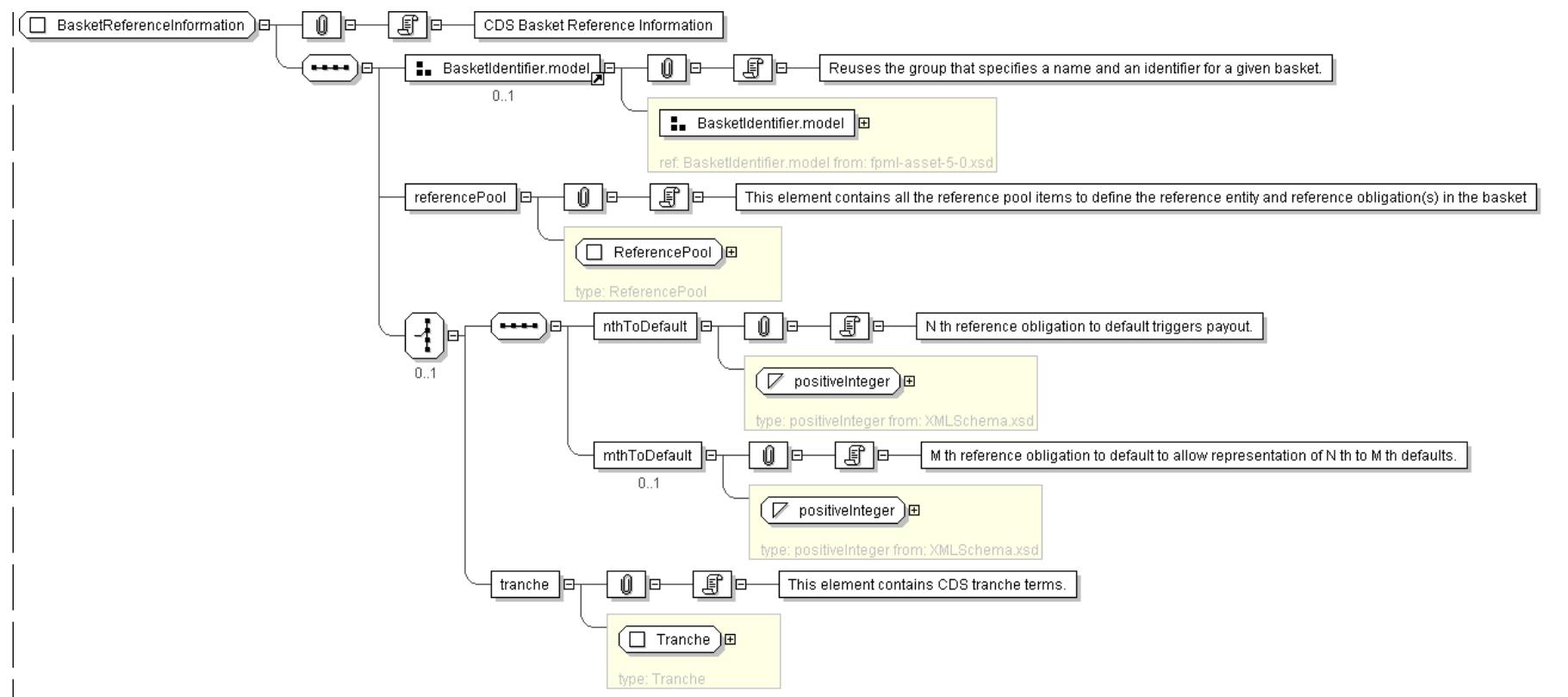
  <mthToDefault> xsd:positiveInteger </mthToDefault> [0..1]
    'M th reference obligation to default to allow representation of N th to M th defaults.'

  <tranche> Tranche </tranche> [1]
    'This element contains CDS tranche terms.'

End Choice
</...>

```

**Diagram**



#### Schema Component Representation

```

<xsd:complexType name="BasketReferenceInformation">
  <xsd:sequence>
    <xsd:group ref=" BasketIdentifier.model " minOccurs="0"/>
    <xsd:element name="referencePool" type=" ReferencePool "/>
    <xsd:choice minOccurs="0">
      <xsd:sequence>
        <xsd:element name="nthToDefault" type=" xsd:positiveInteger "/>
        <xsd:element name="mthToDefault" type=" xsd:positiveInteger " minOccurs="0"/>
      </xsd:sequence>
      <xsd:element name="tranche" type=" Tranche "/>
    </xsd:choice>
  </xsd:sequence>
</xsd:complexType>

```

top

#### Complex Type: CalculationAmount

Super-types:

`Money` < `CalculationAmount` (by extension)

Sub-types:

None

Name

CalculationAmount

Used by (from the same schema document)

Complex Type `FixedAmountCalculation`

Abstract

no

#### XML Instance Representation

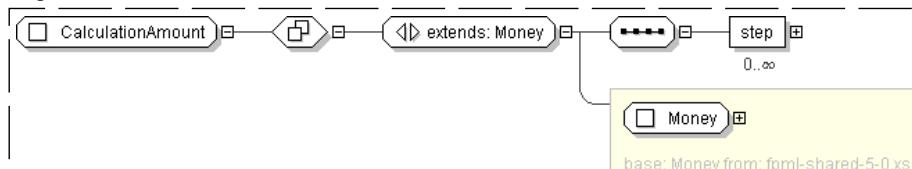
```

<...
  id="xsd:ID [0..1]">
  <currency> Currency </currency> [1]
  'The currency in which an amount is denominated.'

  <amount> xsd:decimal </amount> [1]
  'The monetary quantity in currency units.'

  <step> Step </step> [0..*]
  'A 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.'

</...>
```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="CalculationAmount">
  <xsd:complexContent>
    <xsd:extension base=" Money ">
      <xsd:sequence>
        <xsd:element name="step" type=" Step " minOccurs="0" maxOccurs="unbounded" />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

top

**Complex Type: CashSettlementTerms**

Super-types:	<a href="#">SettlementTerms</a> < <b>CashSettlementTerms</b> (by extension)
Sub-types:	None

Name	CashSettlementTerms
Used by (from the same schema document)	Complex Type <a href="#">CreditDefaultSwap</a>
Abstract	no

**XML Instance Representation**

```

<...
  id="xsd:ID [0..1]">
  <settlementCurrency> Currency </settlementCurrency> [0..1]
  'ISDA 2003 Term: Settlement Currency'

  <valuationDate> ValuationDate </valuationDate> [0..1]
  'The number of business days after conditions to settlement have been satisfied when
  the calculation agent obtains a price quotation on the Reference Obligation for purposes
  of cash settlement. There may be one or more valuation dates. This is typically specified
  if the cash settlement amount is not a fixed amount. ISDA 2003 Term: Valuation Date'

  <valuationTime> BusinessCenterTime </valuationTime> [0..1]
  'The time of day in the specified business center when the calculation agent seeks
```

`<quotations for an amount of the reference obligation for purposes of cash settlement. ISDA 2003 Term: Valuation Time'>`

`<quotationMethod> QuotationRateTypeEnum </quotationMethod> [0..1]`

*'The type of price quotations to be requested from dealers when determining the market value of the reference obligation for purposes of cash settlement. For example, Bid, Offer or Mid-market. ISDA 2003 Term: Quotation Method'*

`<quotationAmount> Money </quotationAmount> [0..1]`

*'In the determination of a cash settlement amount, if weighted average quotations are to be obtained, the quotation amount specifies an upper limit to the outstanding principal balance of the reference obligation for which the quote should be obtained. If not specified, the ISDA definitions provide for a fallback amount equal to the floating rate payer calculation amount. ISDA 2003 Term: Quotation Amount'*

`<minimumQuotationAmount> Money </minimumQuotationAmount> [0..1]`

*'In the determination of a cash settlement amount, if weighted average quotations are to be obtained, the minimum quotation amount specifies a minimum intended threshold amount of outstanding principal balance of the reference obligation for which the quote should be obtained. If not specified, the ISDA definitions provide for a fallback amount of the lower of either USD 1,000,000 (or its equivalent in the relevant obligation currency) or the quotation amount. ISDA 2003 Term: Minimum Quotation Amount'*

`<dealer> xsd:string </dealer> [0..*]`

*'A dealer from whom quotations are obtained by the calculation agent on the reference obligation for purposes of cash settlement. ISDA 2003 Term: Dealer'*

`<cashSettlementBusinessDays> xsd:nonNegativeInteger </cashSettlementBusinessDays> [0..1]`

*'The number of business days used in the determination of the cash settlement payment date. If a cash settlement amount is specified, the cash settlement payment date will be this number of business days following the calculation of the final price. If a cash settlement amount is not specified, the cash settlement payment date will be this number of business days after all conditions to settlement are satisfied. ISDA 2003 Term: Cash Settlement Date'*

Start Group: FixedRecovery.model [0..1]

Start Choice [1]

`<cashSettlementAmount> Money </cashSettlementAmount> [1]`

*'The amount paid by the seller to the buyer for cash settlement on the cash settlement date. If not otherwise specified, would typically be calculated as 100 (or the Reference Price) minus the price of the Reference Obligation (all expressed as a percentage) times Floating Rate Payer Calculation Amount. ISDA 2003 Term: Cash Settlement Amount.'*

`<recoveryFactor> RestrictedPercentage </recoveryFactor> [1]`

*'Used for fixed recovery, specifies the recovery level, determined at contract inception, to be applied on a default. Used to calculate the amount paid by the seller to the buyer for cash settlement on the cash settlement date. Amount calculation is (1 minus the Recovery Factor) multiplied by the Floating Rate Payer Calculation Amount. The currency will be derived from the Floating Rate Payer Calculation Amount.'*

End Choice

End Group: FixedRecovery.model

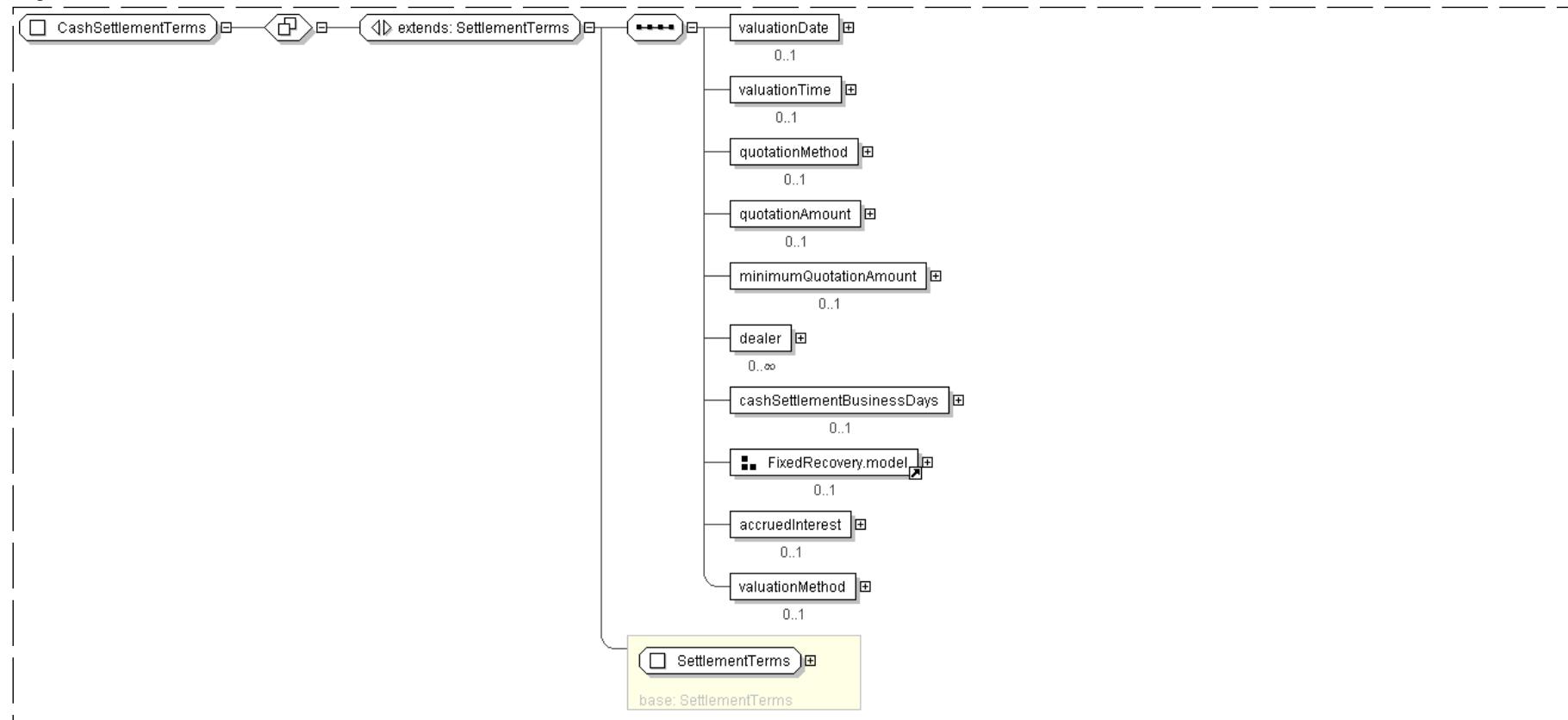
`<accruedInterest> xsd:boolean </accruedInterest> [0..1]`

*'Indicates whether accrued interest is included (true) or not (false). For cash settlement this specifies whether quotations should be obtained inclusive or not of accrued interest. For physical settlement this specifies whether the buyer should deliver the obligation with an outstanding principal balance that includes or excludes accrued interest. ISDA 2003 Term: Include/Exclude Accrued Interest'*

`<valuationMethod> ValuationMethodEnum </valuationMethod> [0..1]`

*'The ISDA defined methodology for determining the final price of the reference obligation for purposes of cash settlement. (ISDA 2003 Term: Valuation Method). For example, Market, Highest etc.'*

`</...>`

**Diagram****Schema Component Representation**

```

<xsd:complexType name="CashSettlementTerms">
  <xsd:complexContent>
    <xsd:extension base=" SettlementTerms ">
      <xsd:sequence>
        <xsd:element name="valuationDate" type=" ValuationDate " minOccurs="0"/>
        <xsd:element name="valuationTime" type=" BusinessCenterTime " minOccurs="0"/>
        <xsd:element name="quotationMethod" type=" QuotationRateTypeEnum " minOccurs="0"/>
        <xsd:element name="quotationAmount" type=" Money " minOccurs="0"/>
        <xsd:element name="minimumQuotationAmount" type=" Money " minOccurs="0"/>
        <xsd:element name="dealer" type=" xsd:string " minOccurs="0" maxOccurs="unbounded"/>
        <xsd:element name="cashSettlementBusinessDays" type=" xsd:nonNegativeInteger " minOccurs="0"/>
        <xsd:group ref=" FixedRecovery.model " minOccurs="0"/>
        <xsd:element name="accruedInterest" type=" xsd:boolean " minOccurs="0"/>
        <xsd:element name="valuationMethod" type=" ValuationMethodEnum " minOccurs="0"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

[top](#)**Complex Type: CreditDefaultSwap**

Super-types:

[Product](#) < CreditDefaultSwap (by extension)

Sub-types:

None

<b>Name</b>	CreditDefaultSwap
<b>Used by (from the same schema document)</b>	Element <a href="#">creditDefaultSwap</a>
<b>Abstract</b>	no

**XML Instance Representation**

```

<...
  id=" xsd:ID [0..1]">
  <productType> ProductType </productType> [0..*]
  'A classification of the type of product. FpML defines a simple product categorization using
  a coding scheme.'

  <productId> ProductId </productId> [0..*]
  'A product 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.'

  <generalTerms> GeneralTerms </generalTerms> [1]
  'This element contains all the data that appears in the section entitled \"1. General Terms
  \\" in the 2003 ISDA Credit Derivatives Confirmation.'

  <feeLeg> FeeLeg </feeLeg> [1]
  'This element contains all the terms relevant to defining the fixed amounts/payments per
  the applicable ISDA definitions.'

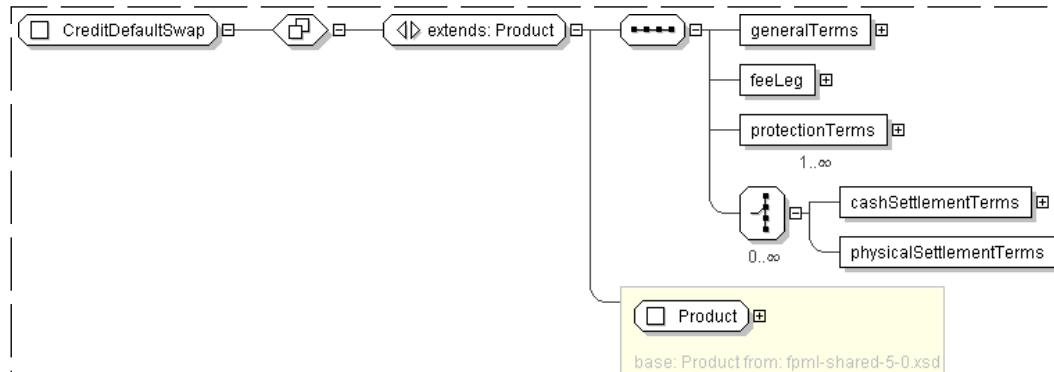
  <protectionTerms> ProtectionTerms </protectionTerms> [1..*]
  'This element contains all the terms relevant to defining the applicable floating rate
  payer calculation amount, credit events and associated conditions to settlement, and
  reference obligations.'

Start Choice [0..*]
  <cashSettlementTerms> CashSettlementTerms </cashSettlementTerms> [1]
  'This element contains all the ISDA terms relevant to cash settlement for when cash
  settlement is applicable. ISDA 2003 Term: Cash Settlement'

  <physicalSettlementTerms> PhysicalSettlementTerms </physicalSettlementTerms> [1]
  'This element contains all the ISDA terms relevant to physical settlement for when
  physical settlement is applicable. ISDA 2003 Term: Physical Settlement'

End Choice
</...>

```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="CreditDefaultSwap">
  <xsd:complexContent>

```

```

<xsd:extension base=" Product " >
  <xsd:sequence>
    <xsd:element name="generalTerms" type=" GeneralTerms " />
    <xsd:element name="feeLeg" type=" FeeLeg " />
    <xsd:element name="protectionTerms" type=" ProtectionTerms " maxOccurs="unbounded" />
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
      <xsd:element name="cashSettlementTerms" type=" CashSettlementTerms " />
      <xsd:element name="physicalSettlementTerms" type=" PhysicalSettlementTerms " />
    </xsd:choice>
  </xsd:sequence>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>

```

## Complex Type: [CreditDefaultSwapOption](#)

<b>Super-types:</b>	<a href="#">OptionBaseExtended</a> < <b>CreditDefaultSwapOption</b> (by extension)
<b>Sub-types:</b>	None

<b>Name</b>	CreditDefaultSwapOption
<b>Used by (from the same schema document)</b>	Element <a href="#">creditDefaultSwapOption</a>
<b>Abstract</b>	no
<b>Documentation</b>	A complex type to support the credit default swap option.

### XML Instance Representation

```

<...
  id=" xsd:ID [0..1]">
  <productType> ProductType </productType> [0..*]
  'A classification of the type of product. FpML defines a simple product categorization using
  a coding scheme.'

  <productId> ProductId </productId> [0..*]
  'A product 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.'

  <buyerPartyReference> PartyReference </buyerPartyReference> [1]
  'A reference to the party that buys this instrument, i.e. pays for this instrument and
  receives the rights defined by it. See 2000 ISDA definitions Article 11.1 (b). In the case
  of FRAs this is the fixed rate payer.'

  <buyerAccountReference> AccountReference </buyerAccountReference> [0..1]
  'A reference to the account that buys this instrument.'

  <sellerPartyReference> PartyReference </sellerPartyReference> [1]
  'A reference to the party that sells ("writes") this instrument, i.e. that grants the
  rights defined by this instrument and in return receives a payment for it. See 2000
  ISDA definitions Article 11.1 (a). In the case of FRAs this is the floating rate payer.'

  <sellerAccountReference> AccountReference </sellerAccountReference> [0..1]
  'A reference to the account that sells this instrument.'

  <optionType> OptionTypeEnum </optionType> [1]
  'The type of option transaction. From a usage standpoint, put/call is the default option
  type, while payer/receiver indicator is used for options index credit default
  swaps, consistently with the industry practice. Straddle is used for the case of
  straddle strategy, that combine a call and a put with the same strike.'

  <premium> Premium </premium> [0..1]
  'The option premium payable by the buyer to the seller.'

```

```

<exercise> ... </exercise> [1]
<exerciseProcedure> ExerciseProcedure </exerciseProcedure> [1]
'A set of parameters defining procedures associated with the exercise.'

<feature> OptionFeature </feature> [0..1]
'An Option feature such as quanto, asian, barrier, knock.'

Start Choice [0..1]
'A choice between an explicit representation of the notional amount, or a reference to
a notional amount defined elsewhere in this document.'

<notionalReference> NotionalAmountReference </notionalReference> [1]
<notionalAmount> Money </notionalAmount> [1]
End Choice
Start Group: OptionDenomination.model [0..1]
<optionEntitlement> PositiveDecimal </optionEntitlement> [1]
'The number of underlyer per option comprised in the option transaction.'

<entitlementCurrency> Currency </entitlementCurrency> [0..1]
'TODO'

<numberOfOptions> PositiveDecimal </numberOfOptions> [0..1]
'The number of options comprised in the option transaction.'

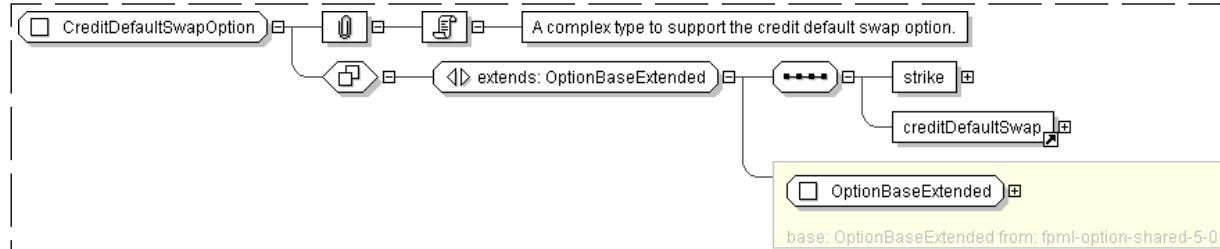
End Group: OptionDenomination.model
<settlementType> SettlementTypeEnum </settlementType> [0..1]
<settlementDate> AdjustableOrRelateDate </settlementDate> [0..1]
Start Group: SettlementAmountOrCurrency.model [0..1]
Start Choice [1]
<settlementAmount> Money </settlementAmount> [1]
'Settlement Amount'

<settlementCurrency> Currency </settlementCurrency> [1]
'Settlement Currency for use where the Settlement Amount cannot be known in advance'

End Choice
End Group: SettlementAmountOrCurrency.model
<strike> CreditOptionStrike </strike> [1]
'Specifies the strike of the option on credit default swap.'

<creditDefaultSwap> ... </creditDefaultSwap> [1]
</...>

```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="CreditDefaultSwapOption">
<xsd:complexContent>
  <xsd:extension base="OptionBaseExtended">
    <xsd:sequence>

```

```

<xsd:element name="strike" type=" CreditOptionStrike "/>
<xsd:element ref=" creditDefaultSwap " />
</xsd:sequence>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>

```

[top](#)

## Complex Type: CreditOptionStrike

Super-types:	None
Sub-types:	None

Name	CreditOptionStrike
Used by (from the same schema document)	Complex Type <a href="#">CreditDefaultSwapOption</a>
Abstract	no
Documentation	A complex type to specify the strike of a credit swaption or a credit default swap option.

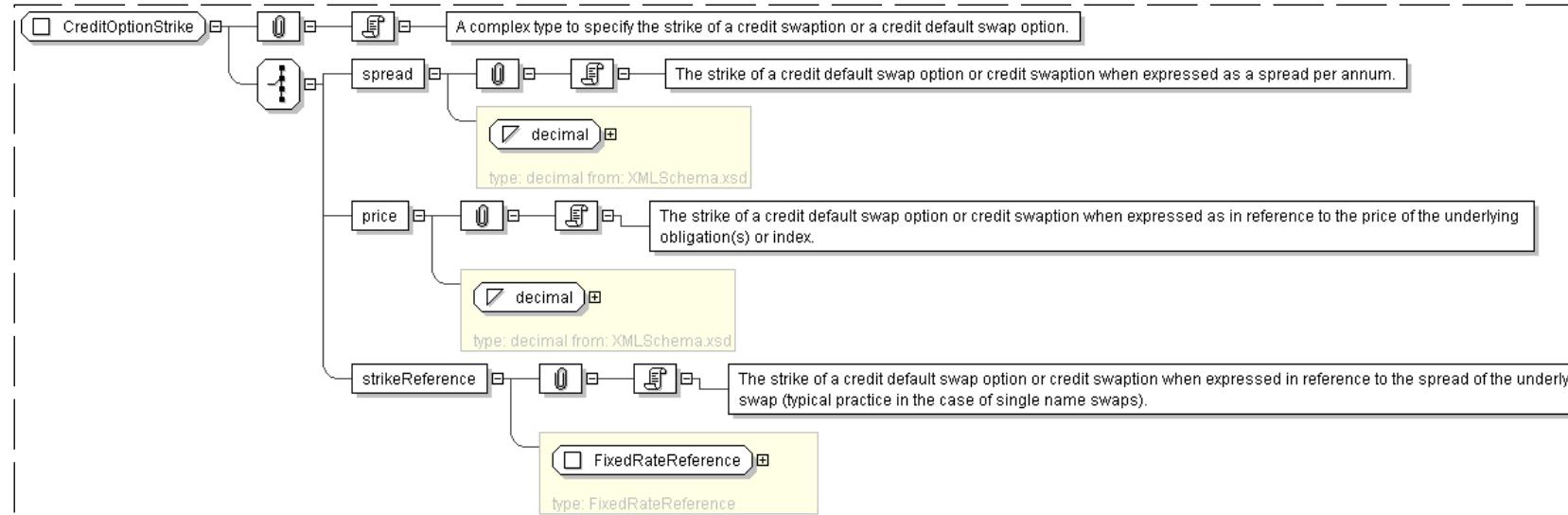
### XML Instance Representation

```

<...>
Start Choice [1]
  <spread> xsd:decimal </spread> [1]
    'The strike of a credit default swap option or credit swaption when expressed as a spread per annum.'
  <price> xsd:decimal </price> [1]
    'The strike of a credit default swap option or credit swaption when expressed as in reference to the price of the underlying obligation(s) or index.'
  <strikeReference> FixedRateReference </strikeReference> [1]
    'The strike of a credit default swap option or credit swaption when expressed in reference to the spread of the underlying swap (typical practice in the case of single name swaps).'
End Choice
</...>

```

### Diagram



**Schema Component Representation**

```
<xsd:complexType name="CreditOptionStrike">
  <xsd:choice>
    <xsd:element name="spread" type="xsd:decimal" />
    <xsd:element name="price" type="xsd:decimal" />
    <xsd:element name="strikeReference" type="FixedRateReference" />
  </xsd:choice>
</xsd:complexType>
```

[top](#)**Complex Type: DeliverableObligations**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None

<b>Name</b>	DeliverableObligations
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">PhysicalSettlementTerms</a>
<b>Abstract</b>	no

**XML Instance Representation**

```
<...>
<accruedInterest> xsd:boolean </accruedInterest> [0..1]
'Indicates whether accrued interest is included (true) or not (false). For cash settlement
this specifies whether quotations should be obtained inclusive or not of accrued interest.
For physical settlement this specifies whether the buyer should deliver the obligation with
an outstanding principal balance that includes or excludes accrued interest. ISDA 2003
Term: Include/Exclude Accrued Interest'
```

```
<category> ObligationCategoryEnum </category> [0..1]
'Used in both obligations and deliverable obligations to represent a class or type
of securities which apply. ISDA 2003 Term: Obligation Category/Deliverable Obligation Category'
```

```
<notSubordinated> xsd:boolean </notSubordinated> [0..1]
'An obligation and deliverable obligation characteristic. An obligation that ranks at
least equal with the most senior Reference Obligation in priority of payment or, if
no Reference Obligation is specified in the related Confirmation, the obligations of
the Reference Entity that are senior. ISDA 2003 Term: Not Subordinated'
```

```
<specifiedCurrency> SpecifiedCurrency </specifiedCurrency> [0..1]
'An obligation and deliverable obligation characteristic. The currency or currencies in
which an obligation or deliverable obligation must be payable. ISDA 2003 Term:
Specified Currency'
```

```
<notSovereignLender> xsd:boolean </notSovereignLender> [0..1]
'An obligation and deliverable obligation characteristic. Any obligation that is not
primarily (majority) owed to a Sovereign or Supranational Organization. ISDA 2003 Term:
Not Sovereign Lender'
```

```
<notDomesticCurrency> NotDomesticCurrency </notDomesticCurrency> [0..1]
'An obligation and deliverable obligation characteristic. Any obligation that is payable in
any currency other than the domestic currency. Domestic currency is either the currency
so specified or, if no currency is specified, the currency of (a) the reference entity, if
the reference entity is a sovereign, or (b) the jurisdiction in which the relevant
reference entity is organised, if the reference entity is not a sovereign. ISDA 2003 Term:
Not Domestic Currency'
```

```
<notDomesticLaw> xsd:boolean </notDomesticLaw> [0..1]
'An obligation and deliverable obligation characteristic. If the reference entity is
a Sovereign, this means any obligation that is not subject to the laws of the reference
entity. If the reference entity is not a sovereign, this means any obligation that is
not subject to the laws of the jurisdiction of the reference entity. ISDA 2003 Term:
Not Domestic Law'
```

<listed> xsd:boolean </listed> [0..1]

'An obligation and deliverable obligation characteristic. Indicates whether or not the obligation is quoted, listed or ordinarily purchased and sold on an exchange. ISDA 2003 Term: Listed'

<notContingent> xsd:boolean </notContingent> [0..1]

'A deliverable obligation characteristic. In essence Not Contingent means the repayment of principal cannot be dependant on a formula/index, i.e. to prevent the risk of being delivered an instrument that may never pay any element of principal, and to ensure that the obligation is interest bearing (on a regular schedule). ISDA 2003 Term: Not Contingent'

<notDomesticIssuance> xsd:boolean </notDomesticIssuance> [0..1]

'An obligation and deliverable obligation characteristic. Any obligation other than an obligation that was intended to be offered for sale primarily in the domestic market of the relevant Reference Entity. This specifies that the obligation must be an internationally recognized bond. ISDA 2003 Term: Not Domestic Issuance'

<assignableLoan> PCDeliverableObligationCharac </assignableLoan> [0..1]

'A deliverable obligation characteristic. A loan that is freely assignable to a bank or financial institution without the consent of the Reference Entity or the guarantor, if any, of the loan (or the consent of the applicable borrower if a Reference Entity is guaranteeing the loan) or any agent. ISDA 2003 Term: Assignable Loan'

<consentRequiredLoan> PCDeliverableObligationCharac </consentRequiredLoan> [0..1]

'A deliverable obligation characteristic. A loan that is capable of being assigned with the consent of the Reference Entity or the guarantor, if any, of the loan or any agent. ISDA 2003 Term: Consent Required Loan'

<directLoanParticipation> LoanParticipation </directLoanParticipation> [0..1]

'A deliverable obligation characteristic. A loan with a participation agreement whereby the buyer is capable of creating, or procuring the creation of, a contractual right in favour of the seller that provides the seller with recourse to the participation seller for a specified share in any payments due under the relevant loan which are received by the participation seller. ISDA 2003 Term: Direct Loan Participation'

<transferable> xsd:boolean </transferable> [0..1]

'A deliverable obligation characteristic. An obligation that is transferable to institutional investors without any contractual, statutory or regulatory restrictions. ISDA 2003 Term: Transferable'

<maximumMaturity> Period </maximumMaturity> [0..1]

'A deliverable obligation characteristic. An obligation that has a remaining maturity from the Physical Settlement Date of not greater than the period specified. ISDA 2003 Term: Maximum Maturity'

<acceleratedOrMatured> xsd:boolean </acceleratedOrMatured> [0..1]

'A deliverable obligation characteristic. An obligation at time of default is due to mature and due to be repaid, or as a result of downgrade/bankruptcy is due to be repaid as a result of an acceleration clause. ISDA 2003 Term: Accelerated or Matured'

<notBearer> xsd:boolean </notBearer> [0..1]

'A deliverable obligation characteristic. Any obligation that is not a bearer instrument. This applies to Bonds only and is meant to avoid tax, fraud and security/delivery provisions that can potentially be associated with Bearer Bonds. ISDA 2003 Term: Not Bearer'

Start Choice [0..1]

<fullFaithAndCreditObLiability> xsd:boolean </fullFaithAndCreditObLiability> [1]

'An obligation and deliverable obligation characteristic. Defined in the ISDA published additional provisions for U.S. Municipal as Reference Entity. ISDA 2003 Term: Full Faith and Credit Obligation Liability'

<generalFundObligationLiability> xsd:boolean </generalFundObligationLiability> [1]

'An obligation and deliverable obligation characteristic. Defined in the ISDA published additional provisions for U.S. Municipal as Reference Entity. ISDA 2003 Term: General Fund Obligation Liability'

<revenueObligationLiability> xsd:boolean </revenueObligationLiability> [1]  
 'An obligation and deliverable obligation characteristic. Defined in the ISDA published additional provisions for U.S. Municipal as Reference Entity. ISDA 2003 Term: Revenue Obligation Liability'

End Choice

<indirectLoanParticipation> LoanParticipation </indirectLoanParticipation> [0..1]  
 'ISDA 1999 Term: Indirect Loan Participation. NOTE: Only applicable as a deliverable obligation under ISDA Credit 1999.'

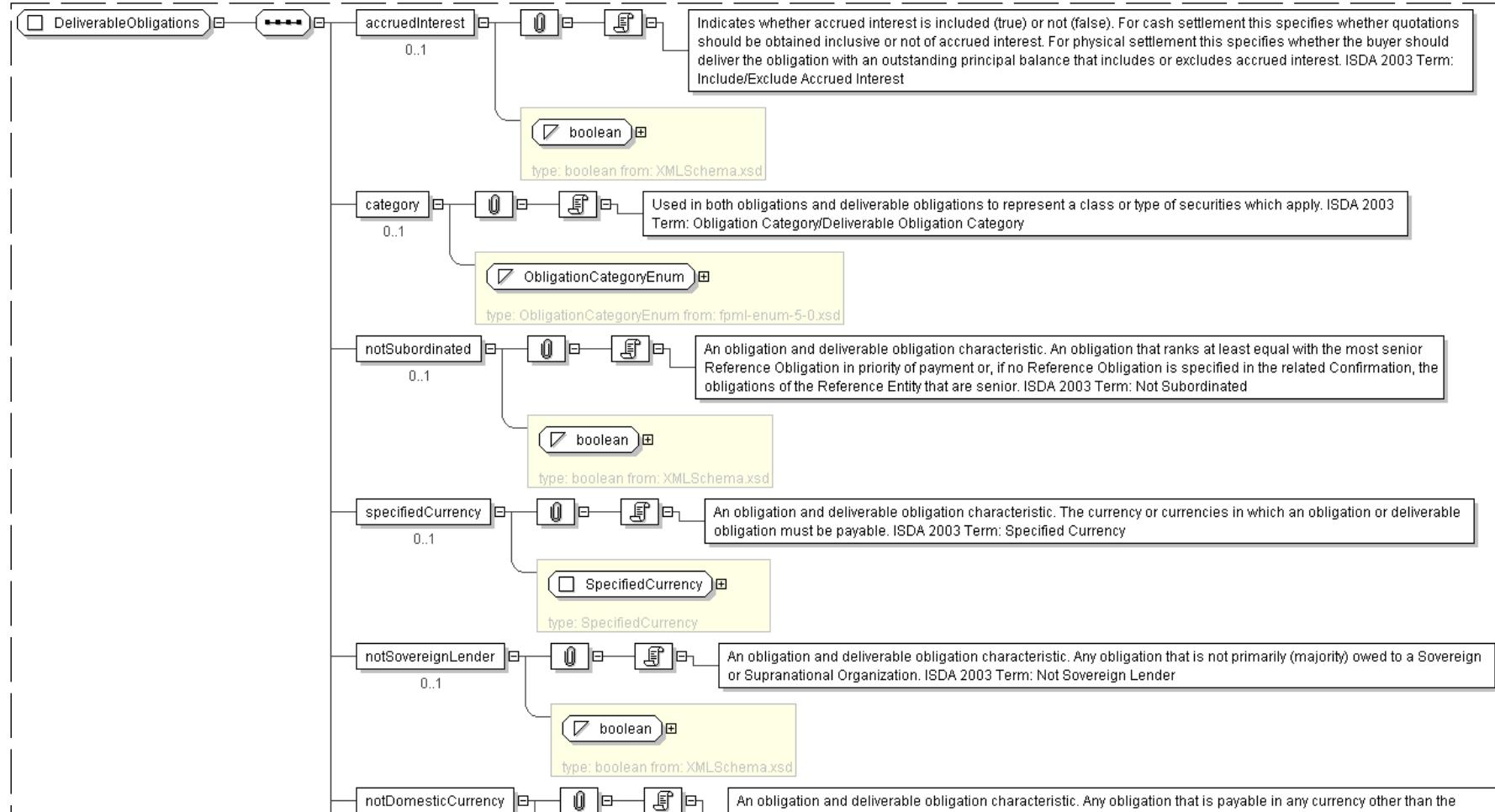
&lt;excluded&gt; xsd:string &lt;/excluded&gt; [0..1]

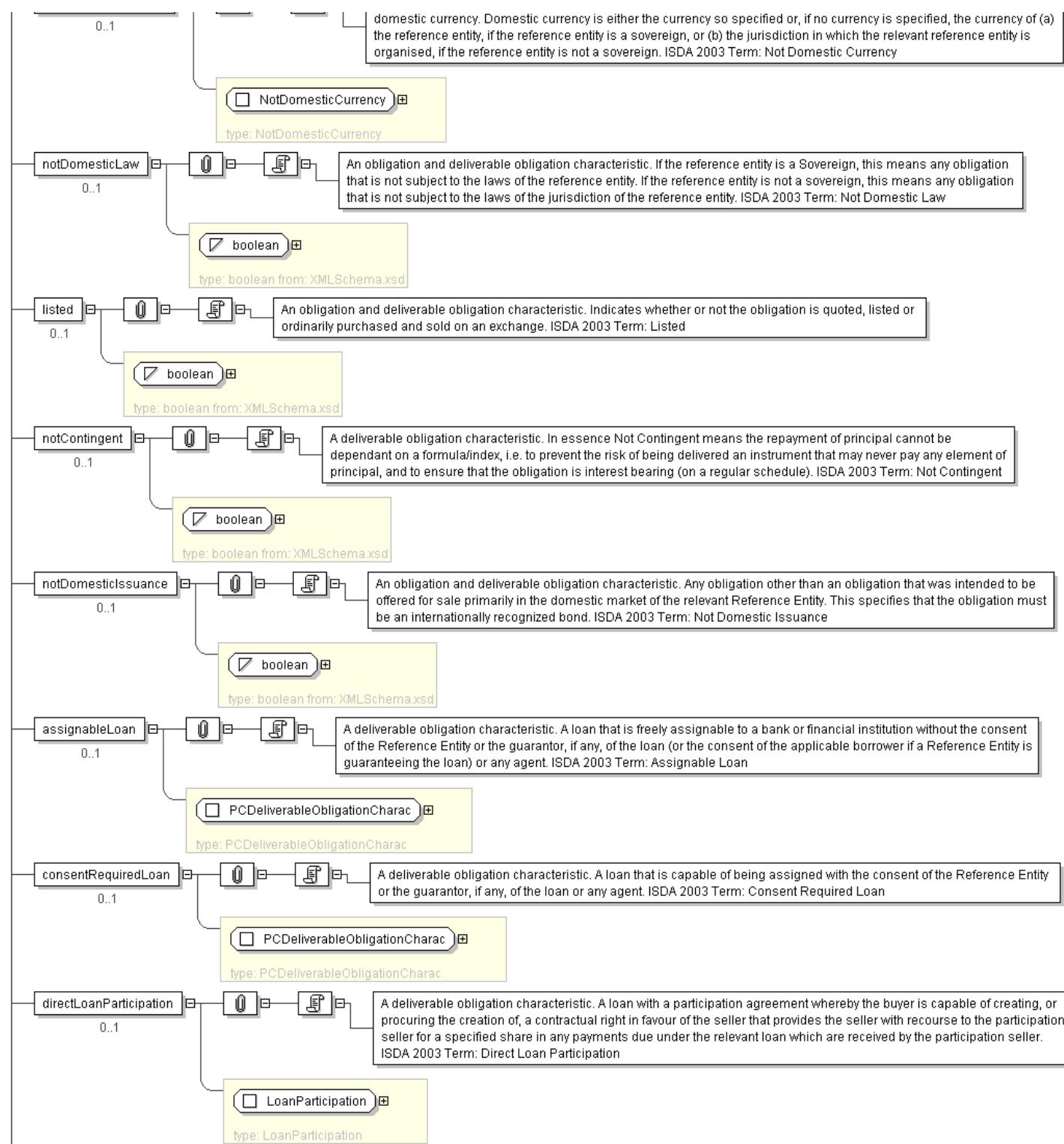
'A free format string to specify any excluded obligations or deliverable obligations, as the case may be, of the reference entity or excluded types of obligations or deliverable obligations. ISDA 2003 Term: Excluded Obligations/Excluded Deliverable Obligations'

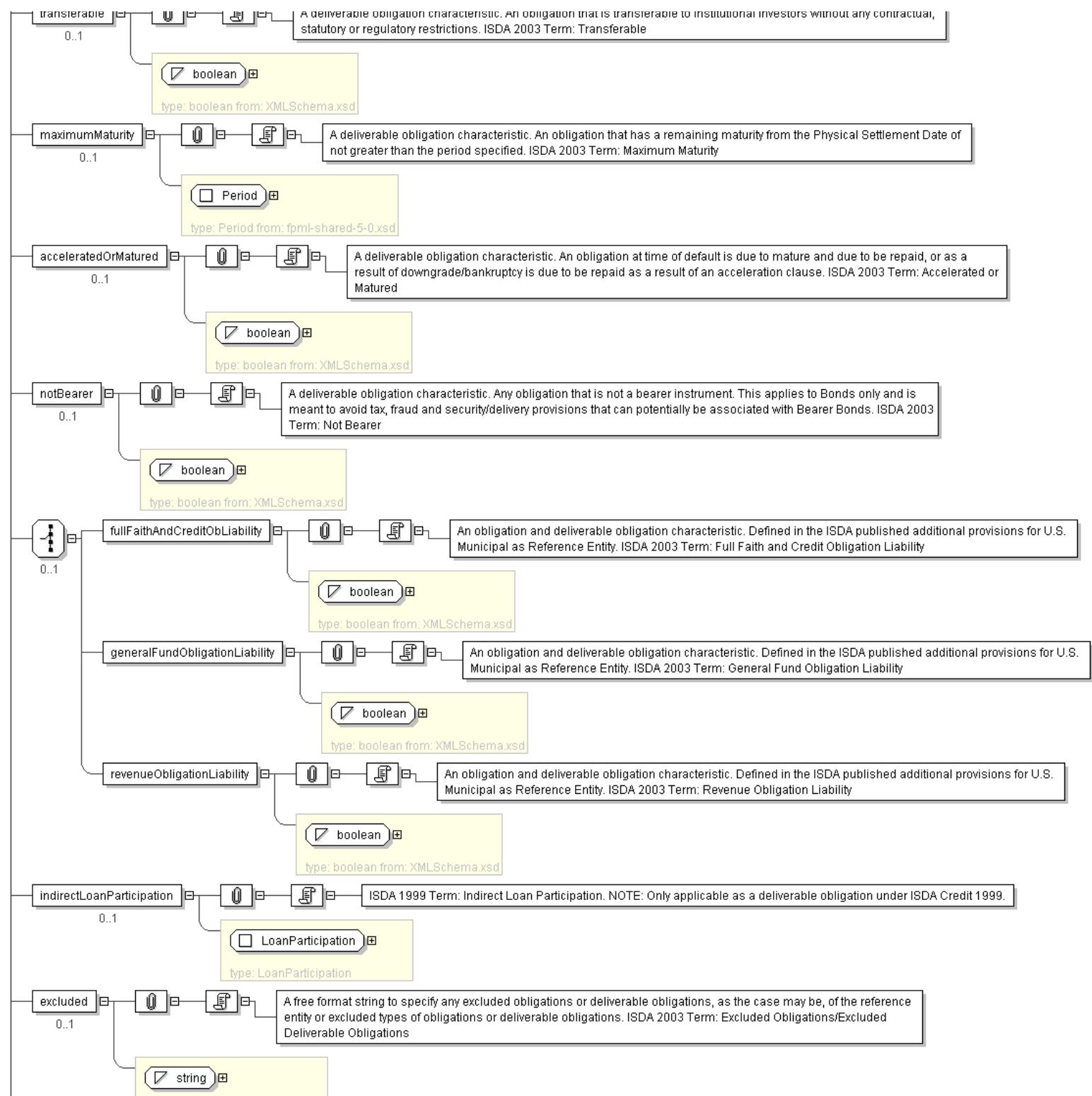
&lt;othReferenceEntityObligations&gt; xsd:string &lt;/othReferenceEntityObligations&gt; [0..1]

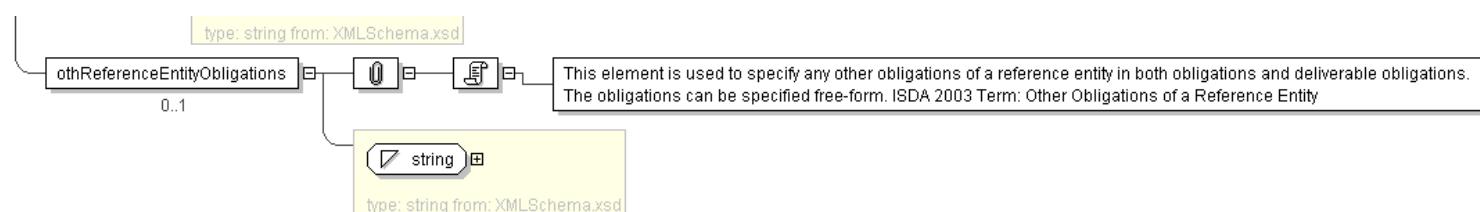
'This element is used to specify any other obligations of a reference entity in both obligations and deliverable obligations. The obligations can be specified free-form. ISDA 2003 Term: Other Obligations of a Reference Entity'

&lt;...&gt;

**Diagram**





**Schema Component Representation**

```

<xsd:complexType name="DeliverableObligations">
  <xsd:sequence>
    <xsd:element name="accruedInterest" type="xsd:boolean" minOccurs="0"/>
    <xsd:element name="category" type="ObligationCategoryEnum" minOccurs="0"/>
    <xsd:element name="notSubordinated" type="xsd:boolean" minOccurs="0"/>
    <xsd:element name="specifiedCurrency" type="SpecifiedCurrency" minOccurs="0"/>
    <xsd:element name="notSovereignLender" type="xsd:boolean" minOccurs="0"/>
    <xsd:element name="notDomesticCurrency" type="NotDomesticCurrency" minOccurs="0"/>
    <xsd:element name="notDomesticLaw" type="xsd:boolean" minOccurs="0"/>
    <xsd:element name="listed" type="xsd:boolean" minOccurs="0"/>
    <xsd:element name="notContingent" type="xsd:boolean" minOccurs="0"/>
    <xsd:element name="notDomesticIssuance" type="xsd:boolean" minOccurs="0"/>
    <xsd:element name="assignableLoan" type="PCDeliverableObligationCharac" minOccurs="0"/>
    <xsd:element name="consentRequiredLoan" type="PCDeliverableObligationCharac" minOccurs="0"/>
    <xsd:element name="directLoanParticipation" type="LoanParticipation" minOccurs="0"/>
    <xsd:element name="transferable" type="xsd:boolean" minOccurs="0"/>
    <xsd:element name="maximumMaturity" type="Period" minOccurs="0"/>
    <xsd:element name="acceleratedorMatured" type="xsd:boolean" minOccurs="0"/>
    <xsd:element name="notBearer" type="xsd:boolean" minOccurs="0"/>
    <xsd:choice minOccurs="0">
      <xsd:element name="fullFaithAndCreditObliability" type="xsd:boolean"/>
      <xsd:element name="generalFundObligationLiability" type="xsd:boolean"/>
      <xsd:element name="revenueObligationLiability" type="xsd:boolean"/>
    </xsd:choice>
    <xsd:element name="indirectLoanParticipation" type="LoanParticipation" minOccurs="0"/>
    <xsd:element name="excluded" type="xsd:string" minOccurs="0"/>
    <xsd:element name="othReferenceEntityObligations" type="xsd:string" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>

```

[top](#)**Complex Type: EntityType**

<b>Super-types:</b>	<a href="#">Scheme</a> < <b>EntityType</b> (by extension)
<b>Sub-types:</b>	None
<b>Name</b>	EntityType
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">ReferencePair</a>
<b>Abstract</b>	no
<b>Documentation</b>	Defines a coding scheme of the entity types defined in the ISDA First to Default documentation.

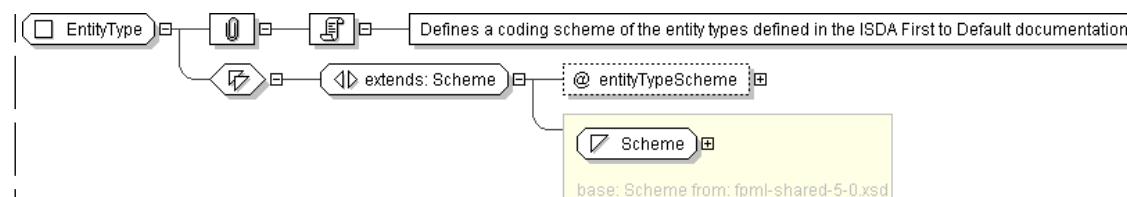
**XML Instance Representation**

```

<...
<entityTypeScheme="xsd:anyURI [0..1]>
<Scheme>
</...>

```

**Diagram**

**Schema Component Representation**

```

<xsd:complexType name="EntityType">
  <xsd:simpleContent>
    <xsd:extension base=" Scheme ">
      <xsd:attribute name="entityTypeScheme" type=" xsd:anyURI " default="http://www.fpml.org/
        coding-scheme/entity-type"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
  
```

[top](#)**Complex Type: FeeLeg**

<b>Super-types:</b>	<a href="#">Leg</a> < <b>FeeLeg</b> (by extension)
<b>Sub-types:</b>	None

<b>Name</b>	FeeLeg
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">CreditDefaultSwap</a>
<b>Abstract</b>	no

**XML Instance Representation**

```

<...>
<id=" xsd:ID [0..1]">
<initialPayment> InitialPayment </initialPayment> [0..1]
  'Specifies a single fixed payment that is payable by the payer to the receiver on the
  initial payment date. The fixed payment to be paid is specified in terms of a known
  currency amount. This element should be used for CDS Index trades and can be used for
  CDS trades where it is necessary to represent a payment from Seller to Buyer. For CDS
  trades where a payment is to be made from Buyer to Seller the feeLeg/singlePayment
  structure must be used.'
<singlePayment> SinglePayment </singlePayment> [0..*]
  'Specifies a single fixed amount that is payable by the buyer to the seller on the fixed
  rate payer payment date. The fixed amount to be paid is specified in terms of a known
  currency amount.'
<periodicPayment> PeriodicPayment </periodicPayment> [0..1]
  'Specifies a periodic schedule of fixed amounts that are payable by the buyer to the seller
  on the fixed rate payer payment dates. The fixed amount to be paid on each payment date can
  be specified in terms of a known currency amount or as an amount calculated on a formula
  basis by reference to a per annum fixed rate. The applicable business day convention
  and business day for adjusting any fixed rate payer payment date if it would otherwise fall
  on a day that is not a business day are those specified in the dateAdjustments element
  within the generalTerms component. ISDA 2003 Term:'
<marketFixedRate> xsd:decimal </marketFixedRate> [0..1]
  'An optional element that only has meaning in a credit index trade. This element contains
  the credit spread ("fair value") at which the trade was executed. Unlike the fixedRate of
  an index, the marketFixedRate varies over the life of the index depending on market
  conditions. The marketFixedRate is the price of the index as quoted by trading desks.'
<paymentDelay> xsd:boolean </paymentDelay> [0..1]
  
```

'Applicable to CDS on MBS to specify whether payment delays are applicable to the fixed Amount. RMBS typically have a payment delay of 5 days between the coupon date of the reference obligation and the payment date of the synthetic swap. CMBS do not, on the other hand, with both payment dates being on the 25th of each month.'

<initialPoints> `xsd:decimal` </initialPoints> [0..1]

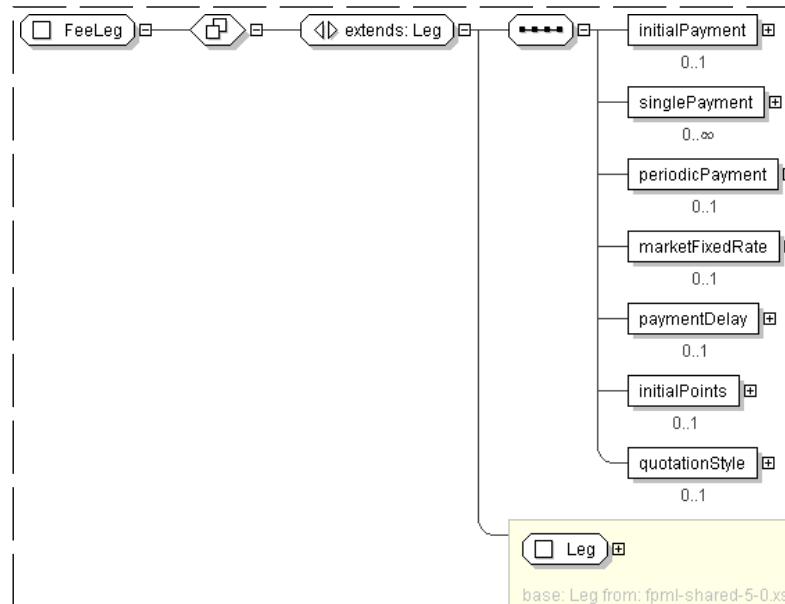
'An optional element that contains the up-front points expressed as a percentage of the notional. An initialPoints value of 5% would be represented as 0.05. The initialPoints element is an alternative to marketFixedRate in quoting the traded level of a trade. When initialPoints is used, the traded level is the sum of fixedRate and initialPoints. The initialPoints is one of the items that are factored into the initialPayment calculation and is payable by the Buyer to the Seller. Note that initialPoints and marketFixedRate may both be present in the same document when both implied values are desired.'

<quotationStyle> `QuotationStyleEnum` </quotationStyle> [0..1]

'The type of quotation that was used between the trading desks. The purpose of this element is to indicate the actual quotation style that was used to quote this trade which may not be apparent when both marketFixedRate and initialPoints are included in the document. When quotationStyle is 'PointsUpFront', the initialPoints element should be populated. When quotationStyle is 'TradedSpread', the marketFixedRate element should be populated.'

<...>

#### Diagram



#### Schema Component Representation

```

<xsd:complexType name="FeeLeg">
  <xsd:complexContent>
    <xsd:extension base=" Leg ">
      <xsd:sequence>
        <xsd:element name="initialPayment" type=" InitialPayment " minOccurs="0"/>
        <xsd:element name="singlePayment" type=" SinglePayment " minOccurs="0" maxOccurs="unbounded"/>
        <xsd:element name="periodicPayment" type=" PeriodicPayment " minOccurs="0"/>
        <xsd:element name="marketFixedRate" type=" xsd:decimal " minOccurs="0"/>
        <xsd:element name="paymentDelay" type=" xsd:boolean " minOccurs="0"/>
        <xsd:element name="initialPoints" type=" xsd:decimal " minOccurs="0"/>
        <xsd:element name="quotationStyle" type=" QuotationStyleEnum " minOccurs="0"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
  
```

```
</xsd:complexContent>
</xsd:complexType>
```

## Complex Type: FixedAmountCalculation

Super-types:	None
Sub-types:	None

Name	FixedAmountCalculation
Used by (from the same schema document)	Complex Type <a href="#">PeriodicPayment</a>
Abstract	no

### XML Instance Representation

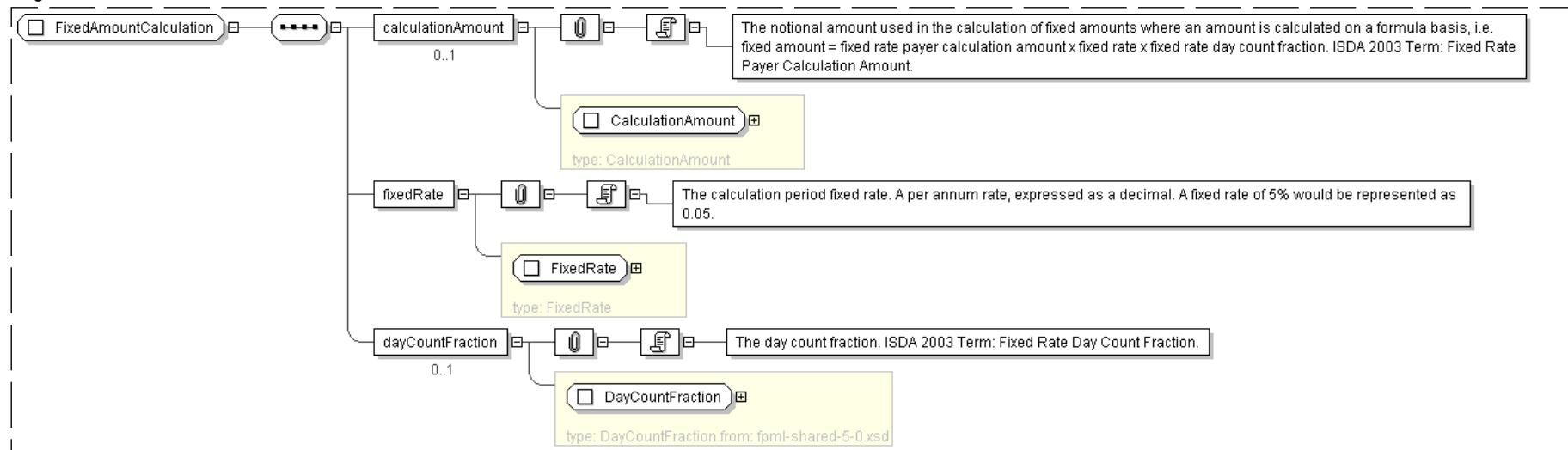
```
<...>
<calculationAmount> CalculationAmount </calculationAmount> [0..1]
'The notional amount used in the calculation of fixed amounts where an amount is calculated
on a formula basis, i.e. fixed amount = fixed rate payer calculation amount x fixed rate
x fixed rate day count fraction. ISDA 2003 Term: Fixed Rate Payer Calculation Amount.'

<fixedRate> FixedRate </fixedRate> [1]
'The calculation period fixed rate. A per annum rate, expressed as a decimal. A fixed rate
of 5% would be represented as 0.05.'

<dayCountFraction> DayCountFraction </dayCountFraction> [0..1]
'The day count fraction. ISDA 2003 Term: Fixed Rate Day Count Fraction.'

</...>
```

### Diagram



### Schema Component Representation

```
<xsd:complexType name="FixedAmountCalculation">
  <xsd:sequence>
    <xsd:element name="calculationAmount" type=" CalculationAmount " minOccurs="0"/>
    <xsd:element name="fixedRate" type=" FixedRate "/>
    <xsd:element name="dayCountFraction" type=" DayCountFraction " minOccurs="0"/>
  </xsd:sequence>
```

**Complex Type: FixedRate**Super-types: [xsd:decimal < FixedRate](#) (by extension)

Sub-types: None

Name FixedRate

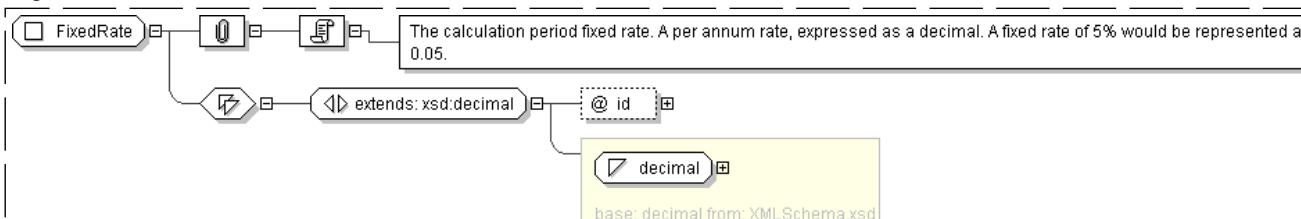
Used by (from the same schema document) Complex Type [FixedAmountCalculation](#)

Abstract no

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

**XML Instance Representation**

```
<...>
  id="xsd:ID [0..1]">
  xsd:decimal
</...>
```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="FixedRate">
  <xsd:simpleContent>
    <xsd:extension base="xsd:decimal">
      <xsd:attribute name="id" type="xsd:ID" use="optional"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
  
```

**Complex Type: FixedRateReference**Super-types: [Reference < FixedRateReference](#) (by extension)

Sub-types: None

Name FixedRateReference

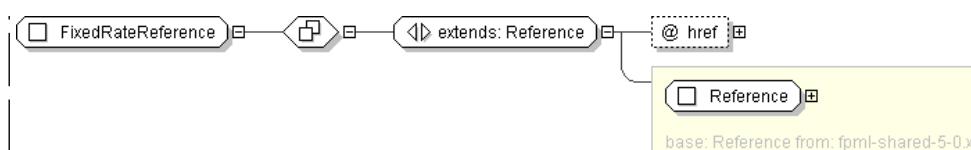
Used by (from the same schema document) Complex Type [CreditOptionStrike](#)

Abstract no

**XML Instance Representation**

```
<...>
  href="xsd:IDREF [1]">
```

**Diagram**

**Schema Component Representation**

```

<xsd:complexType name="FixedRateReference">
  <xsd:complexContent>
    <xsd:extension base=" Reference ">
      <xsd:attribute name="href" type=" xsd:IDREF " use="required" reference="FixedRate" />
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
  
```

[top](#)**Complex Type: FloatingAmountEvents**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None

<b>Name</b>	FloatingAmountEvents
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">ProtectionTerms</a>
<b>Abstract</b>	no

**XML Instance Representation**

```

<...>
<failureToPayPrincipal> xsd:boolean </failureToPayPrincipal> [0..1]
'A floating rate payment event. Corresponds to the failure by the Reference Entity to pay
an expected principal amount or the payment of an actual principal amount that is less than
the expected principal amount. ISDA 2003 Term: Failure to Pay Principal.'

<interestShortfall> InterestShortFall </interestShortfall> [0..1]
'A floating rate payment event. With respect to any Reference Obligation Payment Date,
either (a) the non-payment of an Expected Interest Amount or (b) the payment of an
Actual Interest Amount that is less than the Expected Interest Amount. ISDA 2003 Term:
Interest Shortfall.'

<writedown> xsd:boolean </writedown> [0..1]
'A floating rate payment event. Results from the fact that the underlyer writes down
its outstanding principal amount. ISDA 2003 Term: Writedown.'

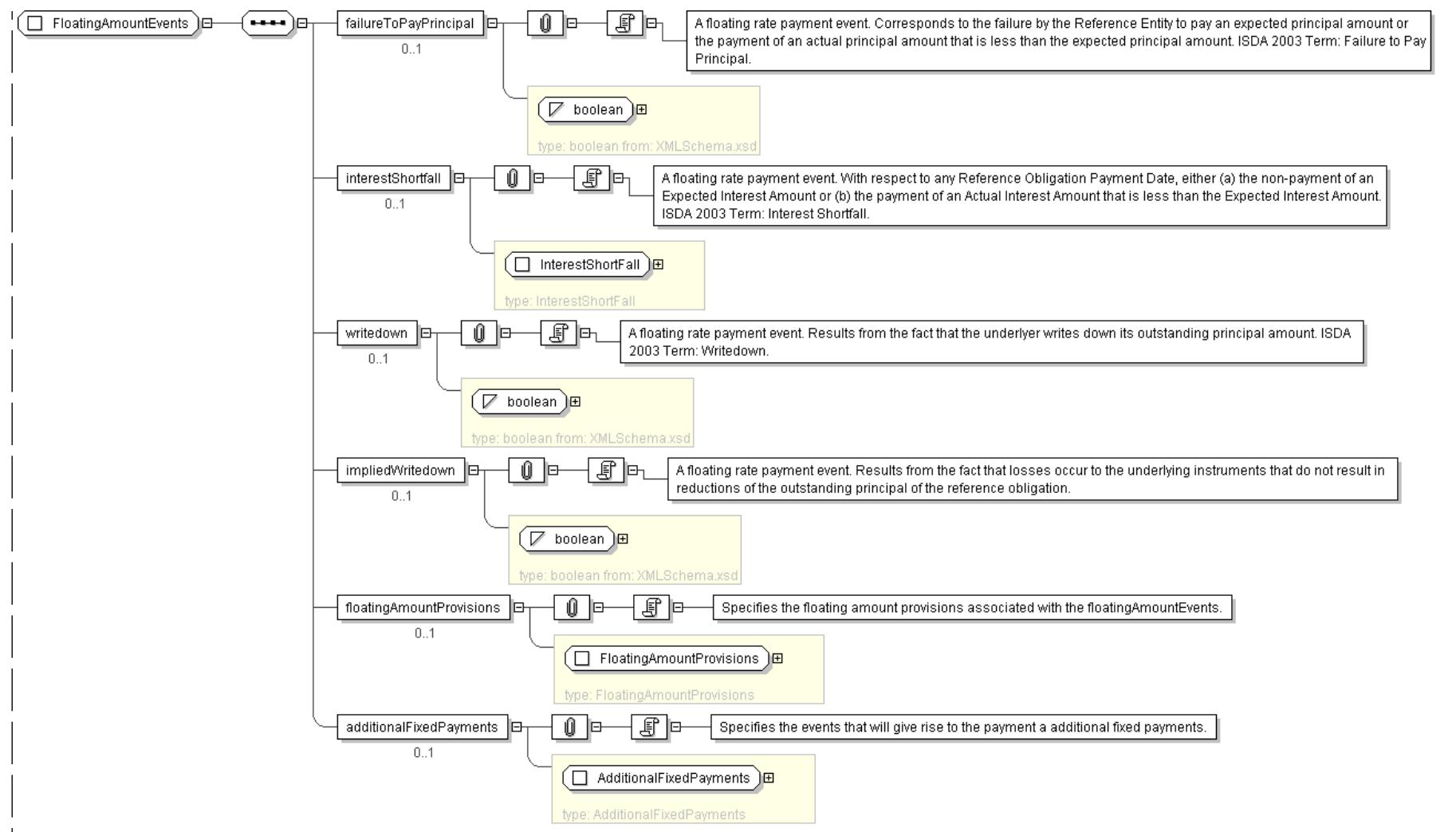
<impliedWritedown> xsd:boolean </impliedWritedown> [0..1]
'A floating rate payment event. Results from the fact that losses occur to the
underlying instruments that do not result in reductions of the outstanding principal of
the reference obligation.'

<floatingAmountProvisions> FloatingAmountProvisions </floatingAmountProvisions> [0..1]
'Specifies the floating amount provisions associated with the floatingAmountEvents.'

<additionalFixedPayments> AdditionalFixedPayments </additionalFixedPayments> [0..1]
'Specifies the events that will give rise to the payment a additional fixed payments.'

</...>
  
```

**Diagram**



#### Schema Component Representation

```

<xsd:complexType name="FloatingAmountEvents">
  <xsd:sequence>
    <xsd:element name="failureToPayPrincipal" type="xsd:boolean" minOccurs="0"/>
    <xsd:element name="interestShortfall" type="InterestShortFall" minOccurs="0"/>
    <xsd:element name="writtenDown" type="xsd:boolean" minOccurs="0"/>
    <xsd:element name="impliedWrittenDown" type="xsd:boolean" minOccurs="0"/>
    <xsd:element name="floatingAmountProvisions" type="FloatingAmountProvisions" minOccurs="0"/>
    <xsd:element name="additionalFixedPayments" type="AdditionalFixedPayments" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>

```

top

#### Complex Type: FloatingAmountProvisions

Super-types:

None

Sub-types:

None

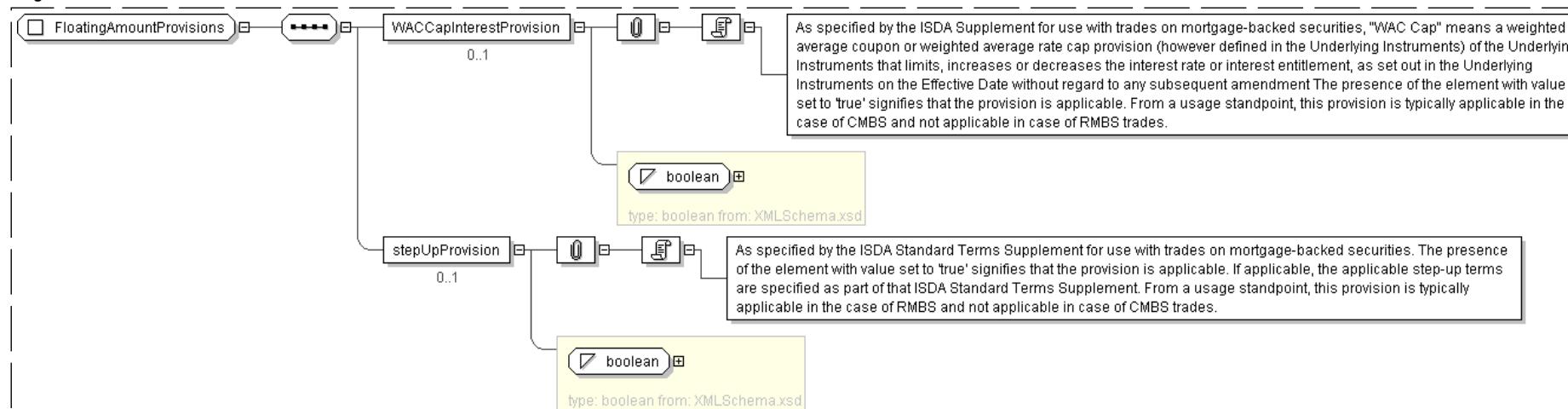
Name	FloatingAmountProvisions
Used by (from the same schema document)	Complex Type <a href="#">FloatingAmountEvents</a>
Abstract	no

**XML Instance Representation**

```
<...>
<WACCapInterestProvision> xsd:boolean </WACCapInterestProvision> [0..1]
'As specified by the ISDA Supplement for use with trades on mortgage-backed securities,
\"WAC Cap\" means a weighted average coupon or weighted average rate cap provision
(however defined in the Underlying Instruments) of the Underlying Instruments that
limits, increases or decreases the interest rate or interest entitlement, as set out in
the Underlying Instruments on the Effective Date without regard to any subsequent amendment
The presence of the element with value set to '\true\' signifies that the provision
is applicable. From a usage standpoint, this provision is typically applicable in the case
of CMBS and not applicable in case of RMBS trades.'

<stepUpProvision> xsd:boolean </stepUpProvision> [0..1]
'As specified by the ISDA Standard Terms Supplement for use with trades on mortgage-
backed securities. The presence of the element with value set to '\true\' signifies that
the provision is applicable. If applicable, the applicable step-up terms are specified as
part of that ISDA Standard Terms Supplement. From a usage standpoint, this provision
is typically applicable in the case of RMBS and not applicable in case of CMBS trades.'

</...>
```

**Diagram****Schema Component Representation**

```
<xsd:complexType name="FloatingAmountProvisions">
  <xsd:sequence>
    <xsd:element name="WACCapInterestProvision" type="xsd:boolean" minOccurs="0"/>
    <xsd:element name="stepUpProvision" type="xsd:boolean" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>
```

top

**Complex Type: GeneralTerms**

Super-types:	None
Sub-types:	None

<b>Name</b>	GeneralTerms
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">CreditDefaultSwap</a>
<b>Abstract</b>	no

**XML Instance Representation**

```

<...>
<effectiveDate> AdjustableDate2 </effectiveDate> [0..1]
'The first day of the term of the trade. This day may be subject to adjustment in
accordance with a business day convention. ISDA 2003 Term: Effective Date.'

<scheduledTerminationDate> AdjustableDate2 </scheduledTerminationDate> [0..1]
'The scheduled date on which the credit protection will lapse. May be specified as an
adjusting or non-adjusting date or alternatively as a period offset from the effective
date. ISDA 2003 Term: Scheduled Termination Date.'

<buyerPartyReference> PartyReference </buyerPartyReference> [1]
'A reference to the party that buys this instrument, i.e. pays for this instrument and
receives the rights defined by it. See 2000 ISDA definitions Article 11.1 (b). In the case
of FRAs this is the fixed rate payer.'

<buyerAccountReference> AccountReference </buyerAccountReference> [0..1]
'A reference to the account that buys this instrument.'

<sellerPartyReference> PartyReference </sellerPartyReference> [1]
'A reference to the party that sells ("writes") this instrument, i.e. that grants the
rights defined by this instrument and in return receives a payment for it. See 2000
ISDA definitions Article 11.1 (a). In the case of FRAs this is the floating rate payer.'

<sellerAccountReference> AccountReference </sellerAccountReference> [0..1]
'A reference to the account that sells this instrument.'

<dateAdjustments> BusinessDayAdjustments </dateAdjustments> [0..1]
'ISDA 2003 Terms: Business Day and Business Day Convention.'

Start Choice [1]
<referenceInformation> ReferenceInformation </referenceInformation> [1]
'This element contains all the terms relevant to defining the reference entity and
reference obligation(s).'

<indexReferenceInformation> IndexReferenceInformation </indexReferenceInformation> [1]
'This element contains all the terms relevant to defining the Credit DefaultSwap Index.'

<basketReferenceInformation> BasketReferenceInformation </basketReferenceInformation> [1]
'This element contains all the terms relevant to defining the Credit Default Swap Basket.'

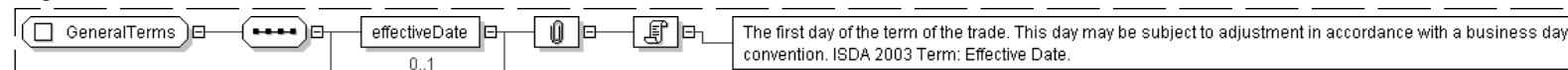
End Choice
<additionalTerm> AdditionalTerm </additionalTerm> [0..*]
'This element is used for representing information contained in the Additional Terms field
of the 2003 Master Credit Derivatives confirm.'

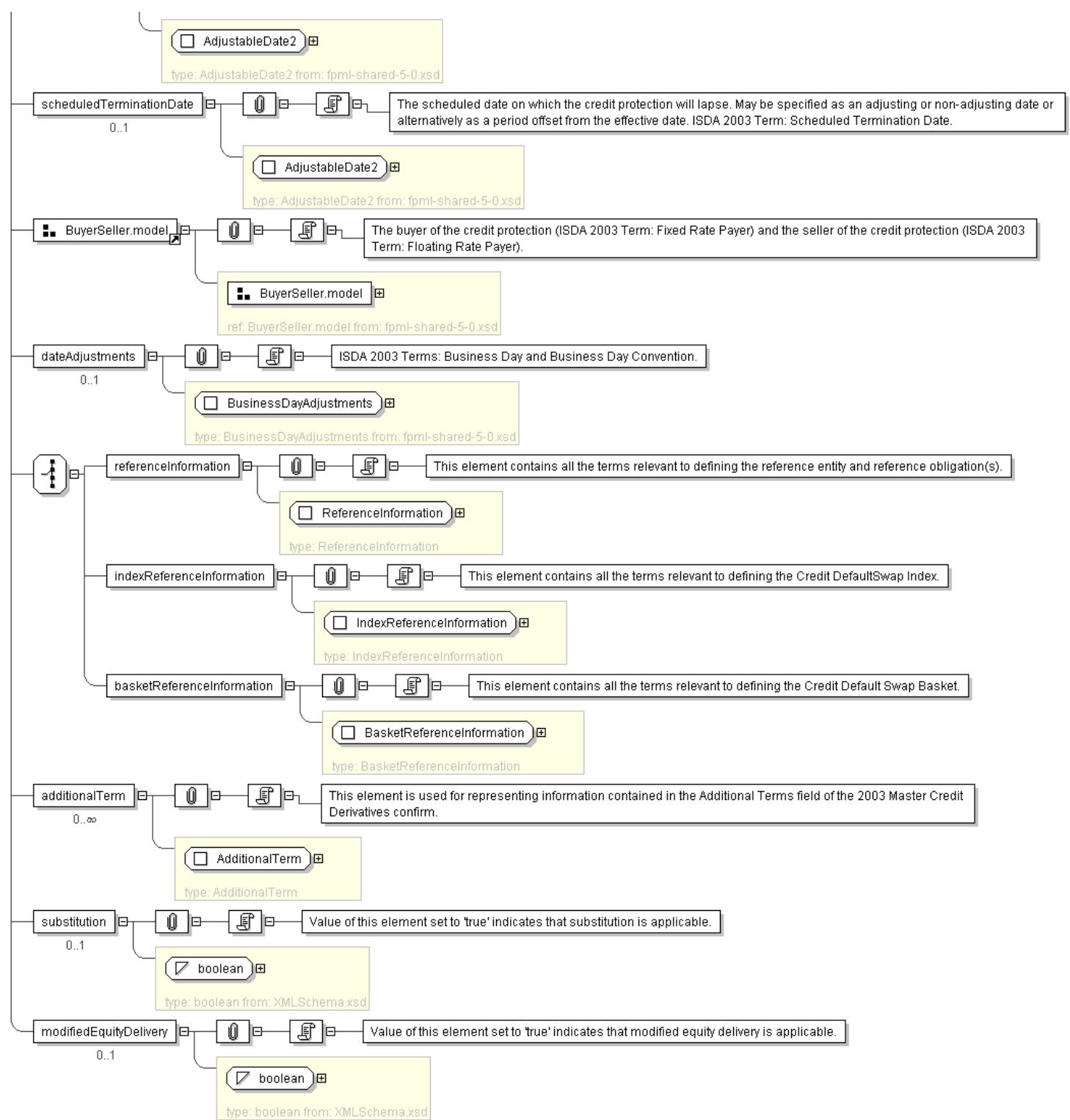
<substitution> xsd:boolean </substitution> [0..1]
'Value of this element set to \'true\' indicates that substitution is applicable.'

<modifiedEquityDelivery> xsd:boolean </modifiedEquityDelivery> [0..1]
'Value of this element set to \'true\' indicates that modified equity delivery is applicable.'

</...>

```

**Diagram**



**Schema Component Representation**

```

<xsd:complexType name="GeneralTerms">
  <xsd:sequence>
    <xsd:element name="effectiveDate" type=" AdjustableDate2 " minOccurs="0" />
    <xsd:element name="scheduledTerminationDate" type=" AdjustableDate2 " minOccurs="0" />
    <xsd:group ref="# BuyerSeller.model "# />
    <xsd:element name="dateAdjustments" type=" BusinessDayAdjustments " minOccurs="0" />
    <xsd:choice>
      <xsd:element name="referenceInformation" type=" ReferenceInformation " />
      <xsd:element name="indexReferenceInformation" type=" IndexReferenceInformation " />
      <xsd:element name="basketReferenceInformation" type=" BasketReferenceInformation " />
    </xsd:choice>
    <xsd:element name="additionalTerm" type=" AdditionalTerm " minOccurs="0" maxOccurs="unbounded" />
    <xsd:element name="substitution" type=" xsd:boolean " minOccurs="0" />
    <xsd:element name="modifiedEquityDelivery" type=" xsd:boolean " minOccurs="0" />
  </xsd:sequence>
</xsd:complexType>

```

[top](#)**Complex Type: IndexAnnexSource**

**Super-types:** [Scheme](#) < **IndexAnnexSource** (by extension)

**Sub-types:** None

<b>Name</b>	IndexAnnexSource
-------------	------------------

<b>Used by (from the same schema document)</b>	Complex Type <a href="#">IndexReferenceInformation</a>
--	--

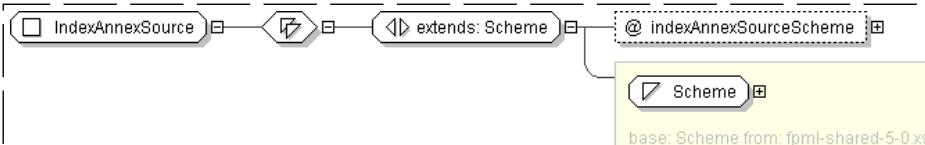
<b>Abstract</b>	no
-----------------	----

**XML Instance Representation**

```

<...
  indexAnnexSourceScheme=" xsd:anyURI [0..1]">
  Scheme
</...>

```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="IndexAnnexSource">
  <xsd:simpleContent>
    <xsd:extension base=" Scheme ">
      <xsd:attribute name="indexAnnexSourceScheme" type=" xsd:anyURI " default="http://www.fpml.org/coding-scheme/cdx-index-annex-source"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>

```

[top](#)**Complex Type: IndexId**

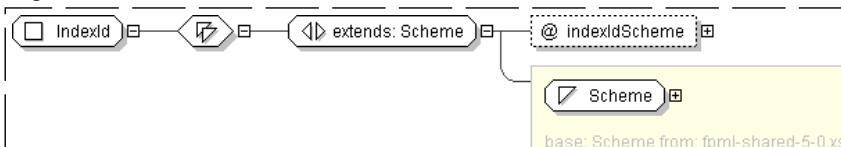
**Super-types:** [Scheme](#) < **IndexId** (by extension)

**Sub-types:** None

<b>Name</b>	IndexId
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">IndexReferenceInformation</a> , Complex Type <a href="#">IndexReferenceInformation</a>
<b>Abstract</b>	no

**XML Instance Representation**

```
<...>
<@ indexIdScheme=" xsd:anyURI [0..1]">
  Scheme
</...>
```

**Diagram****Schema Component Representation**

```
<xsd:complexType name="IndexId">
  <xsd:simpleContent>
    <xsd:extension base=" Scheme ">
      <xsd:attribute name="indexIdScheme" type=" xsd:anyURI " />
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
```

top

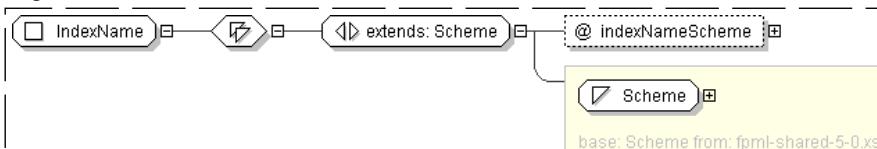
**Complex Type: IndexName**

<b>Super-types:</b>	<a href="#">Scheme</a> < <b>IndexName</b> (by extension)
<b>Sub-types:</b>	None

<b>Name</b>	IndexName
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">IndexReferenceInformation</a>
<b>Abstract</b>	no

**XML Instance Representation**

```
<...>
<@ indexNameScheme=" xsd:anyURI [0..1]">
  Scheme
</...>
```

**Diagram****Schema Component Representation**

```
<xsd:complexType name="IndexName">
  <xsd:simpleContent>
    <xsd:extension base=" Scheme ">
      <xsd:attribute name="indexNameScheme" type=" xsd:anyURI " />
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
```

```
</xsd:simpleContent>
</xsd:complexType>
```

## Complex Type: IndexReferenceInformation

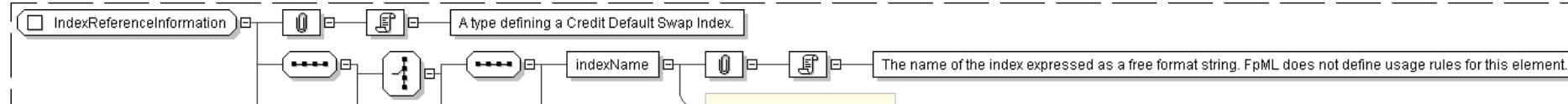
Super-types:	None
Sub-types:	None

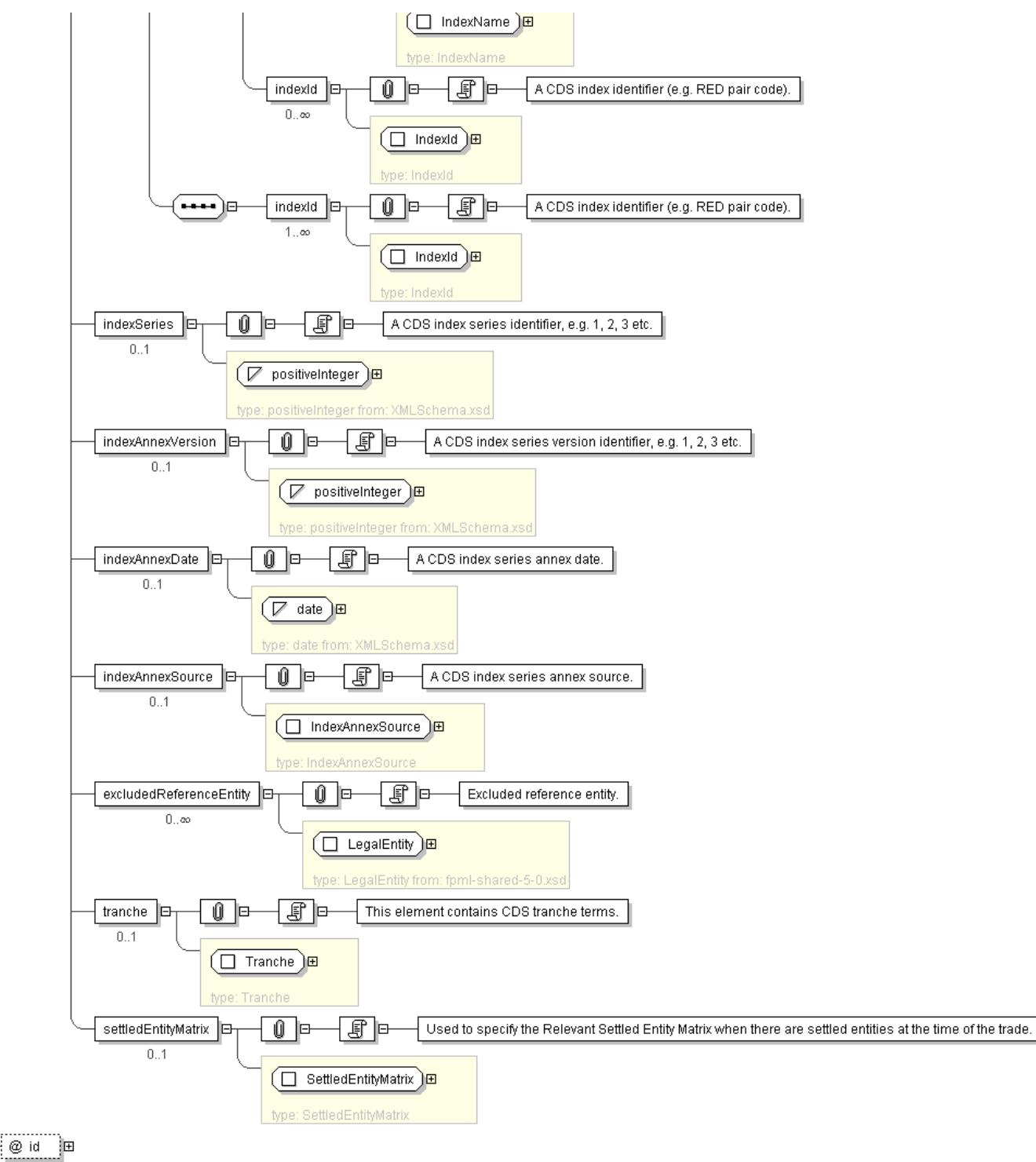
Name	IndexReferenceInformation
Used by (from the same schema document)	Complex Type <a href="#">GeneralTerms</a>
Abstract	no
Documentation	A type defining a Credit Default Swap Index.

### XML Instance Representation

```
<...
  id=" xsd:ID [0..1]">
  Start Choice [1]
    <indexName> IndexName </indexName> [1]
      'The name of the index expressed as a free format string. FpML does not define usage rules
      for this element.'
    <indexId> IndexId </indexId> [0..*]
      'A CDS index identifier (e.g. RED pair code).'
    <indexId> IndexId </indexId> [1..*]
      'A CDS index identifier (e.g. RED pair code).'
  End Choice
  <indexSeries> xsd:positiveInteger </indexSeries> [0..1]
  'A CDS index series identifier, e.g. 1, 2, 3 etc.'
  <indexAnnexVersion> xsd:positiveInteger </indexAnnexVersion> [0..1]
  'A CDS index series version identifier, e.g. 1, 2, 3 etc.'
  <indexAnnexDate> xsd:date </indexAnnexDate> [0..1]
  'A CDS index series annex date.'
  <indexAnnexSource> IndexAnnexSource </indexAnnexSource> [0..1]
  'A CDS index series annex source.'
  <excludedReferenceEntity> LegalEntity </excludedReferenceEntity> [0..*]
  'Excluded reference entity.'
  <tranche> Tranche </tranche> [0..1]
  'This element contains CDS tranche terms.'
  <settledEntityMatrix> SettledEntityMatrix </settledEntityMatrix> [0..1]
  'Used to specify the Relevant Settled Entity Matrix when there are settled entities at the
  time of the trade.'
<...>
```

### Diagram



**Schema Component Representation**

```

<xsd:complexType name="IndexReferenceInformation">
  <xsd:sequence>
    <xsd:choice>
      <xsd:sequence>
        <xsd:element name="indexName" type=" IndexName " />
        <xsd:element name="indexId" type=" IndexId " minOccurs="0" maxOccurs="unbounded" />
      </xsd:sequence>
      <xsd:sequence>
        <xsd:element name="indexId" type=" IndexId " maxOccurs="unbounded" />
      </xsd:sequence>
    </xsd:choice>
    <xsd:element name="indexSeries" type=" xsd:positiveInteger " minOccurs="0" />
    <xsd:element name="indexAnnexVersion" type=" xsd:positiveInteger " minOccurs="0" />
    <xsd:element name="indexAnnexDate" type=" xsd:date " minOccurs="0" />
    <xsd:element name="indexAnnexSource" type=" IndexAnnexSource " minOccurs="0" />
    <xsd:element name="excludedReferenceEntity" type=" LegalEntity "
      minOccurs="0" maxOccurs="unbounded" />
    <xsd:element name="tranche" type=" Tranche " minOccurs="0" />
    <xsd:element name="settledEntityMatrix" type=" SettledEntityMatrix " minOccurs="0" />
  </xsd:sequence>
  <xsd:attribute name="id" type=" xsd:ID " />
</xsd:complexType>

```

[top](#)

## Complex Type: InitialPayment

Super-types:	<a href="#">PaymentBase</a> < <b>InitialPayment</b> (by extension)
Sub-types:	None

Name	InitialPayment
Used by (from the same schema document)	Complex Type <a href="#">FeeLeg</a>
Abstract	no

### XML Instance Representation

```

<...
  id=" xsd:ID [0..1]">
  <payerPartyReference> PartyReference </payerPartyReference> [1]
  'A reference to the party responsible for making the payments defined by this structure.'

  <payerAccountReference> AccountReference </payerAccountReference> [0..1]
  'A reference to the account responsible for making the payments defined by this structure.'

  <receiverPartyReference> PartyReference </receiverPartyReference> [1]
  'A reference to the party that receives the payments corresponding to this structure.'

  <receiverAccountReference> AccountReference </receiverAccountReference> [0..1]
  'A reference to the account that receives the payments corresponding to this structure.'

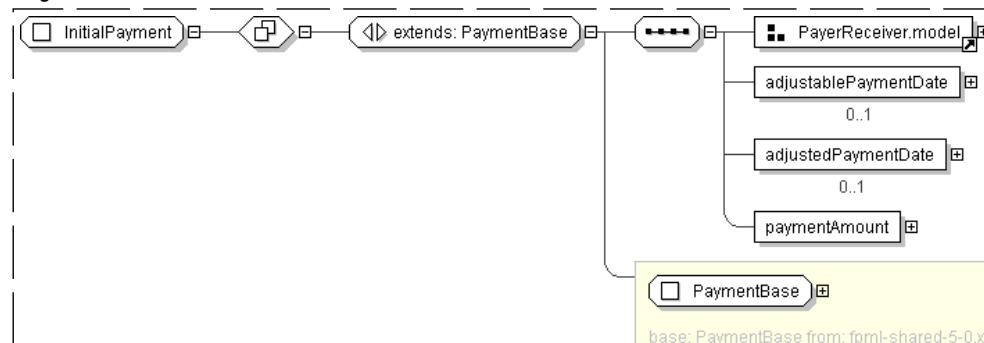
  <adjustablePaymentDate> xsd:date </adjustablePaymentDate> [0..1]
  'A fixed payment date that shall be subject to adjustment in accordance with the
  applicable business day convention if it would otherwise fall on a day that is not a
  business day. The applicable business day convention and business day are those specified
  in the dateAdjustments element within the generalTerms component.'

  <adjustedPaymentDate> xsd:date </adjustedPaymentDate> [0..1]
  'The adjusted payment date. This date should already be adjusted for any applicable
  business day convention. This component 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.'

  <paymentAmount> Money </paymentAmount> [1]
  'A fixed payment amount.'

```

&lt;/...&gt;

**Diagram****Schema Component Representation**

```

<xsd:complexType name="InitialPayment">
  <xsd:complexContent>
    <xsd:extension base=" PaymentBase ">
      <xsd:sequence>
        <xsd:group ref=" PayerReceiver.model ">
          <xsd:element name="adjustablePaymentDate" type=" xsd:date " minOccurs="0"/>
          <xsd:element name="adjustedPaymentDate" type=" xsd:date " minOccurs="0"/>
          <xsd:element name="paymentAmount" type=" Money "/>
        </xsd:sequence>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>

```

top

**Complex Type: InterestShortFall**

Super-types:	None
Sub-types:	None

Name	InterestShortFall
Used by (from the same schema document)	Complex Type <a href="#">FloatingAmountEvents</a>
Abstract	no

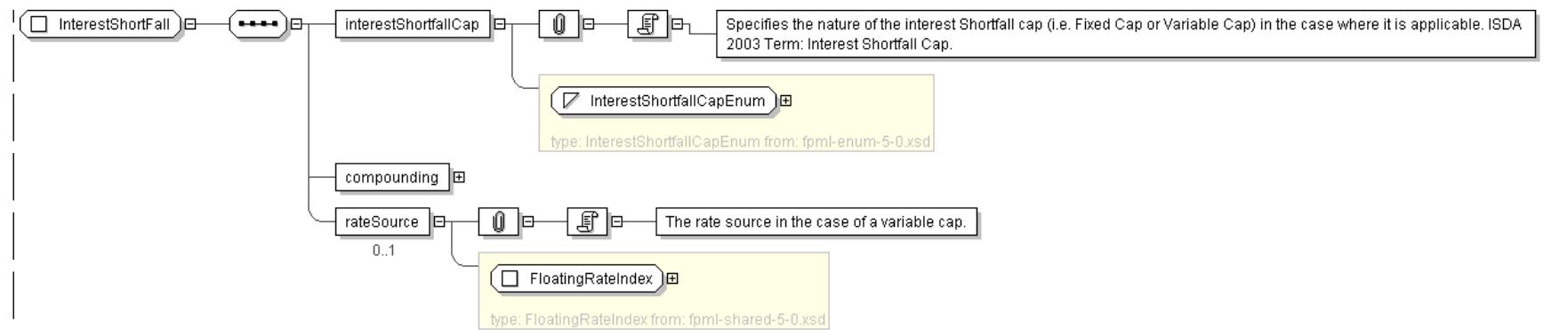
**XML Instance Representation**

```

<...>
<interestShortfallCap> InterestShortfallCapEnum </interestShortfallCap> [1]
'Specifies the nature of the interest Shortfall cap (i.e. Fixed Cap or Variable Cap) in
the case where it is applicable. ISDA 2003 Term: Interest Shortfall Cap.'
<compounding> xsd:boolean </compounding> [1]
<rateSource> FloatingRateIndex </rateSource> [0..1]
'The rate source in the case of a variable cap.'
</...>

```

**Diagram**

**Schema Component Representation**

```

<xsd:complexType name="InterestShortFall">
  <xsd:sequence>
    <xsd:element name="interestShortfallCap" type=" InterestShortfallCapEnum " />
    <xsd:element name="compounding" type=" xsd:boolean " />
    <xsd:element name="rateSource" type=" floatingRateIndex " minOccurs="0 " />
  </xsd:sequence>
</xsd:complexType>

```

[top](#)**Complex Type: LoanParticipation**

<b>Super-types:</b>	<a href="#">PCDeliverableObligationCharac</a> < <b>LoanParticipation</b> (by extension)
<b>Sub-types:</b>	None

<b>Name</b>	LoanParticipation
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">DeliverableObligations</a> , Complex Type <a href="#">DeliverableObligations</a>
<b>Abstract</b>	no

**XML Instance Representation**

```

<...>
<applicable> xsd:boolean </applicable> [1]
'Indicates whether the provision is applicable.'

<partialCashSettlement> xsd:boolean </partialCashSettlement> [0..1]
'Specifies whether either \'Partial Cash Settlement of Assignable Loans\', \'Partial
Cash Settlement of Consent Required Loans\' or \'Partial Cash Settlement of Participations\' is
applicable. If this element is specified and Assignable Loan is a Deliverable Obligation Characteristic, any Assignable Loan that is deliverable, but where a non-receipt of Consent by the Physical Settlement Date has occurred, the Loan can be cash settled rather than physically delivered. If this element is specified and Consent Required Loan is a Deliverable Obligation Characteristic, any Consent Required Loan that is deliverable, but where a non-receipt of Consent by the Physical Settlement Date has occurred, the Loan can be cash settled rather than physically delivered. If this element is specified and Direct Loan Participation is a Deliverable Obligation Characteristic, any Participation that is deliverable, but where this participation has not been effected (has not come into effect) by the Physical Settlement Date, the participation can be cash settled rather than physically delivered.'

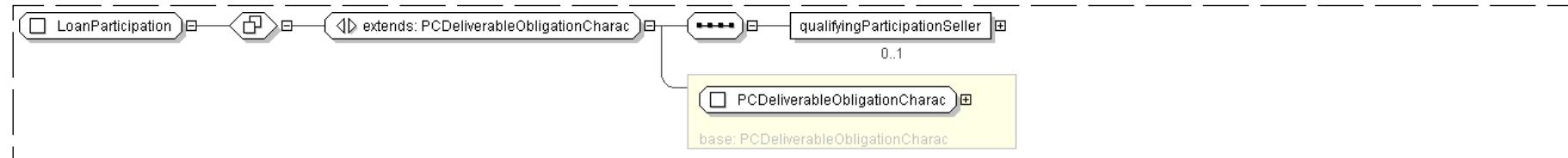
```

```

<qualifyingParticipationSeller> xsd:string </qualifyingParticipationSeller> [0..1]
'If Direct Loan Participation is specified as a deliverable obligation characteristic,
this specifies any requirements for the Qualifying Participation Seller. The requirements may be listed free-form. ISDA 2003 Term: Qualifying Participation Seller'

```

&lt;/...&gt;

**Diagram****Schema Component Representation**

```

<xsd:complexType name="LoanParticipation">
  <xsd:complexContent>
    <xsd:extension base=" PCDeliverableObligationCharac ">
      <xsd:sequence>
        <xsd:element name="qualifyingParticipationSeller" type=" xsd:string " minOccurs="0"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
  
```

[top](#)**Complex Type: MatrixSource**

Super-types:	<a href="#">Scheme</a> < MatrixSource (by extension)
Sub-types:	None

Name	MatrixSource
Used by (from the same schema document)	Complex Type <a href="#">SettledEntityMatrix</a>
Abstract	no

**XML Instance Representation****Diagram****Schema Component Representation**

```

<xsd:complexType name="MatrixSource">
  <xsd:simpleContent>
    <xsd:extension base=" Scheme ">
      <xsd:attribute name="settledEntityMatrixSourceScheme" type=" xsd:anyURI " default="http://
        www.fpml.org/coding-scheme/settled-entity-matrix-source"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
  
```

[top](#)**Complex Type: MultipleValuationDates**

**Super-types:** [SingleValuationDate](#) < [MultipleValuationDates](#) (by extension)

**Sub-types:** None

**Name** MultipleValuationDates

**Used by (from the same schema document)** Complex Type [ValuationDate](#)

**Abstract** no

#### XML Instance Representation

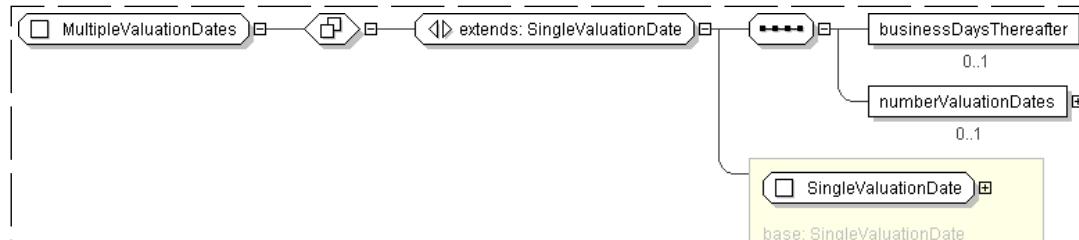
```
<...>
<businessDays> xsd:nonNegativeInteger </businessDays> [0..1]
'A number of business days. Its precise meaning is dependant on the context in which
this element is used. ISDA 2003 Term: Business Day'

<businessDaysThereafter> xsd:positiveInteger </businessDaysThereafter> [0..1]
'The number of business days between successive valuation dates when multiple valuation
dates are applicable for cash settlement. ISDA 2003 Term: Business Days thereafter'

<numberValuationDates> xsd:positiveInteger </numberValuationDates> [0..1]
'Where multiple valuation dates are specified as being applicable for cash settlement,
this element specifies (a) the number of applicable valuation dates, and (b) the number
of business days after satisfaction of all conditions to settlement when the first
such valuation date occurs, and (c) the number of business days thereafter of each
successive valuation date. ISDA 2003 Term: Multiple Valuation Dates'

</...>
```

#### Diagram



#### Schema Component Representation

```

<xsd:complexType name="MultipleValuationDates">
  <xsd:complexContent>
    <xsd:extension base=" SingleValuationDate ">
      <xsd:sequence>
        <xsd:element name="businessDaysThereafter" type=" xsd:positiveInteger " minOccurs="0"/>
        <xsd:element name="numberValuationDates" type=" xsd:positiveInteger " minOccurs="0"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
  
```

top

#### Complex Type: NotDomesticCurrency

**Super-types:** None

**Sub-types:** None

**Name** NotDomesticCurrency

**Used by (from the same schema document)** Complex Type [DeliverableObligations](#) , Complex Type [Obligations](#)

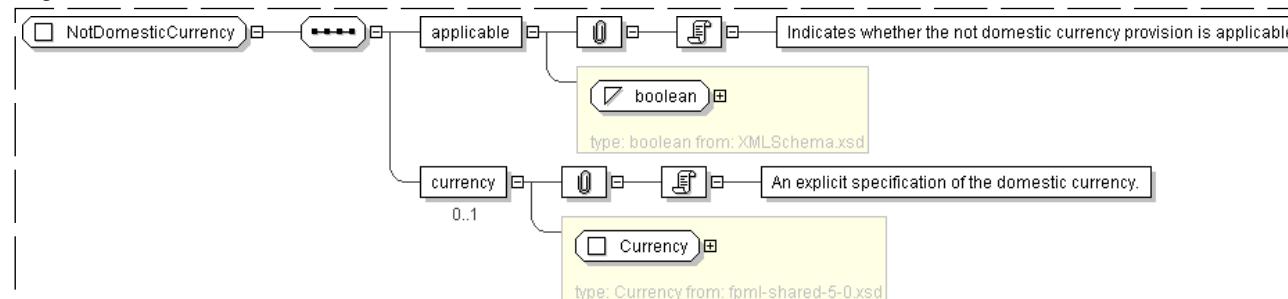
**Abstract** no

**XML Instance Representation**

```
<...>
<applicable> xsd:boolean </applicable> [1]
'Indicates whether the not domestic currency provision is applicable.'

<currency> Currency </currency> [0..1]
'An explicit specification of the domestic currency.

</...>
```

**Diagram****Schema Component Representation**

```
<xsd:complexType name="NotDomesticCurrency">
  <xsd:sequence>
    <xsd:element name="applicable" type="xsd:boolean" />
    <xsd:element name="currency" type="Currency" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>
```

[top](#)**Complex Type: Obligations**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None

<b>Name</b>	Obligations
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">ProtectionTerms</a>
<b>Abstract</b>	no

**XML Instance Representation**

```
<...>
<category> ObligationCategoryEnum </category> [1]
'Used in both obligations and deliverable obligations to represent a class or type
of securities which apply. ISDA 2003 Term: Obligation Category/Deliverable Obligation Category'

<notSubordinated> xsd:boolean </notSubordinated> [0..1]
'An obligation and deliverable obligation characteristic. An obligation that ranks at
least equal with the most senior Reference Obligation in priority of payment or, if
no Reference Obligation is specified in the related Confirmation, the obligations of
the Reference Entity that are senior. ISDA 2003 Term: Not Subordinated'

<specifiedCurrency> SpecifiedCurrency </specifiedCurrency> [0..1]
'An obligation and deliverable obligation characteristic. The currency or currencies in
which an obligation or deliverable obligation must be payable. ISDA 2003 Term:
Specified Currency'
```

<notSovereignLender> xsd:boolean </notSovereignLender> [0..1]

'An obligation and deliverable obligation characteristic. Any obligation that is not primarily (majority) owed to a Sovereign or Supranational Organization. ISDA 2003 Term: Not Sovereign Lender'

<notDomesticCurrency> NotDomesticCurrency </notDomesticCurrency> [0..1]

'An obligation and deliverable obligation characteristic. Any obligation that is payable in any currency other than the domestic currency. Domestic currency is either the currency so specified or, if no currency is specified, the currency of (a) the reference entity, if the reference entity is a sovereign, or (b) the jurisdiction in which the relevant reference entity is organised, if the reference entity is not a sovereign. ISDA 2003 Term: Not Domestic Currency'

<notDomesticLaw> xsd:boolean </notDomesticLaw> [0..1]

'An obligation and deliverable obligation characteristic. If the reference entity is a Sovereign, this means any obligation that is not subject to the laws of the reference entity. If the reference entity is not a sovereign, this means any obligation that is not subject to the laws of the jurisdiction of the reference entity. ISDA 2003 Term: Not Domestic Law'

<listed> xsd:boolean </listed> [0..1]

'An obligation and deliverable obligation characteristic. Indicates whether or not the obligation is quoted, listed or ordinarily purchased and sold on an exchange. ISDA 2003 Term: Listed'

<notDomesticIssuance> xsd:boolean </notDomesticIssuance> [0..1]

'An obligation and deliverable obligation characteristic. Any obligation other than an obligation that was intended to be offered for sale primarily in the domestic market of the relevant Reference Entity. This specifies that the obligation must be an internationally recognized bond. ISDA 2003 Term: Not Domestic Issuance'

Start Choice [0..1]

<fullFaithAndCreditObLiability> xsd:boolean </fullFaithAndCreditObLiability> [1]

'An obligation and deliverable obligation characteristic. Defined in the ISDA published additional provisions for U.S. Municipal as Reference Entity. ISDA 2003 Term: Full Faith and Credit Obligation Liability'

<generalFundObligationLiability> xsd:boolean </generalFundObligationLiability> [1]

'An obligation and deliverable obligation characteristic. Defined in the ISDA published additional provisions for U.S. Municipal as Reference Entity. ISDA 2003 Term: General Fund Obligation Liability'

<revenueObligationLiability> xsd:boolean </revenueObligationLiability> [1]

'An obligation and deliverable obligation characteristic. Defined in the ISDA published additional provisions for U.S. Municipal as Reference Entity. ISDA 2003 Term: Revenue Obligation Liability'

End Choice

<notContingent> xsd:boolean </notContingent> [0..1]

'NOTE: Only allowed as an obligation characteristic under ISDA Credit 1999. In essence Not Contingent means the repayment of principal cannot be dependant on a formula/index, i.e. to prevent the risk of being delivered an instrument that may never pay any element of principal, and to ensure that the obligation is interest bearing (on a regular schedule). ISDA 2003 Term: Not Contingent'

<excluded> xsd:string </excluded> [0..1]

'A free format string to specify any excluded obligations or deliverable obligations, as the case may be, of the reference entity or excluded types of obligations or deliverable obligations. ISDA 2003 Term: Excluded Obligations/Excluded Deliverable Obligations'

<othReferenceEntityObligations> xsd:string </othReferenceEntityObligations> [0..1]

'This element is used to specify any other obligations of a reference entity in both obligations and deliverable obligations. The obligations can be specified free-form. ISDA 2003 Term: Other Obligations of a Reference Entity'

<designatedPriority> Lien </designatedPriority> [0..1]

'Applies to Loan CDS, to indicate what lien level is appropriate for a deliverable obligation. Applies to European Loan CDS, to indicate the Ranking of the obligation. Example: a 2nd lien Loan CDS would imply that the deliverable obligations are 1st or 2nd lien loans.'

<cashSettlementOnly> xsd:boolean </cashSettlementOnly> [0..1]

'An obligation and deliverable obligation characteristic. Defined in the ISDA published Standard Terms Supplement for use with CDS Transactions on Leveraged Loans. ISDA 2003 Term: Cash Settlement Only.'

<deliveryOfCommitments> xsd:boolean </deliveryOfCommitments> [0..1]

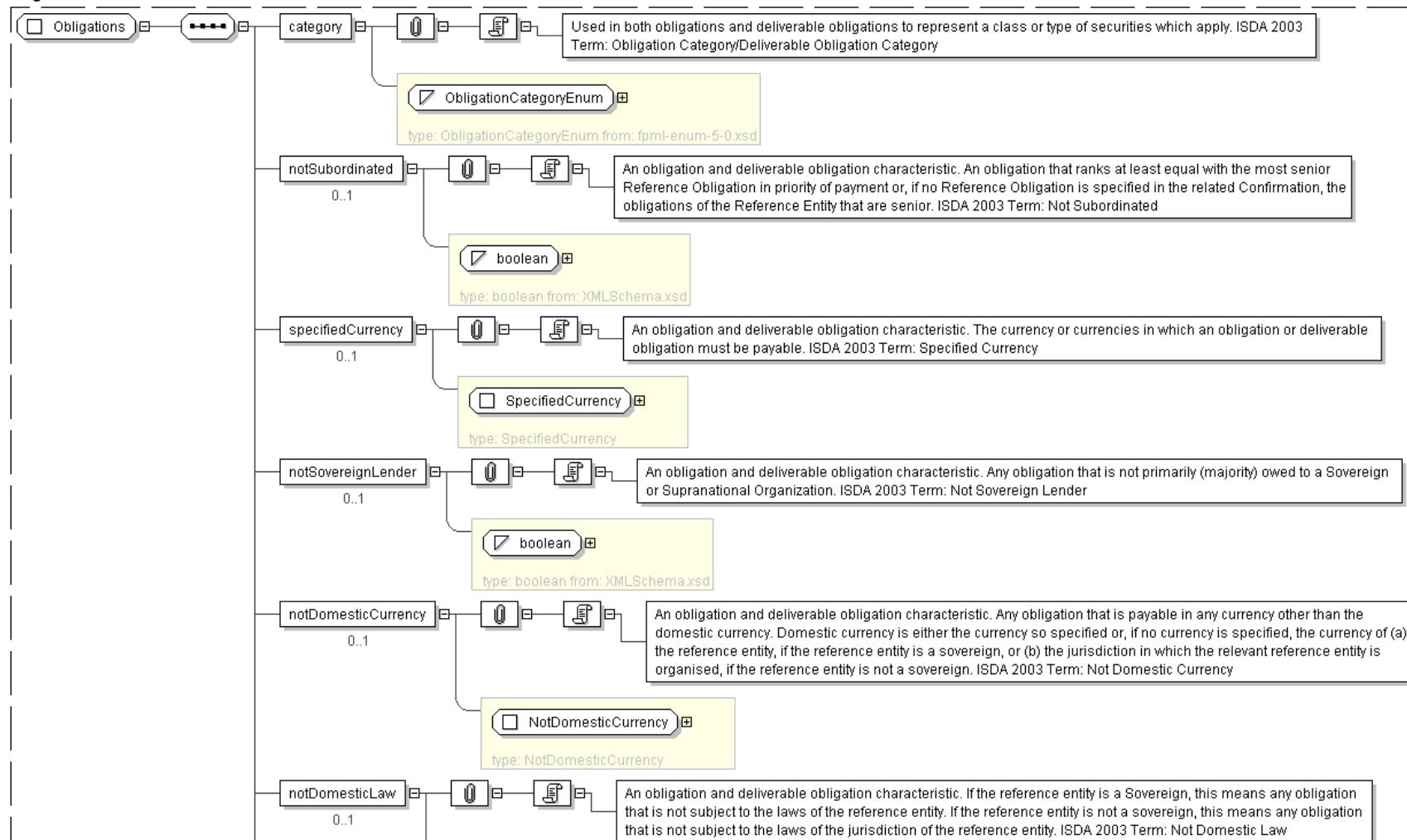
'An obligation and deliverable obligation characteristic. Defined in the ISDA published Standard Terms Supplement for use with CDS Transactions on Leveraged Loans. ISDA 2003 Term: Delivery of Commitments.'

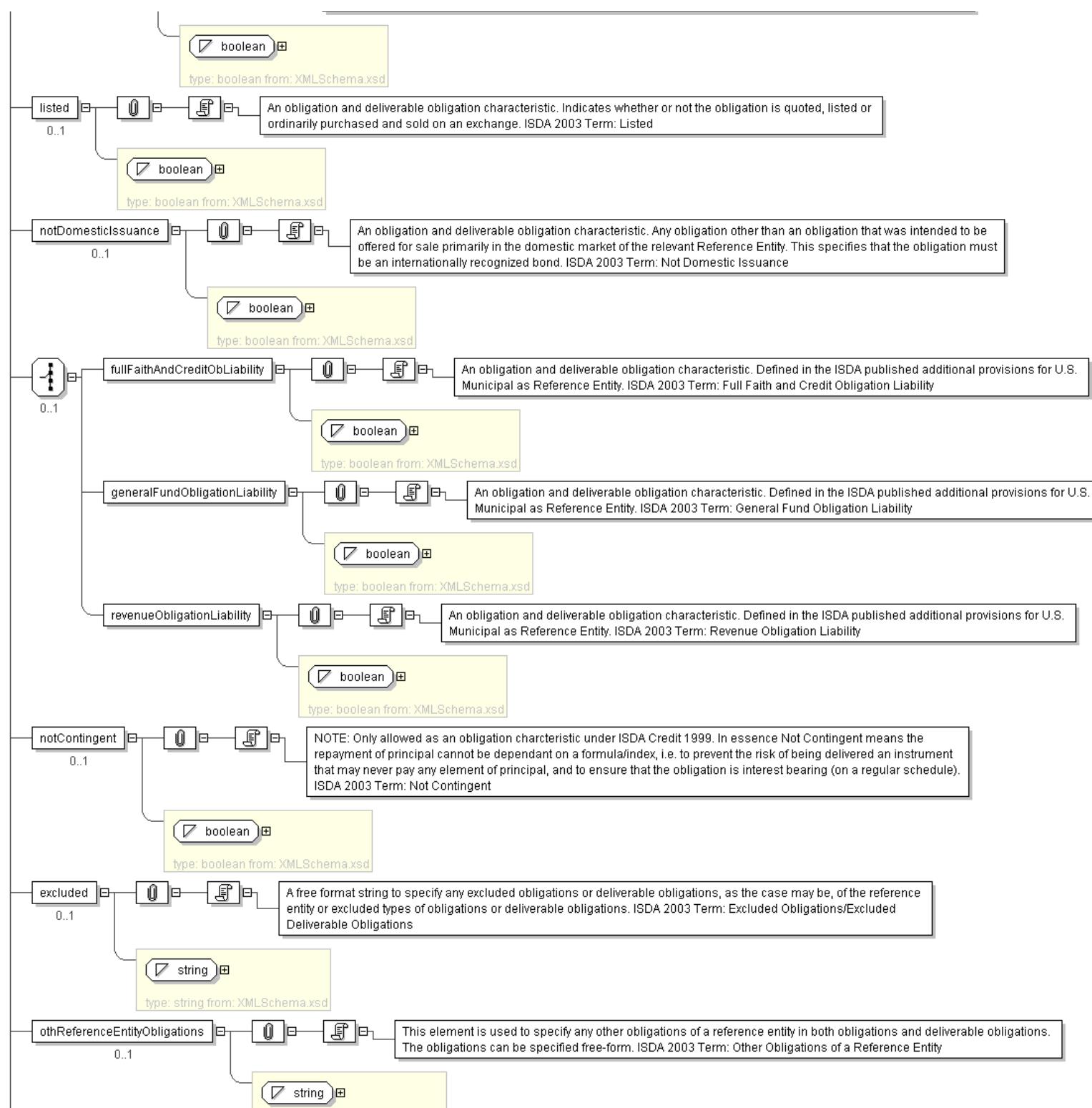
<continuity> xsd:boolean </continuity> [0..1]

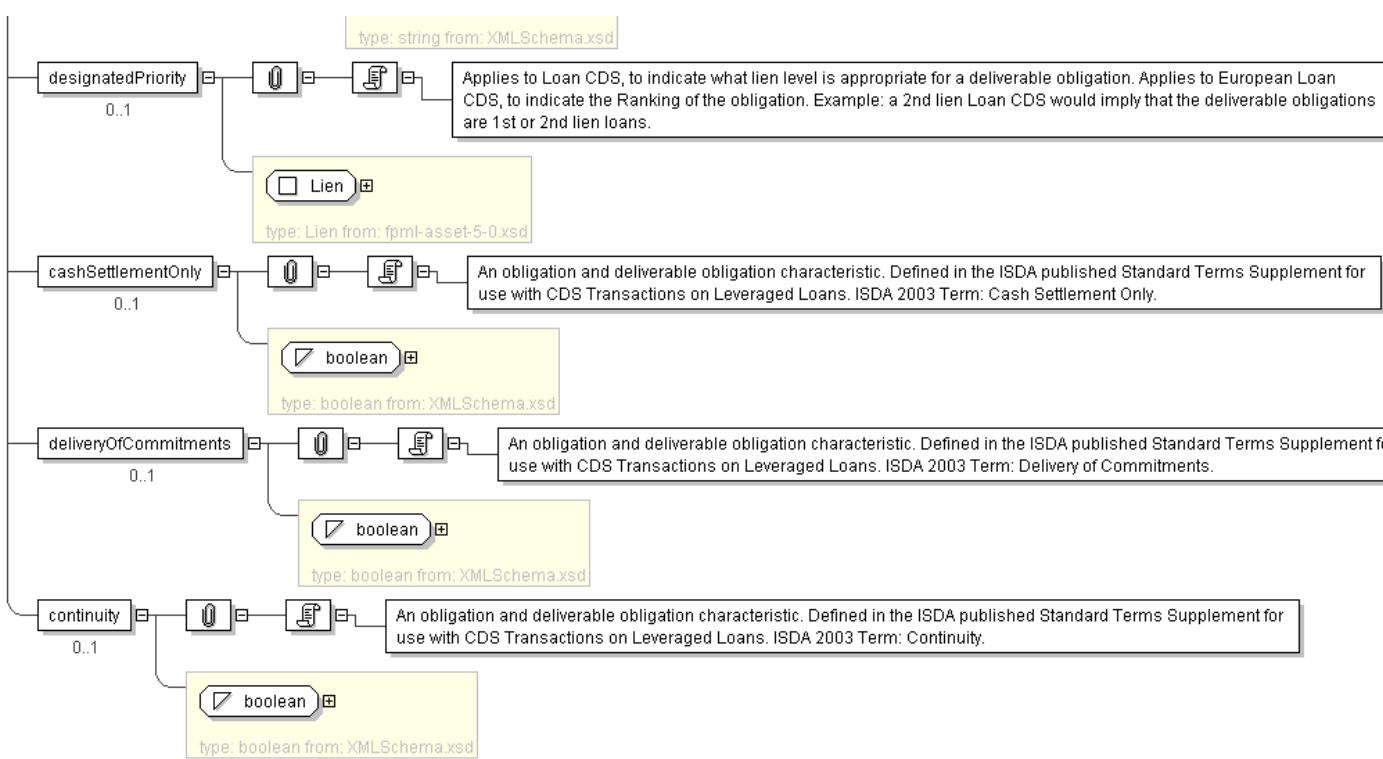
'An obligation and deliverable obligation characteristic. Defined in the ISDA published Standard Terms Supplement for use with CDS Transactions on Leveraged Loans. ISDA 2003 Term: Continuity.'

</...>

#### Diagram







#### Schema Component Representation

```

<xsd:complexType name="Obligations">
  <xsd:sequence>
    <xsd:element name="category" type=" ObligationCategoryEnum " />
    <xsd:element name="notSubordinated" type=" xsd:boolean " minOccurs="0" />
    <xsd:element name="specifiedCurrency" type=" SpecifiedCurrency " minOccurs="0" />
    <xsd:element name="notSovereignLender" type=" xsd:boolean " minOccurs="0" />
    <xsd:element name="notDomesticCurrency" type=" NotDomesticCurrency " minOccurs="0" />
    <xsd:element name="notDomesticLaw" type=" xsd:boolean " minOccurs="0" />
    <xsd:element name="listed" type=" xsd:boolean " minOccurs="0" />
    <xsd:element name="notDomesticIssuance" type=" xsd:boolean " minOccurs="0" />
    <xsd:choice minOccurs="0">
      <xsd:element name="fullFaithAndCreditObLiability" type=" xsd:boolean " />
      <xsd:element name="generalFundObligationLiability" type=" xsd:boolean " />
      <xsd:element name="revenueObligationLiability" type=" xsd:boolean " />
    </xsd:choice>
    <xsd:element name="notContingent" type=" xsd:boolean " minOccurs="0" />
    <xsd:element name="excluded" type=" xsd:string " minOccurs="0" />
    <xsd:element name="othReferenceEntityObligations" type=" xsd:string " minOccurs="0" />
    <xsd:element name="designatedPriority" type=" Lien " minOccurs="0" />
    <xsd:element name="cashSettlementOnly" type=" xsd:boolean " minOccurs="0" />
    <xsd:element name="deliveryOfCommitments" type=" xsd:boolean " minOccurs="0" />
    <xsd:element name="continuity" type=" xsd:boolean " minOccurs="0" />
  </xsd:sequence>
</xsd:complexType>

```

<b>Super-types:</b>	None
<b>Sub-types:</b>	<ul style="list-style-type: none"> <li>• <a href="#">LoanParticipation</a> (by extension)</li> </ul>

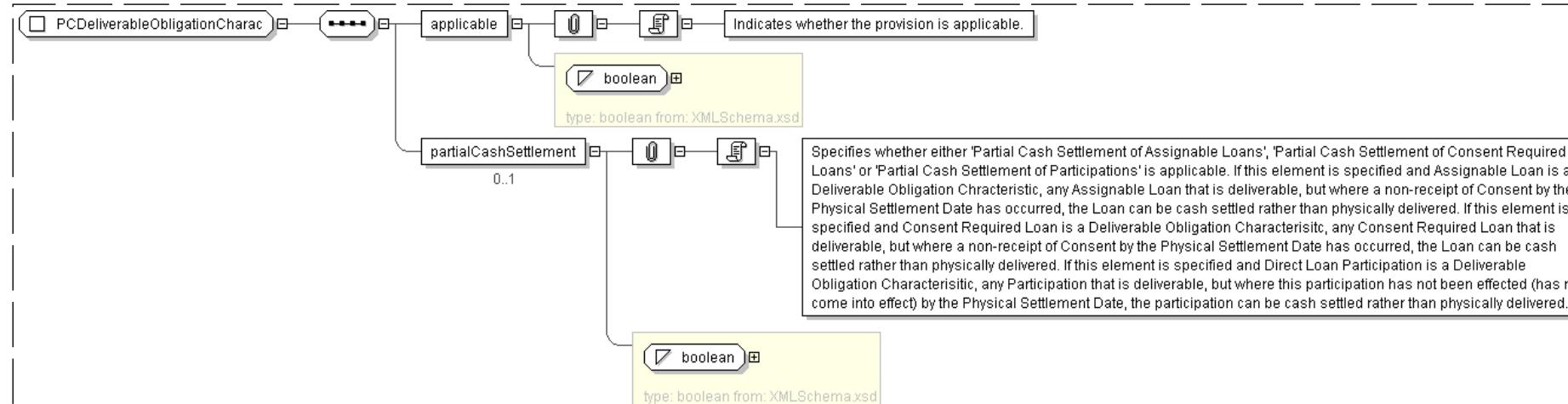
<b>Name</b>	PCDeliverableObligationCharac
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">DeliverableObligations</a> , Complex Type <a href="#">DeliverableObligations</a>
<b>Abstract</b>	no

**XML Instance Representation**

```
<...>
<applicable> xsd:boolean </applicable> [1]
  'Indicates whether the provision is applicable.'

<partialCashSettlement> xsd:boolean </partialCashSettlement> [0..1]
  'Specifies whether either \'Partial Cash Settlement of Assignable Loans\', \'Partial
  Cash Settlement of Consent Required Loans\' or \'Partial Cash Settlement of Participations\' is
  applicable. If this element is specified and Assignable Loan is a Deliverable
  Obligation Characteristic, any Assignable Loan that is deliverable, but where a non-receipt
  of Consent by the Physical Settlement Date has occurred, the Loan can be cash settled
  rather than physically delivered. If this element is specified and Consent Required Loan is
  a Deliverable Obligation Characteristic, any Consent Required Loan that is deliverable,
  but where a non-receipt of Consent by the Physical Settlement Date has occurred, the Loan
  can be cash settled rather than physically delivered. If this element is specified and
  Direct Loan Participation is a Deliverable Obligation Characteristic, any Participation
  that is deliverable, but where this participation has not been effected (has not come
  into effect) by the Physical Settlement Date, the participation can be cash settled rather
  than physically delivered.'

</...>
```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="PCDeliverableObligationCharac">
  <xsd:sequence>
    <xsd:element name="applicable" type="xsd:boolean" />
    <xsd:element name="partialCashSettlement" type="xsd:boolean" minOccurs="0" />
  </xsd:sequence>
</xsd:complexType>
  
```

**Super-types:** [PaymentBase](#) < **PeriodicPayment** (by extension)

**Sub-types:** None

<b>Name</b>	PeriodicPayment
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">FeeLeg</a>
<b>Abstract</b>	no

#### XML Instance Representation

```

<...
  id=" xsd:ID [0..1]">
  <paymentFrequency> Period </paymentFrequency> [0..1]
  'The time interval between regular fixed rate payer payment dates.'

  <firstPeriodStartDate> xsd:date </firstPeriodStartDate> [0..1]
  'The start date of the initial calculation period if such date is not equal to the
  trade's effective date. It must only be specified if it is not equal to the effective date.
  The applicable business day convention and business day are those specified in
  the dateAdjustments element within the generalTerms component (or in a transaction
  supplement FpML representation defined within the referenced general terms
  confirmation agreement).'

  <firstPaymentDate> xsd:date </firstPaymentDate> [0..1]
  'The first unadjusted fixed rate payer payment date. The applicable business day convention
  and business day are those specified in the dateAdjustments element within the
  generalTerms component (or in a transaction supplement FpML representation defined within
  the referenced general terms confirmation agreement). ISDA 2003 Term: Fixed Rate Payer
  Payment Date'

  <lastRegularPaymentDate> xsd:date </lastRegularPaymentDate> [0..1]
  'The last regular unadjusted fixed rate payer payment date. The applicable business
  day convention and business day are those specified in the dateAdjustments element within
  the generalTerms component (or in a transaction supplement FpML representation defined
  within the referenced general terms confirmation agreement). This element should only
  be included if there is a final payment stub, i.e. where the last regular unadjusted fixed
  rate payer payment date is not equal to the scheduled termination date. ISDA 2003 Term:
  Fixed Rate Payer Payment Date'

  <rollConvention> RollConventionEnum </rollConvention> [0..1]
  'Used in conjunction with the effectiveDate, scheduledTerminationDate,
  firstPaymentDate, lastRegularPaymentDate and paymentFrequency to determine the regular
  fixed rate payer payment dates.'

Start Choice [1]
  <fixedAmount> Money </fixedAmount> [1]
  'A fixed payment amount. ISDA 2003 Term: Fixed Amount'

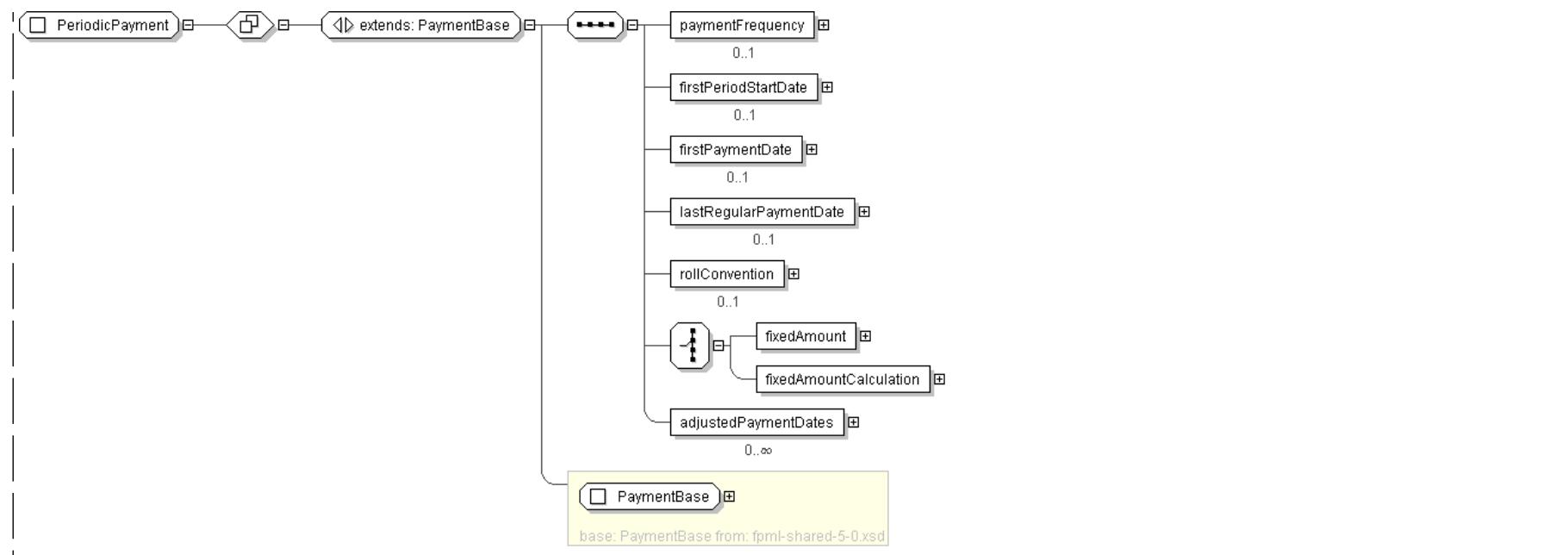
  <fixedAmountCalculation> FixedAmountCalculation </fixedAmountCalculation> [1]
  'This element contains all the terms relevant to calculating a fixed amount where the
  fixed amount is calculated by reference to a per annum fixed rate. There is no
  corresponding ISDA 2003 Term. The equivalent is Sec 5.1 ("Calculation of Fixed Amount")
  but this in itself is not a defined Term.'

End Choice
  <adjustedPaymentDates> AdjustedPaymentDates </adjustedPaymentDates> [0..*]
  'An optional cashflow-like structure allowing the equivalent representation of the
  periodic fixed payments in terms of a series of adjusted payment dates and amounts. This
  is intended to support application integration within an organisation and is not intended
  for use in inter-firm communication or confirmations. ISDA 2003 Term: Fixed Rate Payer
  Payment Date'

</...>

```

#### Diagram



#### Schema Component Representation

```

<xsd:complexType name="PeriodicPayment">
  <xsd:complexContent>
    <xsd:extension base=" PaymentBase " >
      <xsd:sequence>
        <xsd:element name="paymentFrequency" type=" Period " minOccurs="0"/>
        <xsd:element name="firstPeriodStartDate" type=" xsd:date " minOccurs="0"/>
        <xsd:element name="firstPaymentDate" type=" xsd:date " minOccurs="0"/>
        <xsd:element name="lastRegularPaymentDate" type=" xsd:date " minOccurs="0"/>
        <xsd:element name="rollConvention" type=" RollConventionEnum " minOccurs="0"/>
        <xsd:choice>
          <xsd:element name="fixedAmount" type=" Money " />
          <xsd:element name="fixedAmountCalculation" type=" FixedAmountCalculation " />
        </xsd:choice>
        <xsd:element name="adjustedPaymentDates" type=" AdjustedPaymentDates "
          minOccurs="0" maxOccurs="unbounded"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

top

#### Complex Type: PhysicalSettlementPeriod

Super-types:	None
Sub-types:	None

Name	PhysicalSettlementPeriod
Used by (from the same schema document)	Complex Type <a href="#">PhysicalSettlementTerms</a>
Abstract	no

#### XML Instance Representation

```

<...>
| Start Choice [1]
|
```

```

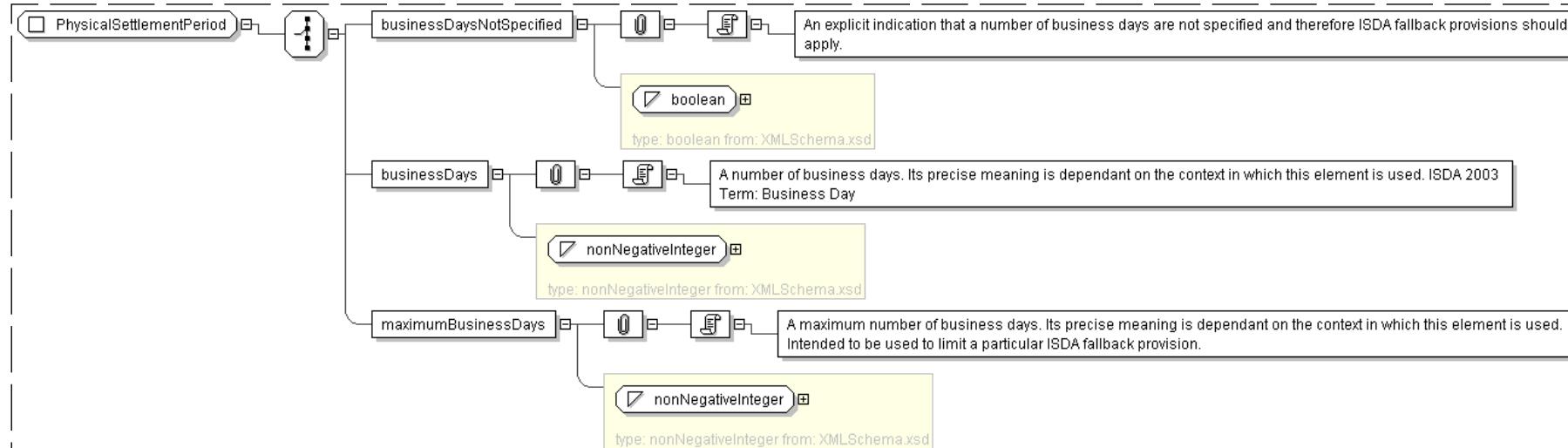
<businessDaysNotSpecified> xsd:boolean </businessDaysNotSpecified> [1]
'An explicit indication that a number of business days are not specified and therefore
ISDA fallback provisions should apply.'

<businessDays> xsd:nonNegativeInteger </businessDays> [1]
'A number of business days. Its precise meaning is dependant on the context in which
this element is used. ISDA 2003 Term: Business Day'

<maximumBusinessDays> xsd:nonNegativeInteger </maximumBusinessDays> [1]
'A maximum number of business days. Its precise meaning is dependant on the context in
which this element is used. Intended to be used to limit a particular ISDA fallback provision.'

```

End Choice  
</...>

**Diagram****Schema Component Representation**

```

<xsd:complexType name="PhysicalSettlementPeriod">
  <xsd:choice>
    <xsd:element name="businessDaysNotSpecified" type="xsd:boolean" />
    <xsd:element name="businessDays" type="xsd:nonNegativeInteger" />
    <xsd:element name="maximumBusinessDays" type="xsd:nonNegativeInteger" />
  </xsd:choice>
</xsd:complexType>

```

top

**Complex Type: PhysicalSettlementTerms**

Super-types:	<a href="#">SettlementTerms</a> < <b>PhysicalSettlementTerms</b> (by extension)
Sub-types:	None

Name	PhysicalSettlementTerms
Used by (from the same schema document)	Complex Type <a href="#">CreditDefaultSwap</a>
Abstract	no

**XML Instance Representation**

&lt;...&gt;

```

'id=" xsd:ID [0..1]">
<settlementCurrency> Currency </settlementCurrency> [0..1]
'ISDA 2003 Term: Settlement Currency'

<physicalSettlementPeriod> PhysicalSettlementPeriod </physicalSettlementPeriod> [0..1]
'The number of business days used in the determination of the physical settlement date.
The physical settlement date is this number of business days after all applicable conditions
to settlement are satisfied. If a number of business days is not specified fallback
provisions apply for determining the number of business days. If Section 8.5/8.6 of
the 1999/2003 ISDA Definitions are to apply the businessDaysNotSpecified element should
be included. If a specified number of business days are to apply these should be specified
in the businessDays element. If Section 8.5/8.6 of the 1999/2003 ISDA Definitions are to
apply but capped at a maximum number of business days then the maximum number should
be specified in the maximumBusinessDays element. ISDA 2003 Term: Physical Settlement Period'

<deliverableObligations> DeliverableObligations </deliverableObligations> [0..1]
'This element contains all the ISDA terms relevant to defining the deliverable obligations.'

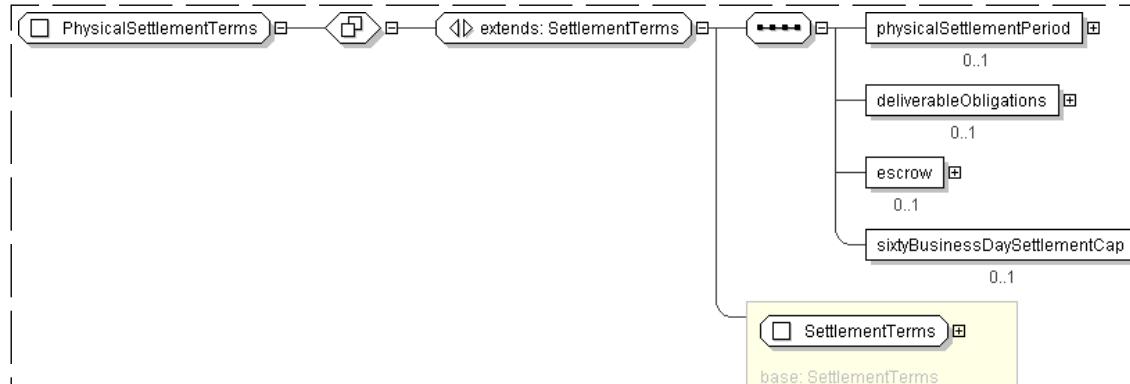
<escrow> xsd:boolean </escrow> [0..1]
'If this element is specified and set to \'true\', indicates that physical settlement must
take place through the use of an escrow agent. (For Canadian counterparties this is
always \'Not Applicable\'. ISDA 2003 Term: Escrow.'

<sixtyBusinessDaySettlementCap> xsd:boolean </sixtyBusinessDaySettlementCap> [0..1]
'If this element is specified and set to \'true\', for a transaction documented under the
2003 ISDA Credit Derivatives Definitions, has the effect of incorporating the language
set forth below into the confirmation. The section references are to the 2003 ISDA
Credit Derivatives Definitions. Notwithstanding Section 1.7 or any provisions of Sections
9.9 or 9.10 to the contrary, but without prejudice to Section 9.3 and (where
applicable) Sections 9.4, 9.5 and 9.6, if the Termination Date has not occurred on or prior
to the date that is 60 Business Days following the Physical Settlement Date, such 60th
Business Day shall be deemed to be the Termination Date with respect to this Transaction
except in relation to any portion of the Transaction (an \"Affected Portion\") in respect
of which: (1) a valid notice of Buy-in Price has been delivered that is effective fewer
than three Business Days prior to such 60th Business Day, in which case the Termination
Date for that Affected Portion shall be the third Business Day following the date on which
such notice is effective; or (2) Buyer has purchased but not Delivered Deliverable
Obligations validly specified by Seller pursuant to Section 9.10(b), in which case
the Termination Date for that Affected Portion shall be the tenth Business Day following
the date on which Seller validly specified such Deliverable Obligations to Buyer.'

<...>

```

## Diagram



## Schema Component Representation

```

<xsd:complexType name="PhysicalSettlementTerms">
  <xsd:complexContent>
    <xsd:extension base=" SettlementTerms ">

```

```

<xsd:sequence>
  <xsd:element name="physicalSettlementPeriod" type=" PhysicalSettlementPeriod " minOccurs="0" />
  <xsd:element name="deliverableObligations" type=" DeliverableObligations " minOccurs="0" />
  <xsd:element name="escrow" type=" xsd:boolean " minOccurs="0" />
  <xsd:element name="sixtyBusinessDaySettlementCap" type=" xsd:boolean " minOccurs="0" />
</xsd:sequence>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>

```

[top](#)

## Complex Type: ProtectionTerms

Super-types:	None
Sub-types:	None

Name	ProtectionTerms
Used by (from the same schema document)	Complex Type <a href="#">CreditDefaultSwap</a>
Abstract	no

### XML Instance Representation

```

<...
  id=" xsd:ID [0..1]">
  <calculationAmount> Money </calculationAmount> [1]
  'The notional amount of protection coverage. ISDA 2003 Term: Floating Rate Payer
  Calculation Amount'

  <creditEvents> CreditEvents </creditEvents> [0..1]
  'This element contains all the ISDA terms relating to credit events.'

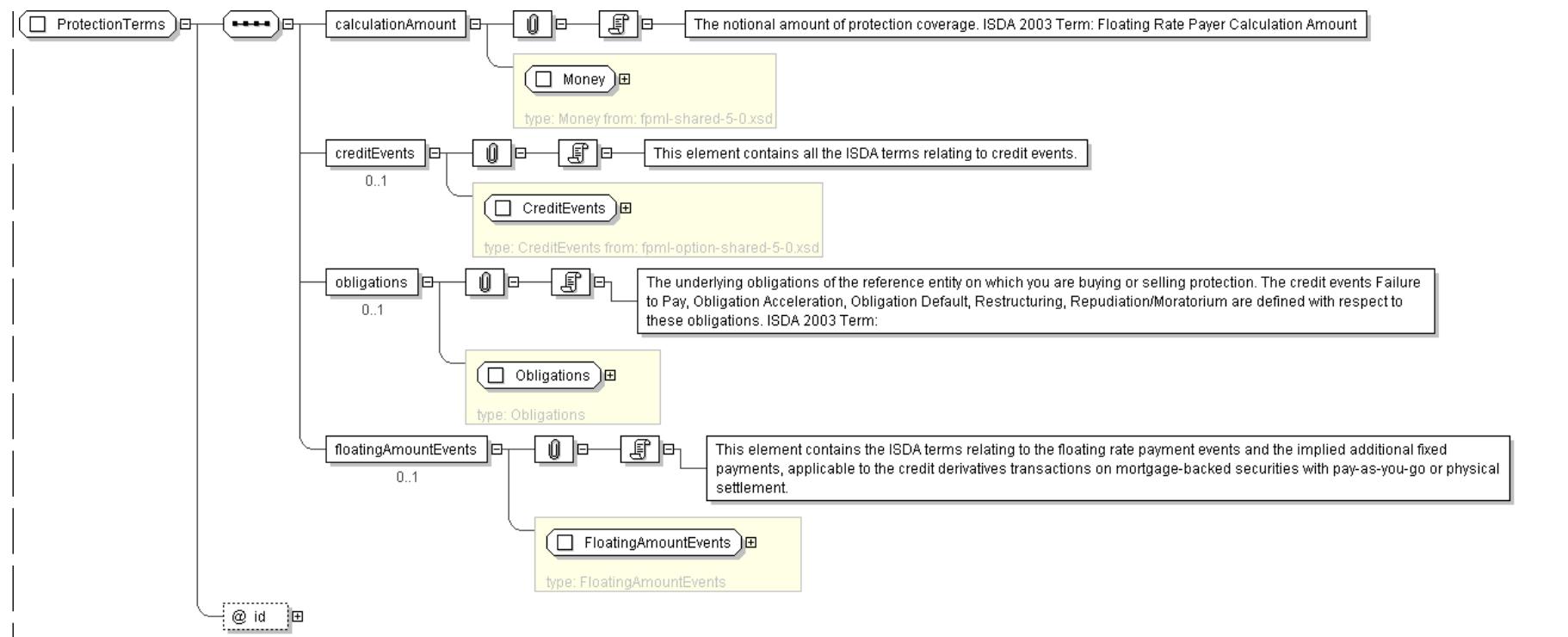
  <obligations> Obligations </obligations> [0..1]
  'The underlying obligations of the reference entity on which you are buying or
  selling protection. The credit events Failure to Pay, Obligation Acceleration,
  Obligation Default, Restructuring, Repudiation/Moratorium are defined with respect to
  these obligations. ISDA 2003 Term:'

  <floatingAmountEvents> FloatingAmountEvents </floatingAmountEvents> [0..1]
  'This element contains the ISDA terms relating to the floating rate payment events and
  the implied additional fixed payments, applicable to the credit derivatives transactions
  on mortgage-backed securities with pay-as-you-go or physical settlement.'

</...>

```

### Diagram

**Schema Component Representation**

```

<xsd:complexType name="ProtectionTerms">
  <xsd:sequence>
    <xsd:element name="calculationAmount" type="Money" />
    <xsd:element name="creditEvents" type="CreditEvents" minOccurs="0"/>
    <xsd:element name="obligations" type="Obligations" minOccurs="0"/>
    <xsd:element name="floatingAmountEvents" type="FloatingAmountEvents" minOccurs="0"/>
  </xsd:sequence>
  <xsd:attribute name="id" type="xsd:ID" use="optional"/>
</xsd:complexType>

```

[top](#)**Complex Type: ProtectionTermsReference**

Super-types:

[Reference](#) < **ProtectionTermsReference** (by extension)

Sub-types:

None

**Name**

ProtectionTermsReference

**Used by (from the same schema document)**Complex Type [ReferencePoolItem](#)**Abstract**

no

**Documentation**

Reference to protectionTerms component.

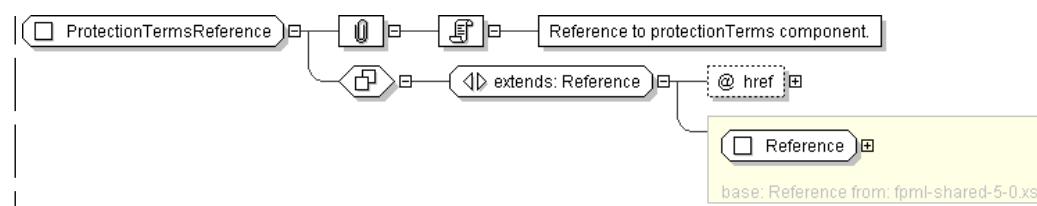
**XML Instance Representation**

```

<...
  href="# xsd:IDREF [1]" />

```

**Diagram**

**Schema Component Representation**

```

<xsd:complexType name="ProtectionTermsReference">
  <xsd:complexContent>
    <xsd:extension base=" Reference ">
      <xsd:attribute name="href" type=" xsd:IDREF " use="required" reference="ProtectionTerms" />
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

top

**Complex Type: ReferenceInformation**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None

<b>Name</b>	ReferenceInformation
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">GeneralTerms</a>
<b>Abstract</b>	no

**XML Instance Representation**

```

<...>
<referenceEntity> LegalEntity </referenceEntity> [1]
'The corporate or sovereign entity on which you are buying or selling protection and
any successor that assumes all or substantially all of its contractual and other
obligations. It is vital to use the correct legal name of the entity and to be careful not
to choose a subsidiary if you really want to trade protection on a parent company. Please
note, Reference Entities cannot be senior or subordinated. It is the obligations of
the Reference Entities that can be senior or subordinated. ISDA 2003 Term: Reference Entity'

```

```

Start Choice [1]
<referenceObligation> ReferenceObligation </referenceObligation> [1..*]
'The Reference Obligation is a financial instrument that is either issued or guaranteed by
the reference entity. It serves to clarify the precise reference entity protection is
being offered upon, and its legal position with regard to other related firms
(parents/subsidiaries). Furthermore the Reference Obligation is ALWAYS deliverable
and establishes the Pari Passu ranking (as the deliverable bonds must rank equal to
the reference obligation). ISDA 2003 Term: Reference Obligation'

```

```

<noReferenceObligation> xsd:boolean </noReferenceObligation> [1]
'Used to indicate that there is no Reference Obligation associated with this Credit
Default Swap and that there will never be one.'

```

```

<unknownReferenceObligation> xsd:boolean </unknownReferenceObligation> [1]
'Used to indicate that the Reference obligation associated with the Credit Default Swap
is currently not known. This is not valid for Legal Confirmation purposes, but is valid
for earlier stages in the trade life cycle (e.g. Broker Confirmation).'

```

```

End Choice
<allGuarantees> xsd:boolean </allGuarantees> [0..1]
'Indicates whether an obligation of the Reference Entity, guaranteed by the Reference Entity
on behalf of a non-Affiliate, is to be considered an Obligation for the purpose of
the transaction. It will be considered an obligation if allGuarantees is applicable (true)
and not if allGuarantees is inapplicable (false). ISDA 2003 Term: All Guarantees'

```

<referencePrice> xsd:decimal </referencePrice> [0..1]

'Used to determine (a) for physically settled trades, the Physical Settlement Amount, which equals the Floating Rate Payer Calculation Amount times the Reference Price and (b) for cash settled trades, the Cash Settlement Amount, which equals the greater of (i) the difference between the Reference Price and the Final Price and (ii) zero. ISDA 2003 Term: Reference Price'

<referencePolicy> xsd:boolean </referencePolicy> [0..1]

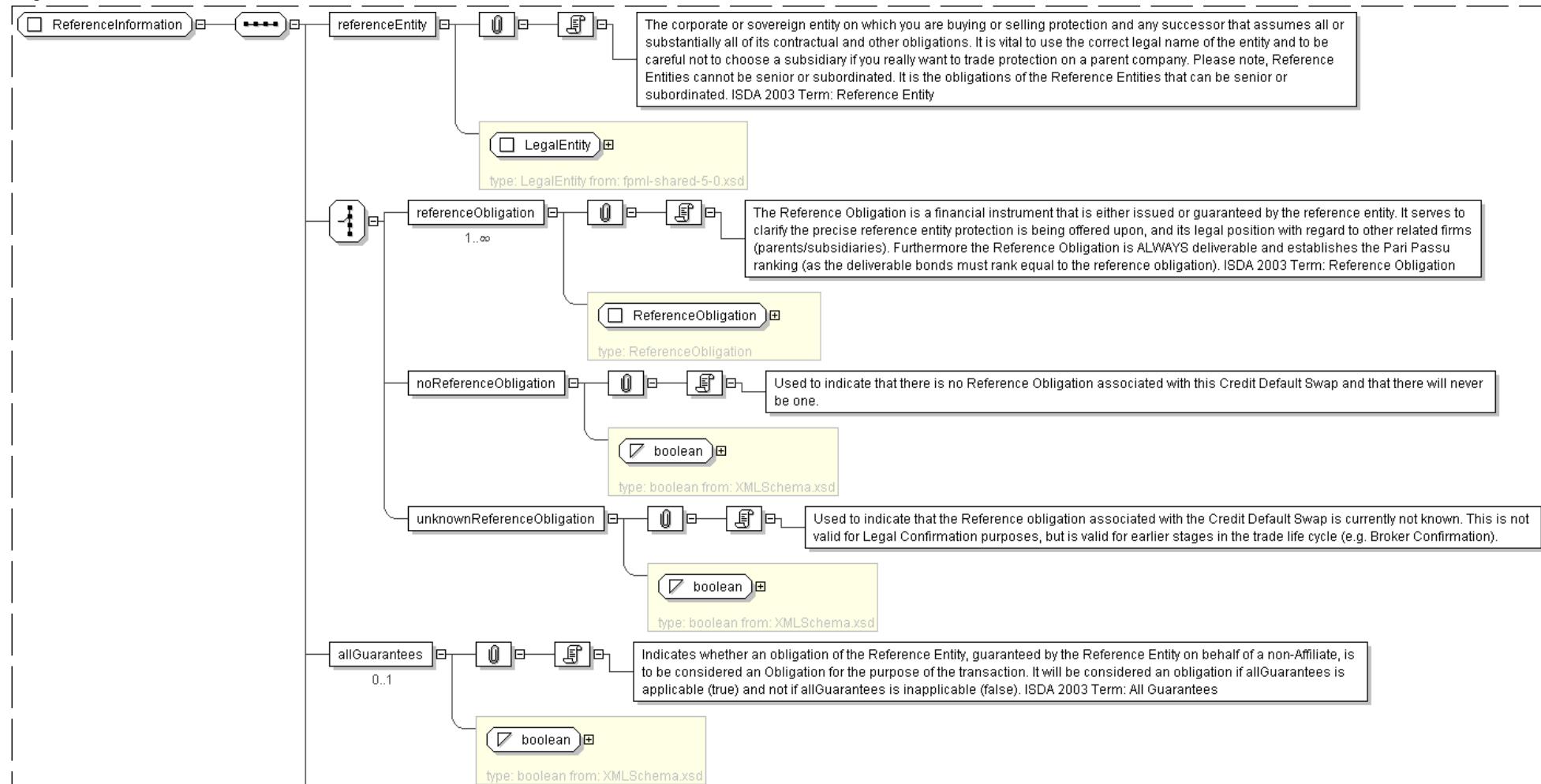
'Applicable to the transactions on mortgage-backed security, which can make use of a reference policy. Presence of the element with value set to '\true\' indicates that the reference policy is applicable; absence implies that it is not.'

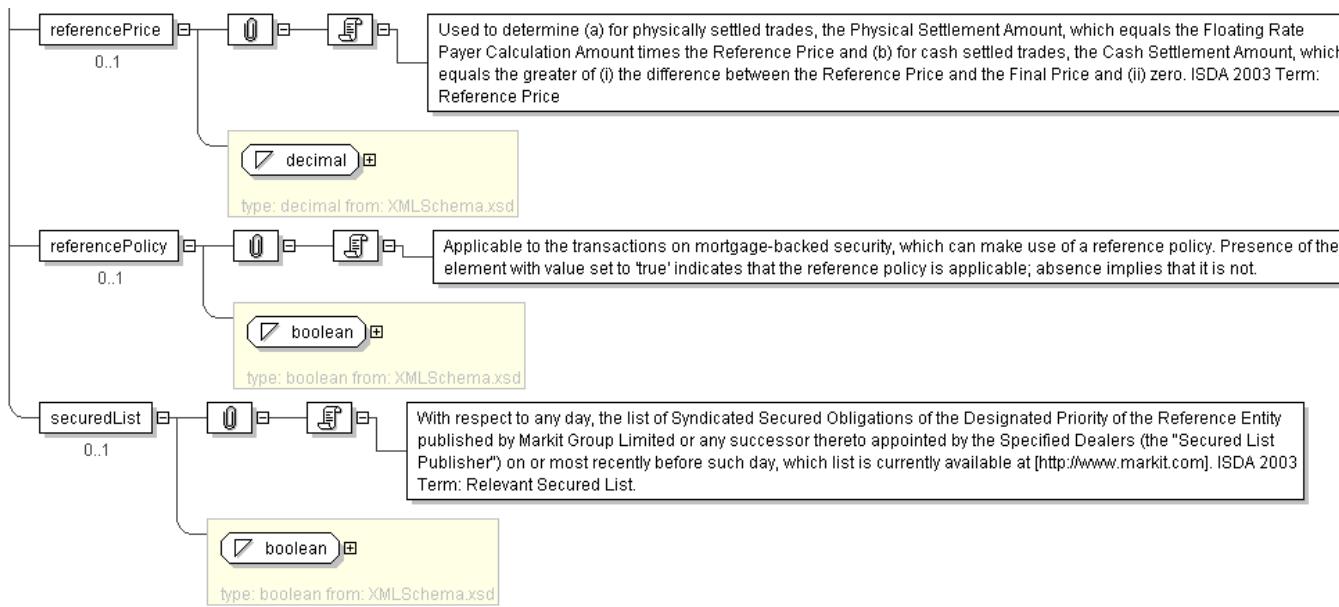
<securedList> xsd:boolean </securedList> [0..1]

'With respect to any day, the list of Syndicated Secured Obligations of the Designated Priority of the Reference Entity published by Markit Group Limited or any successor thereto appointed by the Specified Dealers (the \"Secured List Publisher\") on or most recently before such day, which list is currently available at [http://www.markit.com]. ISDA 2003 Term: Relevant Secured List.'

</...>

#### Diagram





#### Schema Component Representation

```

<xsd:complexType name="ReferenceInformation">
  <xsd:sequence>
    <xsd:element name="referenceEntity" type=" LegalEntity " />
    <xsd:choice>
      <xsd:element name="referenceObligation" type=" ReferenceObligation " maxOccurs="unbounded"/>
      <xsd:element name="noReferenceObligation" type=" xsd:boolean " />
      <xsd:element name="unknownReferenceObligation" type=" xsd:boolean " />
    </xsd:choice>
    <xsd:element name="allGuarantees" type=" xsd:boolean " minOccurs="0"/>
    <xsd:element name="referencePrice" type=" xsd:decimal " minOccurs="0"/>
    <xsd:element name="referencePolicy" type=" xsd:boolean " minOccurs="0"/>
    <xsd:element name="securedList" type=" xsd:boolean " minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>

```

top

#### Complex Type: ReferenceObligation

Super-types:	None
Sub-types:	None

Name	ReferenceObligation
Used by (from the same schema document)	Complex Type <a href="#">ReferenceInformation</a> , Complex Type <a href="#">ReferencePair</a>
Abstract	no

#### XML Instance Representation

```

<...>
Start Choice [1]
  <bond> ... </bond> [1]
  <convertibleBond> ... </convertibleBond> [1]
  <mortgage> ... </mortgage> [1]
  <loan> ... </loan> [1]
End Choice

```

```

Start Choice [0..1]
<primaryObligor> LegalEntity </primaryObligor> [1]
'The entity primarily responsible for repaying debt to a creditor as a result of borrowing or issuing bonds. ISDA 2003 Term: Primary Obligor'

<primaryObligorReference> LegalEntityReference </primaryObligorReference> [1]
'A pointer style reference to a reference entity defined elsewhere in the document. Used when the reference entity is the primary obligor.'

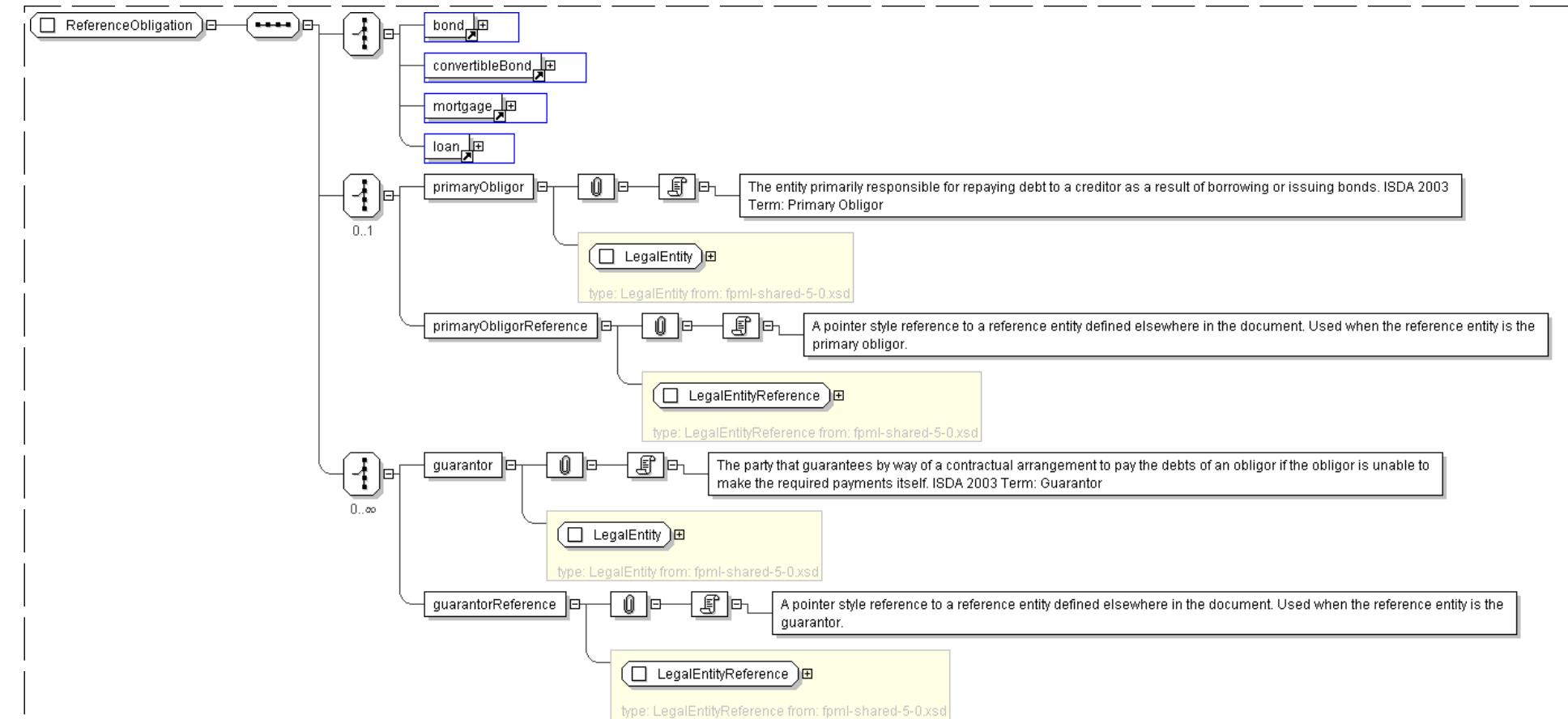
End Choice
Start Choice [0..*]
<guarantor> LegalEntity </guarantor> [1]
'The party that guarantees by way of a contractual arrangement to pay the debts of an obligor if the obligor is unable to make the required payments itself. ISDA 2003 Term: Guarantor'

<guarantorReference> LegalEntityReference </guarantorReference> [1]
'A pointer style reference to a reference entity defined elsewhere in the document. Used when the reference entity is the guarantor.'

```

End Choice  
 </...>

#### Diagram



#### Schema Component Representation

```

<xsd:complexType name="ReferenceObligation">
  <xsd:sequence>
    <xsd:choice>
      <xsd:element ref=" bond " />
      <xsd:element ref=" convertibleBond " />
      <xsd:element ref=" mortgage " />
      <xsd:element ref=" loan " />
    </xsd:choice>
    <xsd:choice minOccurs="0">
      <xsd:element name="primaryObligor" type=" LegalEntity " />
      <xsd:element name="primaryObligorReference" type=" LegalEntityReference " />
    </xsd:choice>
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
      <xsd:element name="guarantor" type=" LegalEntity " />
      <xsd:element name="guarantorReference" type=" LegalEntityReference " />
    </xsd:choice>
  </xsd:sequence>
</xsd:complexType>

```

top

## Complex Type: ReferencePair

Super-types:	None
Sub-types:	None

Name	ReferencePair
Used by (from the same schema document)	Complex Type <a href="#">ReferencePoolItem</a>
Abstract	no

### XML Instance Representation

```

<...>
<referenceEntity> LegalEntity </referenceEntity> [1]
'The corporate or sovereign entity on which you are buying or selling protection and
any successor that assumes all or substantially all of its contractual and other
obligations. It is vital to use the correct legal name of the entity and to be careful not
to choose a subsidiary if you really want to trade protection on a parent company. Please
note, Reference Entities cannot be senior or subordinated. It is the obligations of
the Reference Entities that can be senior or subordinated. ISDA 2003 Term: Reference Entity'

Start Choice [1]
<referenceObligation> ReferenceObligation </referenceObligation> [1]
'The Reference Obligation is a financial instrument that is either issued or guaranteed by
the reference entity. It serves to clarify the precise reference entity protection is
being offered upon, and its legal position with regard to other related firms
(parents/subsidiaries). Furthermore the Reference Obligation is ALWAYS deliverable
and establishes the Pari Passu ranking (as the deliverable bonds must rank equal to
the reference obligation). ISDA 2003 Term: Reference Obligation'

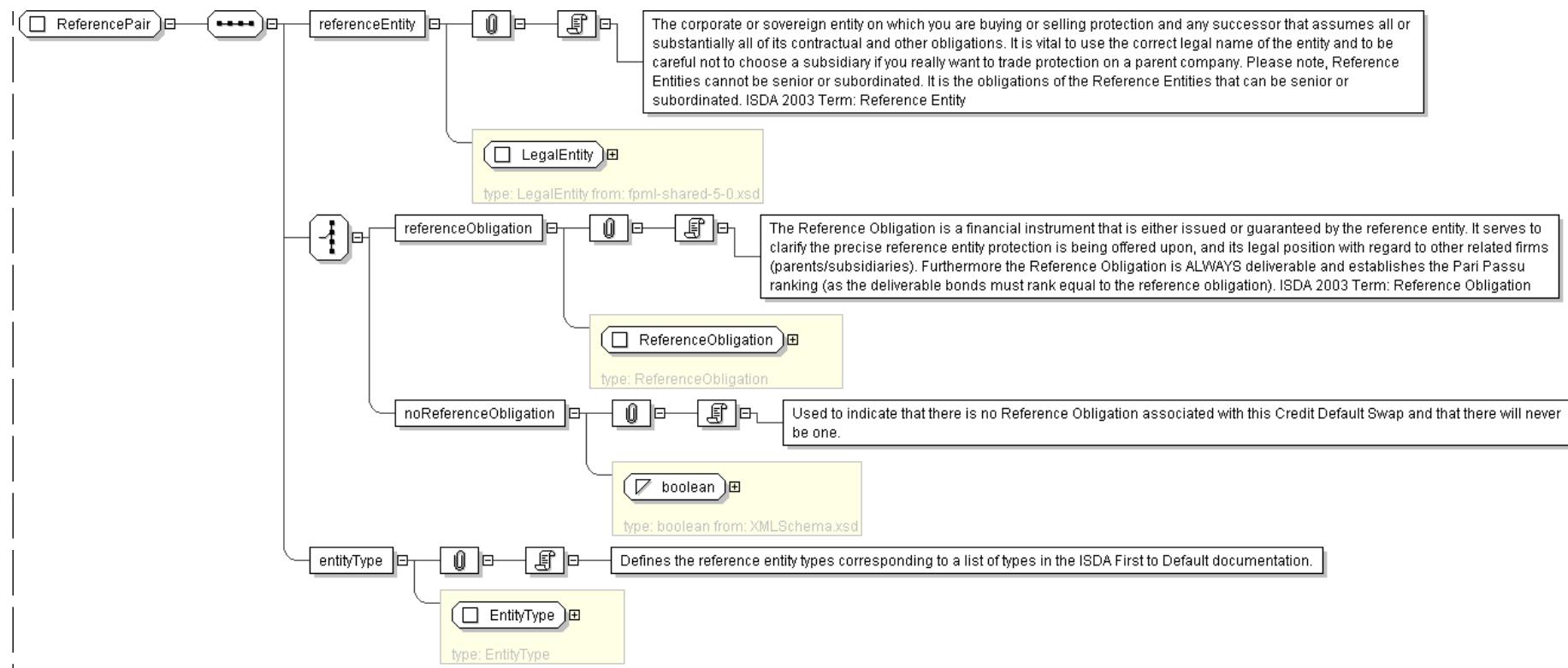
<noReferenceObligation> xsd:boolean </noReferenceObligation> [1]
'Used to indicate that there is no Reference Obligation associated with this Credit
Default Swap and that there will never be one.'

End Choice
<entityType> EntityType </entityType> [1]
'Defines the reference entity types corresponding to a list of types in the ISDA First
to Default documentation.'

</...>

```

### Diagram



#### Schema Component Representation

```

<xsd:complexType name="ReferencePair">
  <xsd:sequence>
    <xsd:element name="referenceEntity" type=" LegalEntity " />
    <xsd:choice>
      <xsd:element name="referenceObligation" type=" ReferenceObligation " />
      <xsd:element name="noReferenceObligation" type=" xsd:boolean " />
    </xsd:choice>
    <xsd:element name="entityType" type=" EntityType " />
  </xsd:sequence>
</xsd:complexType>
  
```

top

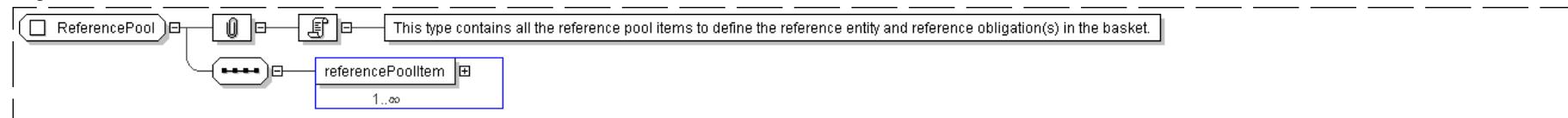
#### Complex Type: ReferencePool

Super-types:	None
Sub-types:	None
Name	ReferencePool
Used by (from the same schema document)	Complex Type <a href="#">BasketReferenceInformation</a>
Abstract	no
Documentation	This type contains all the reference pool items to define the reference entity and reference obligation(s) in the basket.

#### XML Instance Representation

```

<...>
<referencePoolItem> ReferencePoolItem </referencePoolItem> [1..*]
</...>
  
```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="ReferencePool">
  <xsd:sequence>
    <xsd:element name="referencePoolItem" type=" ReferencePoolItem " maxOccurs="unbounded" />
  </xsd:sequence>
</xsd:complexType>
  
```

[top](#)**Complex Type: ReferencePoolItem**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None

<b>Name</b>	ReferencePoolItem
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">ReferencePool</a>
<b>Abstract</b>	no
<b>Documentation</b>	This type contains all the constituent weight and reference information.

**XML Instance Representation**

```

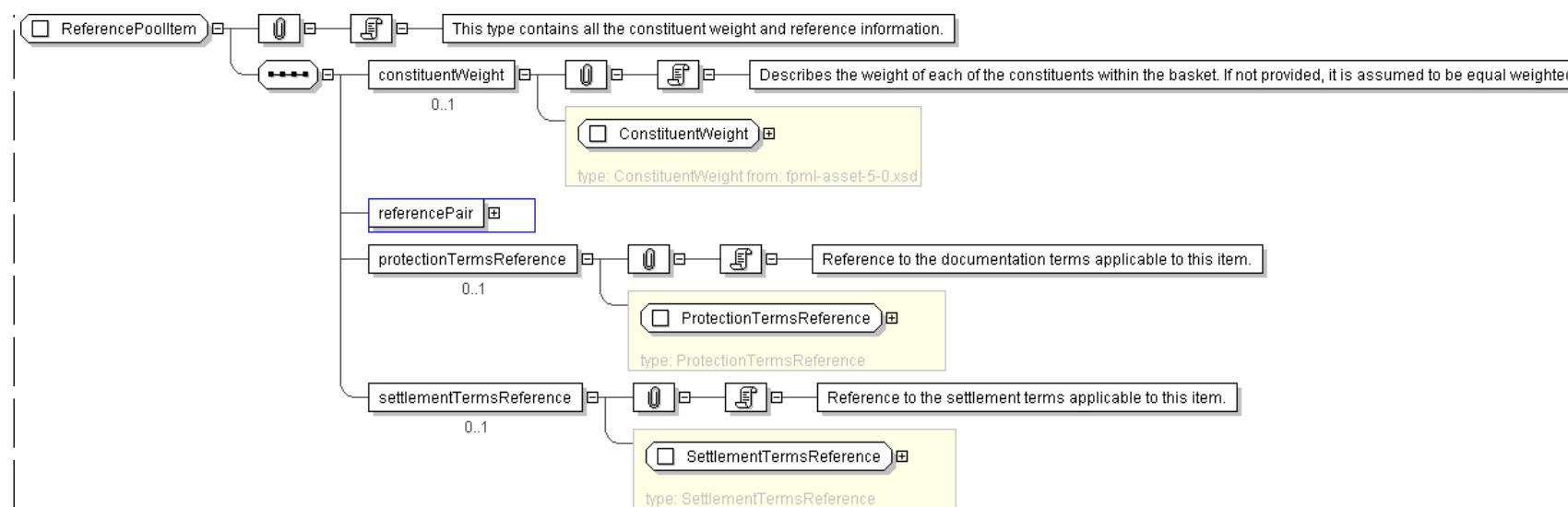
<...>
<constituentWeight> ConstituentWeight </constituentWeight> [0..1]
'Describes the weight of each of the constituents within the basket. If not provided, it
is assumed to be equal weighted.'

<referencePair> ReferencePair </referencePair> [1]
<protectionTermsReference> ProtectionTermsReference </protectionTermsReference> [0..1]
'Reference to the documentation terms applicable to this item.'

<settlementTermsReference> SettlementTermsReference </settlementTermsReference> [0..1]
'Reference to the settlement terms applicable to this item.'

</...>
  
```

**Diagram**

**Schema Component Representation**

```
<xsd:complexType name="ReferencePoolItem">
  <xsd:sequence>
    <xsd:element name="constituentWeight" type="ConstituentWeight" minOccurs="0"/>
    <xsd:element name="referencePair" type="ReferencePair"/>
    <xsd:element name="protectionTermsReference" type="ProtectionTermsReference" minOccurs="0"/>
    <xsd:element name="settlementTermsReference" type="SettlementTermsReference" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>
```

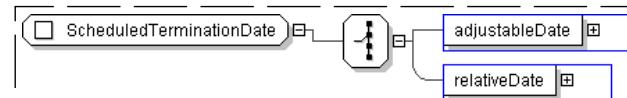
[top](#)**Complex Type: ScheduledTerminationDate**

Super-types:	None
Sub-types:	None

Name	ScheduledTerminationDate
Abstract	no

**XML Instance Representation**

```
<...>
Start Choice [1]
  <adjustableDate> AdjustableDate2 </adjustableDate> [1]
  <relativeDate> Period </relativeDate> [1]
End Choice
<...>
```

**Diagram****Schema Component Representation**

```
<xsd:complexType name="ScheduledTerminationDate">
  <xsd:choice>
```

```

<xsd:element name="adjustableDate" type=" AdjustableDate2 " />
<xsd:element name="relativeDate" type=" Period " />
</xsd:choice>
</xsd:complexType>

```

## Complex Type: SettledEntityMatrix

Super-types:	None
Sub-types:	None

Name	SettledEntityMatrix
Used by (from the same schema document)	Complex Type <a href="#">IndexReferenceInformation</a>
Abstract	no

### XML Instance Representation

```

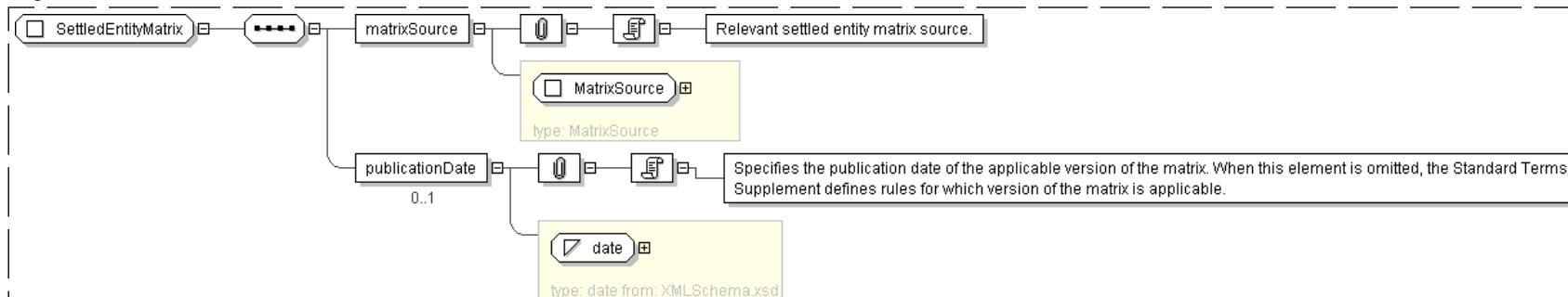
<...>
<matrixSource> MatrixSource </matrixSource> [1]
'Relevant settled entity matrix source.'

<publicationDate> xsd:date </publicationDate> [0..1]
'Specifies the publication date of the applicable version of the matrix. When this element
is omitted, the Standard Terms Supplement defines rules for which version of the matrix
is applicable.'

</...>

```

### Diagram



### Schema Component Representation

```

<xsd:complexType name="SettledEntityMatrix">
  <xsd:sequence>
    <xsd:element name="matrixSource" type=" MatrixSource " />
    <xsd:element name="publicationDate" type=" xsd:date " minOccurs="0 " />
  </xsd:sequence>
</xsd:complexType>

```

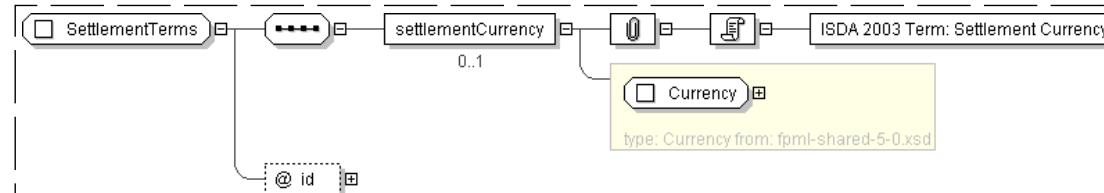
## Complex Type: SettlementTerms

Super-types:	None
Sub-types:	<ul style="list-style-type: none"> <li><a href="#">CashSettlementTerms</a> (by extension)</li> <li><a href="#">PhysicalSettlementTerms</a> (by extension)</li> </ul>

Name	SettlementTerms
Abstract	no

**XML Instance Representation**

```
<...  
id=" xsd:ID [0..1]">  
<settlementCurrency> Currency </settlementCurrency> [0..1]  
'ISDA 2003 Term: Settlement Currency'  
<...>
```

**Diagram****Schema Component Representation**

```
<xsd:complexType name="SettlementTerms">  
  <xsd:sequence>  
    <xsd:element name="settlementCurrency" type=" Currency " minOccurs="0" />  
  </xsd:sequence>  
  <xsd:attribute name="id" type=" xsd:ID " use="optional" />  
</xsd:complexType>
```

top

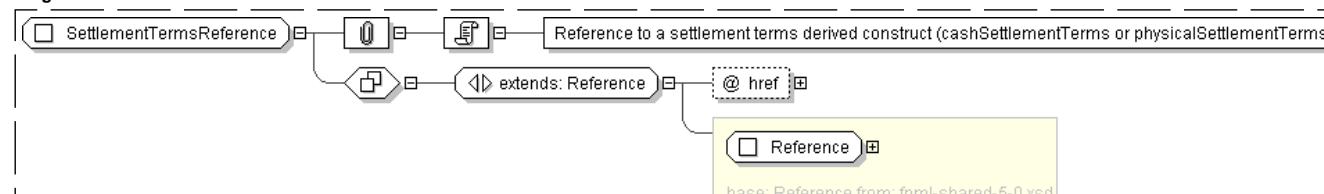
**Complex Type: SettlementTermsReference**

Super-types:	<a href="#">Reference</a> < <b>SettlementTermsReference</b> (by extension)
Sub-types:	None

Name	SettlementTermsReference
Used by (from the same schema document)	Complex Type <a href="#">ReferencePoolItem</a>
Abstract	no
Documentation	Reference to a settlement terms derived construct (cashSettlementTerms or physicalSettlementTerms).

**XML Instance Representation**

```
<...  
href=" xsd:IDREF [1]" />
```

**Diagram****Schema Component Representation**

```
<xsd:complexType name="SettlementTermsReference">  
  <xsd:complexContent>
```

```

<xsd:extension base=" Reference ">
  <xsd:attribute name="href" type=" xsd:IDREF" use="required" reference="SettlementTerms"/>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>

```

## Complex Type: [SinglePayment](#)

Super-types:	<a href="#">PaymentBase</a> < <b>SinglePayment</b> (by extension)
Sub-types:	None

Name	SinglePayment
Used by (from the same schema document)	Complex Type <a href="#">FeeLeg</a>
Abstract	no

### XML Instance Representation

```

<...>
<id=" xsd:ID [0..1]">
<adjustablePaymentDate> xsd:date </adjustablePaymentDate> [1]
'A fixed amount payment date that shall be subject to adjustment in accordance with
the applicable business day convention if it would otherwise fall on a day that is not
a business day. The applicable business day convention and business day are those specified
in the dateAdjustments element within the generalTerms component. ISDA 2003 Term: Fixed
Rate Payer Payment Date'

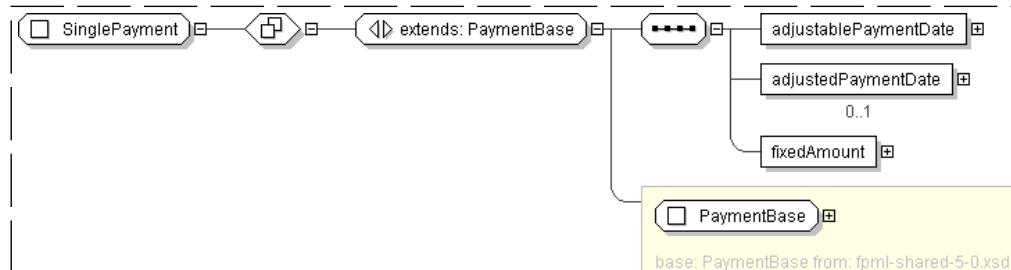
<adjustedPaymentDate> xsd:date </adjustedPaymentDate> [0..1]
'The adjusted payment date. This date should already be adjusted for any applicable
business day convention. This component is not intended for use in trade confirmation but
may be specified to allow the fee structure to also serve as a cashflow type component.'

<fixedAmount> Money </fixedAmount> [1]
'A fixed payment amount. ISDA 2003 Term: Fixed Amount'

</...>

```

### Diagram



### Schema Component Representation

```

<xsd:complexType name="SinglePayment">
<xsd:complexContent>
  <xsd:extension base=" PaymentBase ">
    <xsd:sequence>
      <xsd:element name="adjustablePaymentDate" type=" xsd:date "/>
      <xsd:element name="adjustedPaymentDate" type=" xsd:date " minOccurs="0"/>
      <xsd:element name="fixedAmount" type=" Money "/>
    </xsd:sequence>
  </xsd:extension>
</xsd:complexContent>
</xsd:complexType>

```

## Complex Type: SingleValuationDate

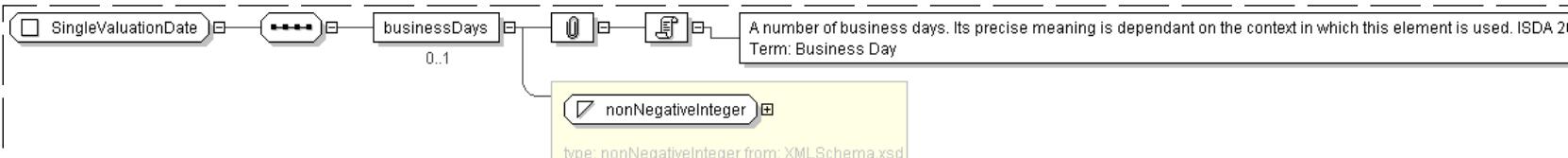
Super-types:	None
Sub-types:	<ul style="list-style-type: none"> <li>MultipleValuationDates (by extension)</li> </ul>

Name	SingleValuationDate
Used by (from the same schema document)	Complex Type <a href="#">ValuationDate</a>
Abstract	no

### XML Instance Representation

```
<...>
<businessDays> xsd:nonNegativeInteger </businessDays> [0..1]
'A number of business days. Its precise meaning is dependant on the context in which
this element is used. ISDA 2003 Term: Business Day'
</...>
```

### Diagram



### Schema Component Representation

```
<xsd:complexType name="SingleValuationDate">
  <xsd:sequence>
    <xsd:element name="businessDays" type="xsd:nonNegativeInteger" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>
```

## Complex Type: SpecifiedCurrency

Super-types:	None
Sub-types:	None

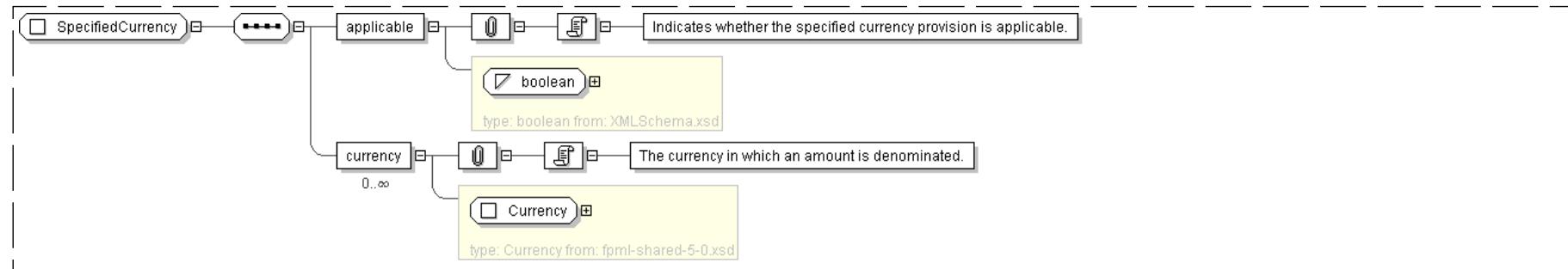
Name	SpecifiedCurrency
Used by (from the same schema document)	Complex Type <a href="#">DeliverableObligations</a> , Complex Type <a href="#">Obligations</a>
Abstract	no

### XML Instance Representation

```
<...>
<applicable> xsd:boolean </applicable> [1]
'Indicates whether the specified currency provision is applicable.'

<currency> Currency </currency> [0..*]
'The currency in which an amount is denominated.'

</...>
```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="SpecifiedCurrency">
  <xsd:sequence>
    <xsd:element name="applicable" type="xsd:boolean" />
    <xsd:element name="currency" type="Currency" minOccurs="0" maxOccurs="unbounded" />
  </xsd:sequence>
</xsd:complexType>
  
```

top

**Complex Type: Tranche**

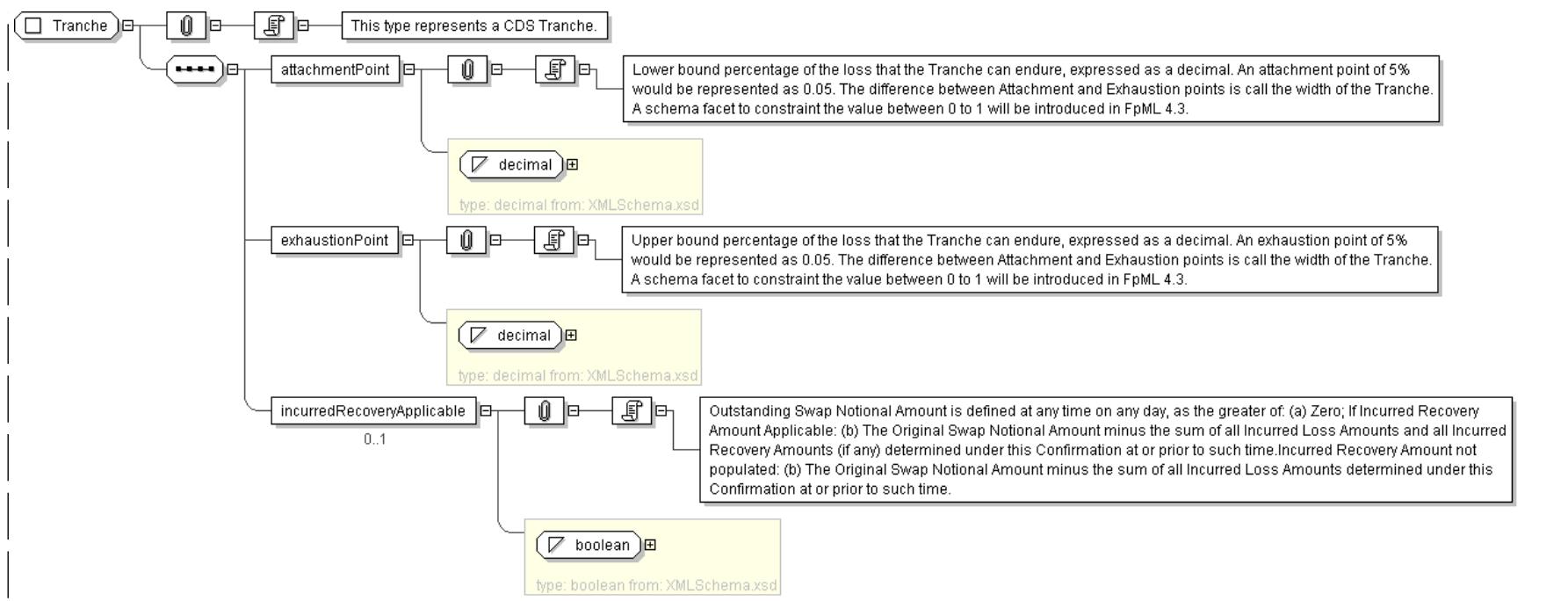
<b>Super-types:</b>	None
<b>Sub-types:</b>	None
<b>Name</b>	Tranche
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">BasketReferenceInformation</a> , Complex Type <a href="#">IndexReferenceInformation</a>
<b>Abstract</b>	no
<b>Documentation</b>	This type represents a CDS Tranche.

**XML Instance Representation**

```

<...>
<attachmentPoint> xsd:decimal </attachmentPoint> [1]
'Lower bound percentage of the loss that the Tranche can endure, expressed as a decimal.
An attachment point of 5% would be represented as 0.05. The difference between Attachment
and Exhaustion points is call the width of the Tranche. A schema facet to constraint the
value between 0 to 1 will be introduced in FpML 4.3.'
<exhaustionPoint> xsd:decimal </exhaustionPoint> [1]
'Upper bound percentage of the loss that the Tranche can endure, expressed as a decimal.
An exhaustion point of 5% would be represented as 0.05. The difference between Attachment
and Exhaustion points is call the width of the Tranche. A schema facet to constraint the
value between 0 to 1 will be introduced in FpML 4.3.'
<incurredRecoveryApplicable> xsd:boolean </incurredRecoveryApplicable> [0..1]
'Outstanding Swap Notional Amount is defined at any time on any day, as the greater of:
(a) Zero; If Incurred Recovery Amount Applicable: (b) The Original Swap Notional Amount
minus the sum of all Incurred Loss Amounts and all Incurred Recovery Amounts (if
any) determined under this Confirmation at or prior to such time.Incurred Recovery Amount
not populated: (b) The Original Swap Notional Amount minus the sum of all Incurred Loss
Amounts determined under this Confirmation at or prior to such time.'
</...>
  
```

**Diagram**



#### Schema Component Representation

```
<xsd:complexType name="Tranche">
  <xsd:sequence>
    <xsd:element name="attachmentPoint" type="xsd:decimal" />
    <xsd:element name="exhaustionPoint" type="xsd:decimal" />
    <xsd:element name="incurredRecoveryApplicable" type="xsd:boolean" minOccurs="0" />
  </xsd:sequence>
</xsd:complexType>
```

top

#### Complex Type: ValuationDate

Super-types:	None
Sub-types:	None

Name	ValuationDate
Used by (from the same schema document)	Complex Type <a href="#">CashSettlementTerms</a>
Abstract	no

#### XML Instance Representation

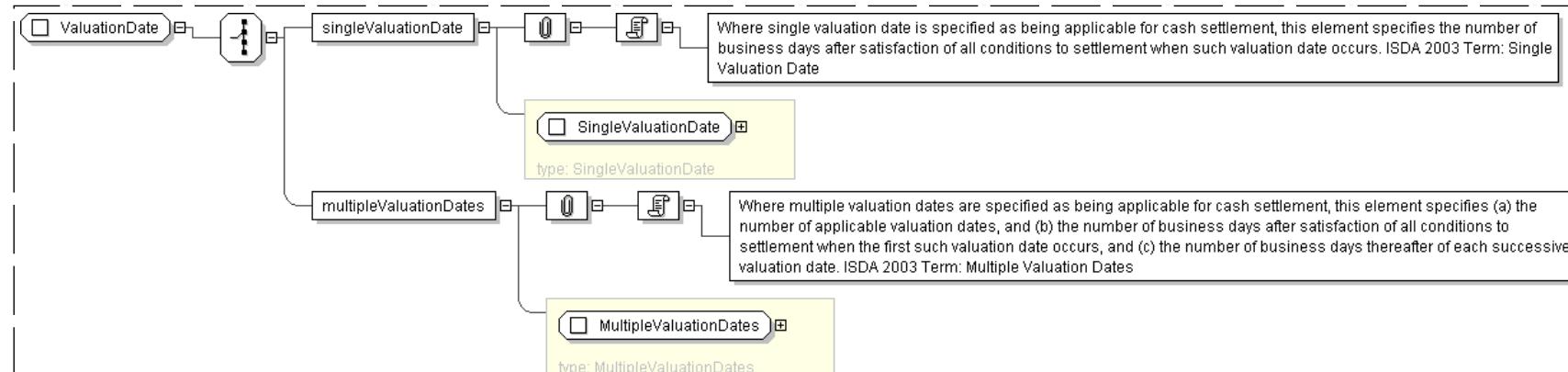
```
<...>
Start Choice [1]
<singleValuationDate> singleValuationDate </singleValuationDate> [1]
'Where single valuation date is specified as being applicable for cash settlement, this
element specifies the number of business days after satisfaction of all conditions
to settlement when such valuation date occurs. ISDA 2003 Term: Single Valuation Date'

<multipleValuationDates> MultipleValuationDates </multipleValuationDates> [1]
'Where multiple valuation dates are specified as being applicable for cash settlement,
this element specifies (a) the number of applicable valuation dates, and (b) the number
of business days after satisfaction of all conditions to settlement when the first
such valuation date occurs, and (c) the number of business days thereafter of each'
```

successive valuation date. ISDA 2003 Term: Multiple Valuation Dates'

End Choice  
</...>

## Diagram



## Schema Component Representation

```

<xsd:complexType name="ValuationDate">
  <xsd:choice>
    <xsd:element name="singleValuationDate" type=" SingleValuationDate " />
    <xsd:element name="multipleValuationDates" type=" MultipleValuationDates " />
  </xsd:choice>
</xsd:complexType>

```

top

## Model Group: FixedRecovery.model

Name	FixedRecovery.model
Used by (from the same schema document)	Complex Type <b>CashSettlementTerms</b>

## XML Instance Representation

```

Start Choice [1]
<cashSettlementAmount> Money </cashSettlementAmount> [1]

'The amount paid by the seller to the buyer for cash settlement on the cash settlement date.
If not otherwise specified, would typically be calculated as 100 (or the Reference Price)
minus the price of the Reference Obligation (all expressed as a percentage) times Floating
Rate Payer Calculation Amount. ISDA 2003 Term: Cash Settlement Amount.'

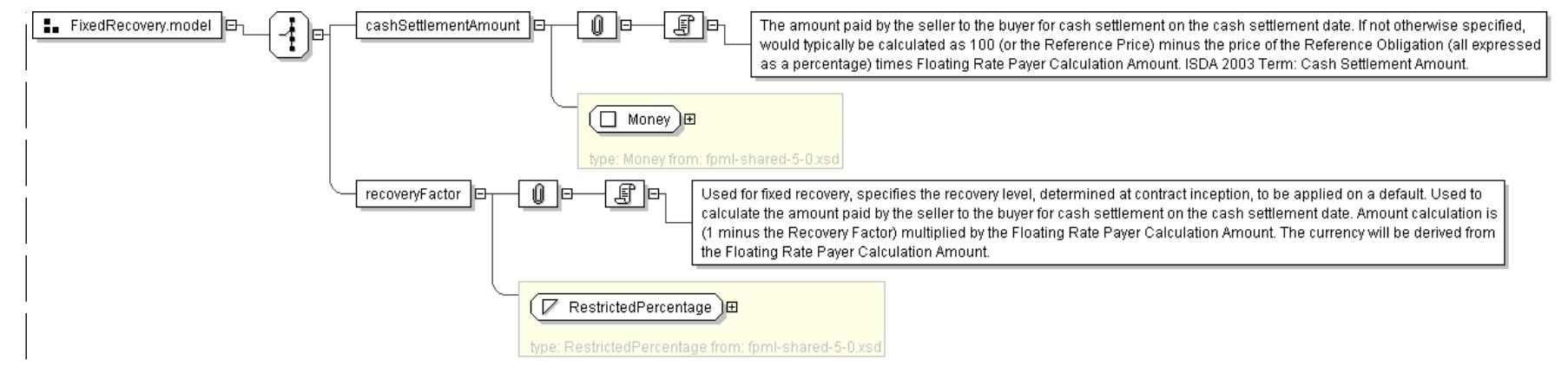
<recoveryFactor> RestrictedPercentage </recoveryFactor> [1]

'Used for fixed recovery, specifies the recovery level, determined at contract inception, to
be applied on a default. Used to calculate the amount paid by the seller to the buyer for
cash settlement on the cash settlement date. Amount calculation is (1 minus the
Recovery Factor) multiplied by the Floating Rate Payer Calculation Amount. The currency will
be derived from the Floating Rate Payer Calculation Amount.'

```

End Choice

## Diagram

**Schema Component Representation**

```
<xsd:group name="FixedRecovery.model">
  <xsd:choice>
    <xsd:element name="cashSettlementAmount" type="Money" />
    <xsd:element name="recoveryFactor" type="RestrictedPercentage" />
  </xsd:choice>
</xsd:group>
```

[top](#)**Legend**

**Complex Type:**  
Schema Component Type

**AusAddress**  
Schema Component Name

<b>Super-types:</b>	<a href="#">Address</a> < AusAddress (by extension)
<b>Sub-types:</b>	• <a href="#">OLDAddress</a> (by restriction)

If this schema component is a type definition, its type hierarchy is shown in a gray-bordered box.

<b>Name</b>	AusAddress
<b>Abstract</b>	no

The table above displays the properties of this schema component.

**XML Instance Representation**

```
<... country="Australia" >
<unitNo> string </unitNo> [0..1]
<houseNo> string </houseNo> [1]
<street> string </street> [1]
Start Choice [1]
<city> string </city> [1]
<town> string </town> [1]
End Choice
<state> AusStates </state> [1]
<postcode> string <>pattern = [1-9][0-9]{3}<> </postcode> [1]
</...>
```

The XML Instance Representation table above shows the schema component's content as an XML instance.

- The minimum and maximum occurrence of elements and attributes are provided in square brackets, e.g. [0..1].
- Model group information are shown in gray, e.g. Start Choice ... End Choice.
- For type derivations, the elements and attributes that have been added to or changed from the base type's content are shown in **bold**.

- If an element/attribute has a fixed value, the fixed value is shown in green, e.g. country="Australia".
- Otherwise, the type of the element/attribute is displayed.
  - If the element/attribute's type is in the schema, a link is provided to it.
  - For local simple type definitions, the constraints are displayed in angle brackets, e.g. <>pattern = [1-9][0-9]{3}</>.

## Schema Component Representation

```
<complexType name="AusAddress">
<complexContent>
<extension base=" Address ">
<sequence>
<element name="state" type=" AusStates " />
<element name="postcode">
<simpleType>
<restriction base=" string ">
<pattern value="[1-9][0-9]{3}" />
</restriction>
</simpleType>
</element>
</sequence>
<attribute name="country" type=" string " fixed="Australia" />
</extension>
</complexContent>
</complexType>
```

The Schema Component Representation table above displays the underlying XML representation of the schema component. (Annotations are not shown.)

[top](#)

## Glossary

**Abstract** (Applies to complex type definitions and element declarations). An abstract element or complex type cannot be used to validate an element instance. If there is a reference to an abstract element, only element declarations that can substitute the abstract element can be used to validate the instance. For references to abstract type definitions, only derived types can be used.

**All Model Group** Child elements can be provided *in any order* in instances. See: <http://www.w3.org/TR/xmlschema-1/#element-all>.

**Choice Model Group** Only one from the list of child elements and model groups can be provided in instances. See: <http://www.w3.org/TR/xmlschema-1/#element-choice>.

**Collapse Whitespace Policy** Replace tab, line feed, and carriage return characters with space character (Unicode character 32). Then, collapse contiguous sequences of space characters into single space character, and remove leading and trailing space characters.

**Disallowed Substitutions** (Applies to element declarations). If *substitution* is specified, then *substitution group* members cannot be used in place of the given element declaration to validate element instances. If *derivation methods*, e.g. extension, restriction, are specified, then the given element declaration will not validate element instances that have types derived from the element declaration's type using the specified derivation methods. Normally, element instances can override their declaration's type by specifying an *xsi:type* attribute.

**Key Constraint** Like [Uniqueness Constraint](#), but additionally requires that the specified value(s) must be provided. See: [http://www.w3.org/TR/xmlschema-1/#clidentity-constraint\\_Definitions](http://www.w3.org/TR/xmlschema-1/#clidentity-constraint_Definitions).

**Key Reference Constraint** Ensures that the specified value(s) must match value(s) from a [Key Constraint](#) or [Uniqueness Constraint](#). See: [http://www.w3.org/TR/xmlschema-1/#clidentity-constraint\\_Definitions](http://www.w3.org/TR/xmlschema-1/#clidentity-constraint_Definitions).

**Model Group** Groups together element content, specifying the order in which the element content can occur and the number of times the group of element content may be repeated. See: [http://www.w3.org/TR/xmlschema-1/#Model\\_Groups](http://www.w3.org/TR/xmlschema-1/#Model_Groups).

**Nillable** (Applies to element declarations). If an element declaration is nillable, instances can use the *xsi:nil* attribute. The *xsi:nil* attribute is the boolean attribute, *nil*, from the <http://www.w3.org/2001/XMLSchema-instance> namespace. If an element instance has an *xsi:nil* attribute set to true, it can be left empty, even though its element declaration may have required content.

**Notation** A notation is used to identify the format of a piece of data. Values of elements and attributes that are of type, NOTATION, must come from the names of declared notations. See: [http://www.w3.org/TR/xmlschema-1/#cNotation\\_Declarations](http://www.w3.org/TR/xmlschema-1/#cNotation_Declarations).

**Preserve Whitespace Policy** Preserve whitespaces exactly as they appear in instances.

**Prohibited Derivations** (Applies to type definitions). Derivation methods that cannot be used to create sub-types from a given type definition.

**Prohibited Substitutions** (Applies to complex type definitions). Prevents sub-types that have been derived using the specified derivation methods from validating element instances in place of the given type definition.

**Replace Whitespace Policy** Replace tab, line feed, and carriage return characters with space character (Unicode character 32).

**Sequence Model Group** Child elements and model groups must be provided *in the specified order* in instances. See: <http://www.w3.org/TR/xmlschema-1/#element-sequence>.

**Substitution Group** Elements that are *members* of a substitution group can be used wherever the *head* element of the substitution group is referenced.

**Substitution Group Exclusions** (Applies to element declarations). Prohibits element declarations from nominating themselves as being able to substitute a given element declaration, if they have types that are derived from the original element's type using the specified derivation methods.

**Target Namespace** The target namespace identifies the namespace that components in this schema belongs to. If no target namespace is provided, then the schema components do not belong to any namespace.

**Uniqueness Constraint** Ensures uniqueness of an element/attribute value, or a combination of values, within a specified scope. See: [http://www.w3.org/TR/xmlschema-1/#clentity-constraint\\_Definitions](http://www.w3.org/TR/xmlschema-1/#clentity-constraint_Definitions).

[top](#)

---

Generated by [eXtensible XML Editor](#) using a modified version of [xs3p](#) that adds schema diagrams and chunking support.

# XML Schema Documentation

## Table of Contents

- [Schema Document Properties](#)
- [Global Declarations](#)
  - [Element: additionalCommodityForwardLeg](#)
  - [Element: additionalCommoditySwapLeg](#)
  - [Element: commodityForward](#)
  - [Element: commodityOption](#)
  - [Element: commoditySwap](#)
- [Global Definitions](#)
  - [Complex Type: AbsoluteTolerance](#)
  - [Complex Type: BullionDeliveryLocation](#)
  - [Complex Type: BullionPhysicalLeg](#)
  - [Complex Type: CalculationPeriodsDatesReference](#)
  - [Complex Type: CalculationPeriodsReference](#)
  - [Complex Type: CalculationPeriodsScheduleReference](#)
  - [Complex Type: CoalAttributeDecimal](#)
  - [Complex Type: CoalAttributePercentage](#)
  - [Complex Type: CoalDelivery](#)
  - [Complex Type: CoalDeliveryPoint](#)
  - [Complex Type: CoalPhysicalLeg](#)
  - [Complex Type: CoalProduct](#)
  - [Complex Type: CoalProductSource](#)
  - [Complex Type: CoalProductSpecifications](#)
  - [Complex Type: CoalProductType](#)
  - [Complex Type: CoalQualityAdjustments](#)
  - [Complex Type: CoalStandardQuality](#)
  - [Complex Type: CoalStandardQualitySchedule](#)
  - [Complex Type: CoalTransportationEquipment](#)
  - [Complex Type: CommodityAmericanExercise](#)
  - [Complex Type: CommodityCalculationPeriodsSchedule](#)
  - [Complex Type: CommodityDeliveryPeriods](#)
  - [Complex Type: CommodityDeliveryPoint](#)
  - [Complex Type: CommodityDeliveryRisk](#)
  - [Complex Type: CommodityEuropeanExercise](#)
  - [Complex Type: CommodityExercise](#)
  - [Complex Type: CommodityExpireRelativeToEvent](#)
  - [Complex Type: CommodityFixedPriceSchedule](#)
  - [Complex Type: CommodityForward](#)
  - [Complex Type: CommodityFrequencyType](#)
  - [Complex Type: CommodityFx](#)
  - [Complex Type: CommodityFxType](#)
  - [Complex Type: CommodityHub](#)
  - [Complex Type: CommodityHubCode](#)
  - [Complex Type: CommodityMarketDisruption](#)
  - [Complex Type: CommodityMultipleExercise](#)
  - [Complex Type: CommodityNotionalQuantity](#)
  - [Complex Type: CommodityNotionalQuantitySchedule](#)
  - [Complex Type: CommodityOption](#)
  - [Complex Type: CommodityPayRelativeToEvent](#)
  - [Complex Type: CommodityPhysicalAmericanExercise](#)
  - [Complex Type: CommodityPhysicalEuropeanExercise](#)
  - [Complex Type: CommodityPhysicalExercise](#)
  - [Complex Type: CommodityPhysicalQuantity](#)
  - [Complex Type: CommodityPhysicalQuantityBase](#)
  - [Complex Type: CommodityPhysicalQuantitySchedule](#)
  - [Complex Type: CommodityPipeline](#)
  - [Complex Type: CommodityPipelineCycle](#)
  - [Complex Type: CommodityPremium](#)
  - [Complex Type: CommodityPricingDates](#)
  - [Complex Type: CommodityProductGrade](#)
  - [Complex Type: CommodityQuantityFrequency](#)
  - [Complex Type: CommodityRelativeExpirationDates](#)
  - [Complex Type: CommodityRelativePaymentDates](#)
  - [Complex Type: CommoditySettlementPeriodsNotionalQuantity](#)
  - [Complex Type: CommoditySettlementPeriodsNotionalQuantitySchedule](#)

- [Complex Type: CommoditySettlementPeriodsPriceSchedule](#)
- [Complex Type: CommoditySpreadSchedule](#)
- [Complex Type: CommodityStrikeSchedule](#)
- [Complex Type: CommoditySwap](#)
- [Complex Type: DisruptionFallback](#)
- [Complex Type: ElectricityDelivery](#)
- [Complex Type: ElectricityDeliveryFirm](#)
- [Complex Type: ElectricityDeliveryPeriods](#)
- [Complex Type: ElectricityDeliveryPoint](#)
- [Complex Type: ElectricityDeliverySystemFirm](#)
- [Complex Type: ElectricityDeliveryUnitFirm](#)
- [Complex Type: ElectricityPhysicalDeliveryQuantity](#)
- [Complex Type: ElectricityPhysicalDeliveryQuantitySchedule](#)
- [Complex Type: ElectricityPhysicalLeg](#)
- [Complex Type: ElectricityPhysicalQuantity](#)
- [Complex Type: ElectricityProduct](#)
- [Complex Type: ElectricityTransmissionContingency](#)
- [Complex Type: ElectricityTransmissionContingencyType](#)
- [Complex Type: FixedPrice](#)
- [Complex Type: FixedPriceLeg](#)
- [Complex Type: FloatingLegCalculation](#)
- [Complex Type: FloatingPriceLeg](#)
- [Complex Type: GasDelivery](#)
- [Complex Type: GasDeliveryPeriods](#)
- [Complex Type: GasDeliveryPoint](#)
- [Complex Type: GasPhysicalLeg](#)
- [Complex Type: GasPhysicalQuantity](#)
- [Complex Type: GasProduct](#)
- [Complex Type: GasQuality](#)
- [Complex Type: Lag](#)
- [Complex Type: LagReference](#)
- [Complex Type: MarketDisruptionEvent](#)
- [Complex Type: NonPeriodicFixedPriceLeg](#)
- [Complex Type: OilDelivery](#)
- [Complex Type: OilPhysicalLeg](#)
- [Complex Type: OilPipelineDelivery](#)
- [Complex Type: OilProduct](#)
- [Complex Type: OilProductType](#)
- [Complex Type: OilTransferDelivery](#)
- [Complex Type: PercentageTolerance](#)
- [Complex Type: PhysicalLeg](#)
- [Complex Type: QuantityReference](#)
- [Complex Type: QuantityScheduleReference](#)
- [Complex Type: SequencedDisruptionFallback](#)
- [Complex Type: SettlementPeriods](#)
- [Complex Type: SettlementPeriodsFixedPrice](#)
- [Complex Type: SettlementPeriodsReference](#)
- [Complex Type: SettlementPeriodsSchedule](#)
- [Complex Type: SettlementPeriodsStep](#)
- [Complex Type: UnitQuantity](#)
- Model Group: [CommodityAsian.model](#)
- Model Group: [CommodityCalculationPeriods.model](#)
- Model Group: [CommodityCalculationPeriodsPointer.model](#)
- Model Group: [CommodityCoalComposition.model](#)
- Model Group: [CommodityCoalProperties.model](#)
- Model Group: [CommodityCoalReducingAtmosphere.model](#)
- Model Group: [CommodityContent.model](#)
- Model Group: [CommodityDeliveryPeriodsPointer.model](#)
- Model Group: [CommodityDeliveryPoints.model](#)
- Model Group: [CommodityFinancialOption.model](#)
- Model Group: [CommodityFixedPhysicalQuantity.model](#)
- Model Group: [CommodityFixedPrice.model](#)
- Model Group: [CommodityFreightFlatRate.model](#)
- Model Group: [CommodityNonPeriodicPaymentDates.model](#)
- Model Group: [CommodityNotionalQuantity.model](#)
- Model Group: [CommodityPaymentDates.model](#)
- Model Group: [CommodityPhysicalOption.model](#)
- Model Group: [CommodityStrikePrice.model](#)
- Model Group: [CommodityUSCoalDelivery.model](#)
- Model Group: [CommodityUSCoalProduct.model](#)
- Model Group: [LagOrReference.model](#)

- Model Group: [Price.model](#)
- Model Group: [PricingDays.model](#)
- [Legend](#)
- [Glossary](#)

## Schema Document Properties

<b>Target Namespace</b>	<a href="http://www.fpml.org/FpML-5/confirmation">http://www.fpml.org/FpML-5/confirmation</a>
<b>Version</b>	\$Revision: 2864 \$
<b>Element and Attribute Namespaces</b>	<ul style="list-style-type: none"> <li>• Global element and attribute declarations belong to this schema's target namespace.</li> <li>• By default, local element declarations belong to this schema's target namespace.</li> <li>• By default, local attribute declarations have no namespace.</li> </ul>
<b>Schema Composition</b>	<ul style="list-style-type: none"> <li>• This schema includes components from the following schema document(s):           <ul style="list-style-type: none"> <li>◦ <a href="#">fpml-option-shared-5-0.xsd</a></li> </ul> </li> </ul>

## Declared Namespaces

Prefix	Namespace
Default namespace	<a href="http://www.fpml.org/FpML-5/confirmation">http://www.fpml.org/FpML-5/confirmation</a>
xml	http://www.w3.org/XML/1998/namespace
fpml-annotation	http://www.fpml.org/annotation
xsd	http://www.w3.org/2001/XMLSchema
ecore	http://www.eclipse.org/emf/2002/Ecore

### Schema Component Representation

```
<xsd:schema targetNamespace="http://www.fpml.org/FpML-5/confirmation"
documentRoot="FpML" nsPrefix="conf" package="org.fpml.confirmation" version="$Revision: 2864 "
$ elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xsd:include schemaLocation="fpml-option-shared-5-0.xsd"/>
  ...
</xsd:schema>
```

## Global Declarations

### Element: additionalCommodityForwardLeg

<b>Name</b>	additionalCommodityForwardLeg
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">CommodityForward</a>
<b>Type</b>	anyType
<b>Nillable</b>	no
<b>Abstract</b>	yes

### Logical Diagram



### XML Instance Representation

```
<additionalCommodityForwardLeg> ... </additionalCommodityForwardLeg>
```

### Diagram



### Schema Component Representation

```
<xsd:element name="additionalCommodityForwardLeg" abstract="true" />
```

**Element: additionalCommoditySwapLeg**

<b>Name</b>	additionalCommoditySwapLeg
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">CommoditySwap</a>
<b>Type</b>	anyType
<b>Nillable</b>	no
<b>Abstract</b>	yes

**Logical Diagram**

```

classDiagram
    class additionalCommoditySwapLeg {
        <<anyType>>
        <<http://www.fpml.org/FpML-5/confirmation>>
        <<@ #any>>
        <<< > #any>>
    }

```

**XML Instance Representation**

```
<additionalCommoditySwapLeg> ... </additionalCommoditySwapLeg>
```

**Diagram**

```

graph LR
    EC[additionalCommoditySwapLeg]

```

**Schema Component Representation**

```
<xsd:element name="additionalCommoditySwapLeg" abstract="true"/>
```

top

**Element: commodityForward**

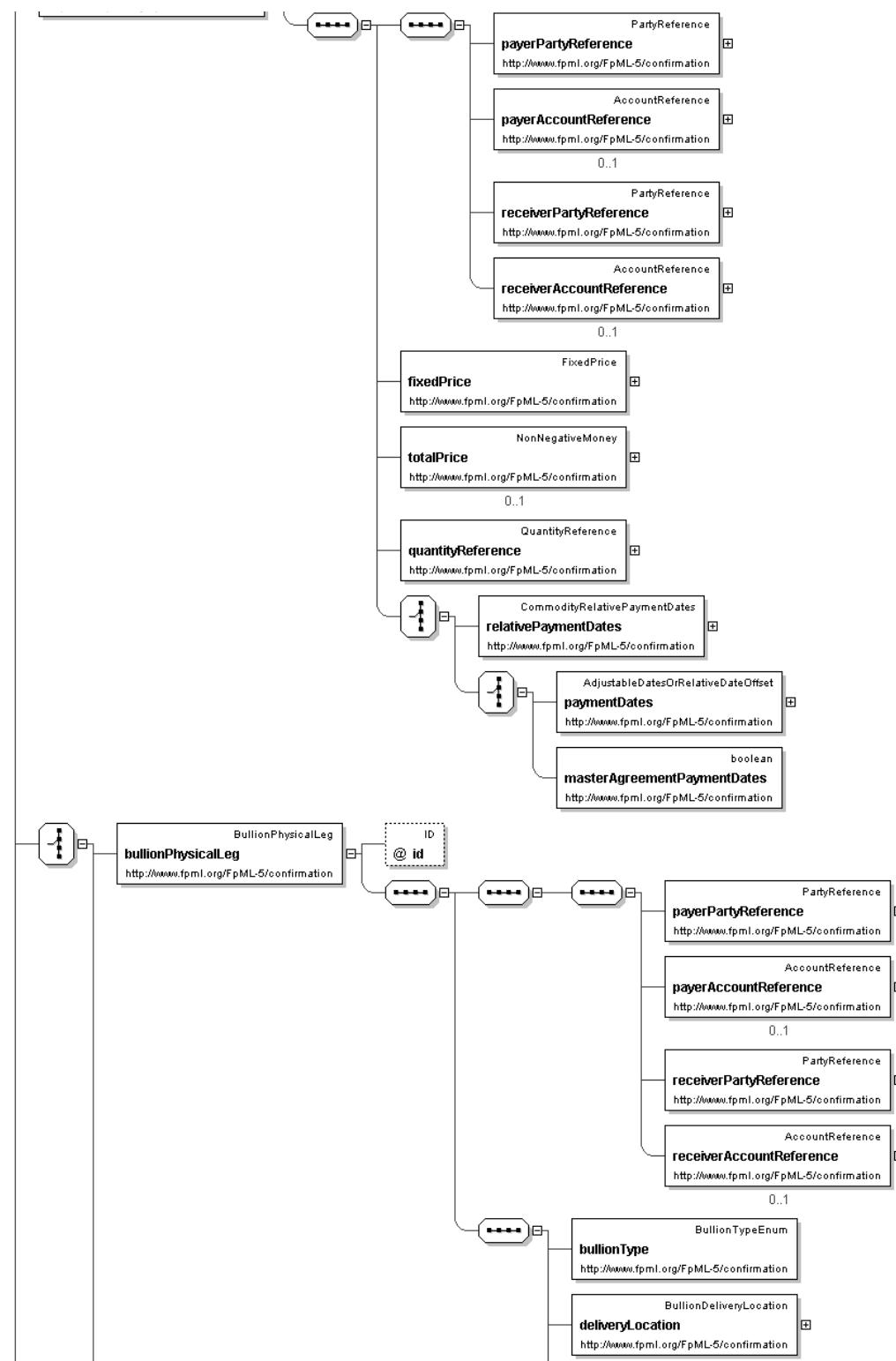
• This element can be used wherever the following element is referenced:	
◦ <a href="#">product</a>	
<b>Name</b>	commodityForward
<b>Used by (from the same schema document)</b>	Model Group <a href="#">CommodityPhysicalOption.model</a>
<b>Type</b>	<a href="#">CommodityForward</a>
<b>Nillable</b>	no
<b>Abstract</b>	no
<b>Documentation</b>	Defines a commodity forward product.

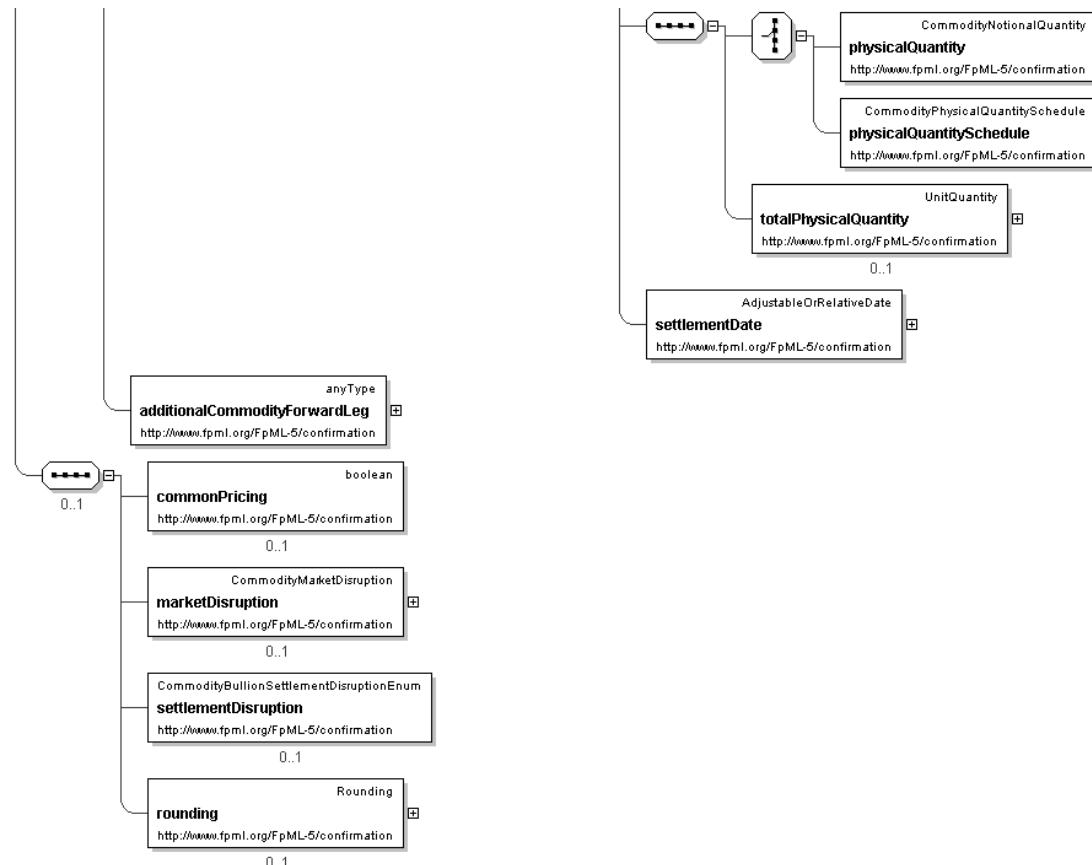
**Logical Diagram**

```

classDiagram
    class commodityForward {
        <<ProductType>>
        <<http://www.fpml.org/FpML-5/confirmation>>
        <<0..∞ >>
        <<0..∞ >>
        <<0..1 >>
        <<0..1 >>
    }
    class productType {
        <<http://www.fpml.org/FpML-5/confirmation>>
        <<0..∞ >>
    }
    class productId {
        <<http://www.fpml.org/FpML-5/confirmation>>
        <<0..∞ >>
    }
    class valueDate {
        <<http://www.fpml.org/FpML-5/confirmation>>
        <<0..1 >>
    }
    class fixedLeg {
        <<http://www.fpml.org/FpML-5/confirmation>>
        <<0..1 >>
    }
    class AdjustableOrRelativeDate {
        <<http://www.fpml.org/FpML-5/confirmation>>
        <<0..1 >>
    }
    class adjustableDate {
        <<http://www.fpml.org/FpML-5/confirmation>>
        <<1..1 >>
    }
    class relativeDateOffset {
        <<http://www.fpml.org/FpML-5/confirmation>>
        <<1..1 >>
    }

```



**XML Instance Representation**

```

<commodityForward
  id=" xsd:ID [0..1]">
  <productType> ProductType </productType> [0..*]
  'A classification of the type of product. FpML defines a simple product categorization using
  a coding scheme.'
  <productId> ProductId </productId> [0..*]
  'A product 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.'
  <valueDate> AdjustableOrRelativeDate </valueDate> [0..1]
  'Specifies the value date of the Commodity Forward Transaction. This is the day on which
  both the cash and the physical commodity settle.'
  <fixedLeg> NonPeriodicFixedPriceLeg </fixedLeg> [1]
  'The fixed leg of a Commodity Forward Transaction'
  Start Choice [1]
  'This choice group is intended to allow legs based on different classes of commodity to
  be added to the schema as this becomes necessary.'
  <bullionPhysicalLeg> BullionPhysicalLeg </bullionPhysicalLeg> [1]
  'The physical leg of a Commodity Forward Transaction for which the underlyer is Bullion.'
  <additionalCommodityForwardLeg> ... </additionalCommodityForwardLeg> [1]
End Choice

```

Start Group: [CommodityContent.model](#) [0..1]  
 <commonPricing> xsd:boolean </commonPricing> [0..1]

'Common pricing may be relevant for a Transaction that references more than one Commodity Reference Price. If Common Pricing is not specified as applicable, it will be deemed not to apply.'

<marketDisruption> [CommodityMarketDisruption](#) </marketDisruption> [0..1]

'Market disruption events as defined in the ISDA 1993 Commodity Definitions or in ISDA 2005 Commodity Definitions, as applicable.'

<settlementDisruption> [CommodityBullionSettlementDisruptionEnum](#) </settlementDisruption> [0..1]

'The consequences of Bullion Settlement Disruption Events.'

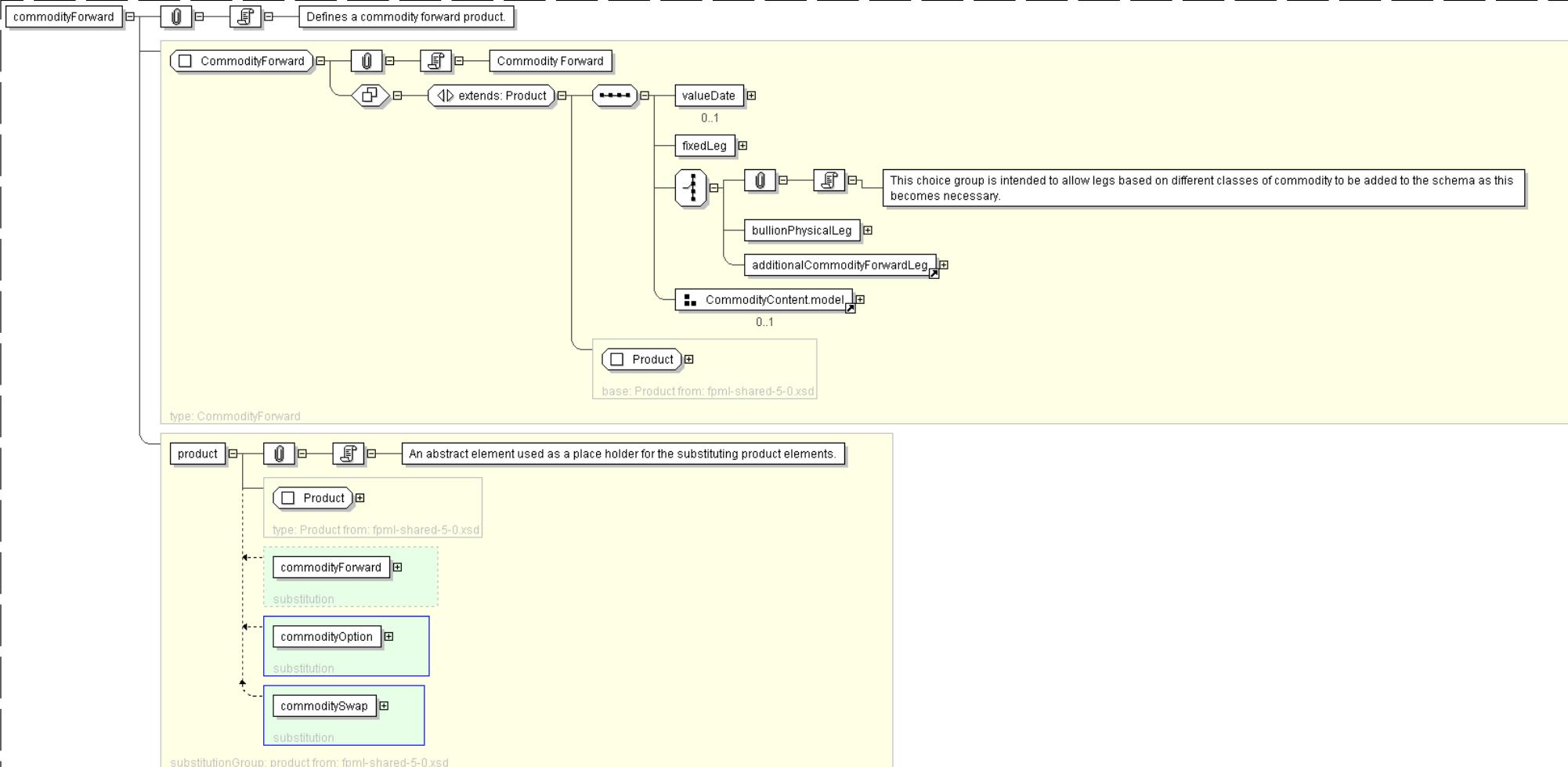
<rounding> [Rounding](#) </rounding> [0..1]

'Rounding direction and precision for amounts.'

End Group: [CommodityContent.model](#)

</commodityForward>

#### Diagram



#### Schema Component Representation

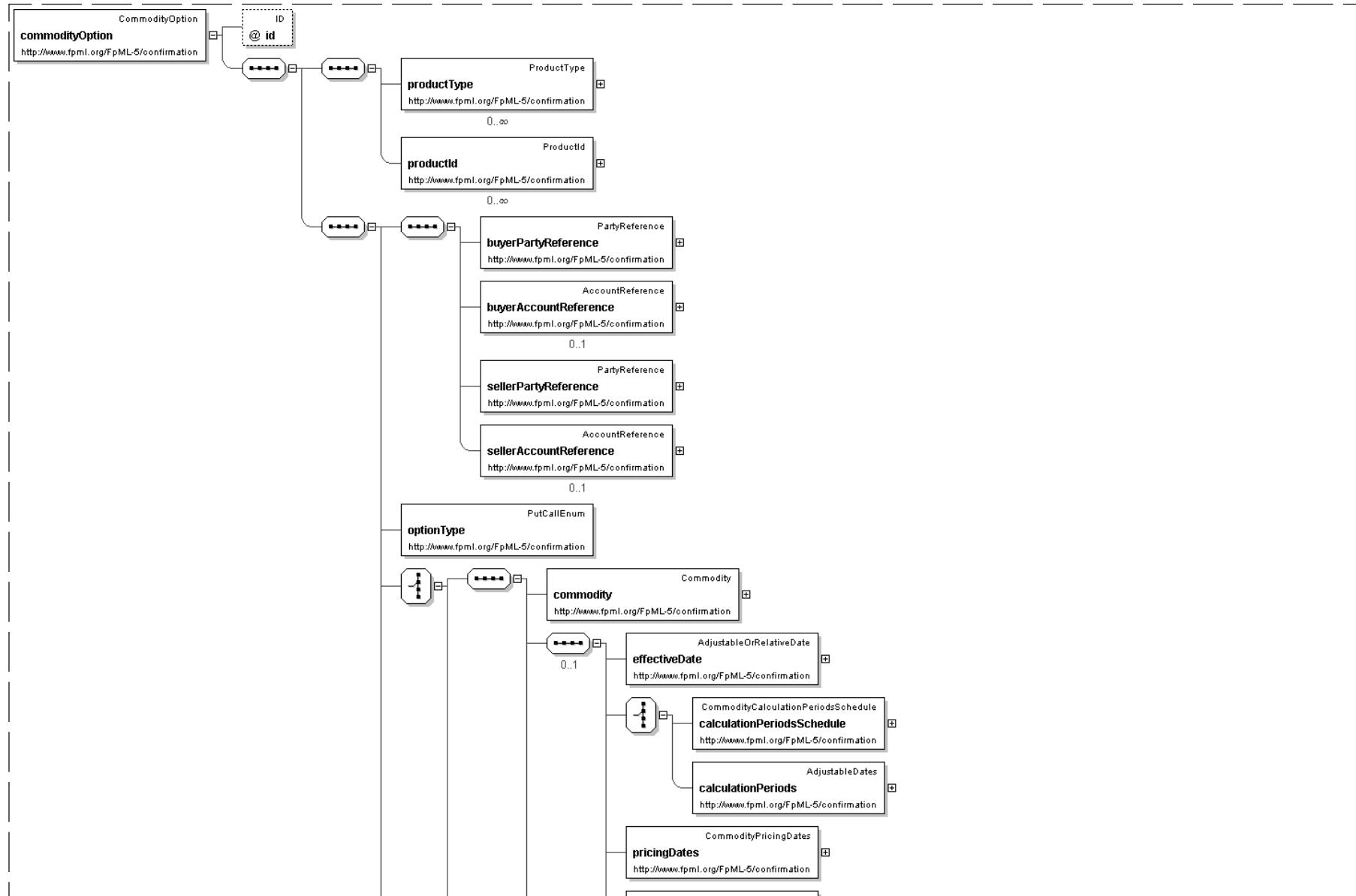
```
<xsd:element name="commodityForward" type=" CommodityForward " substitutionGroup="product"/>
```

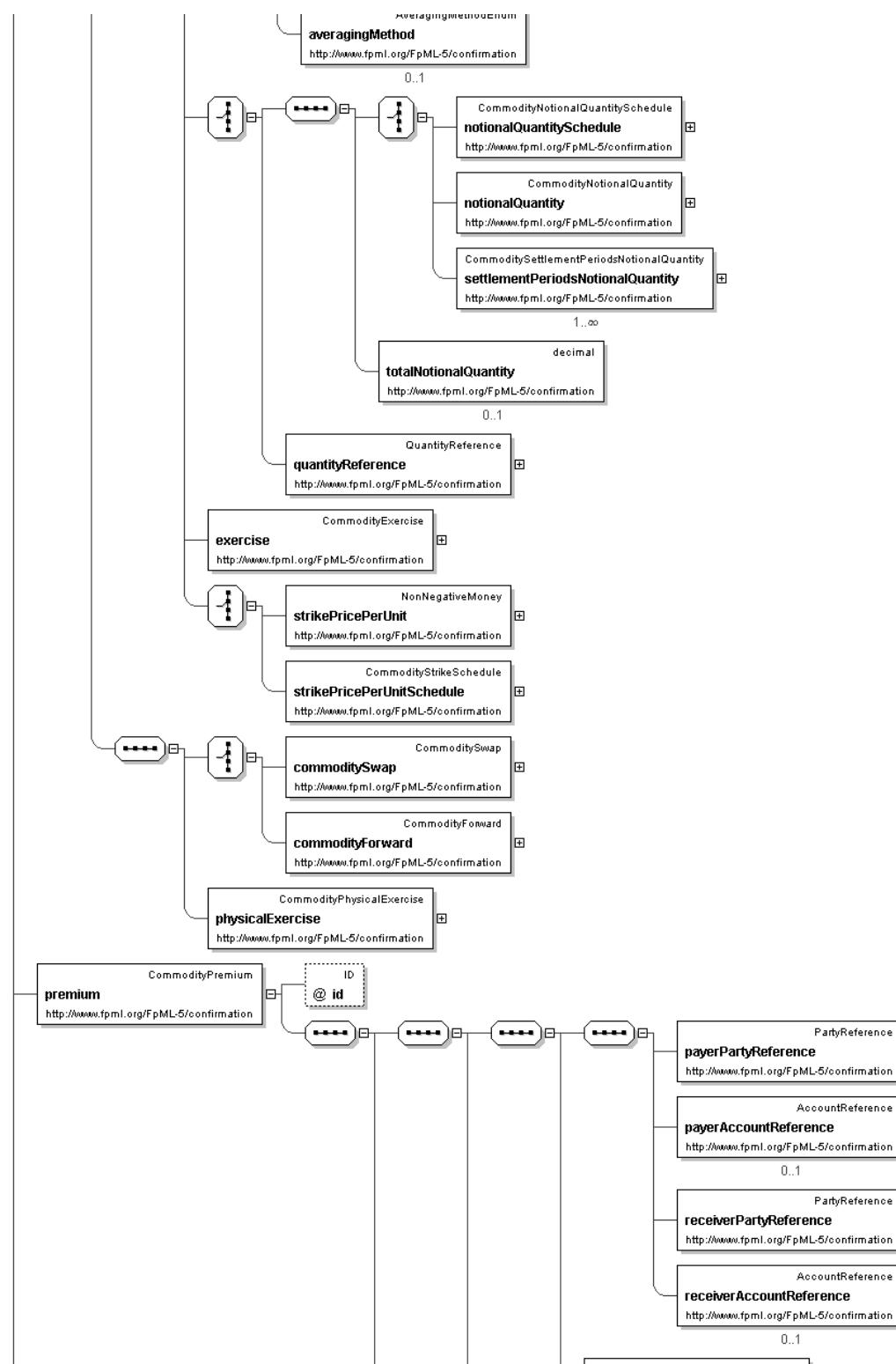
## Element: commodityOption

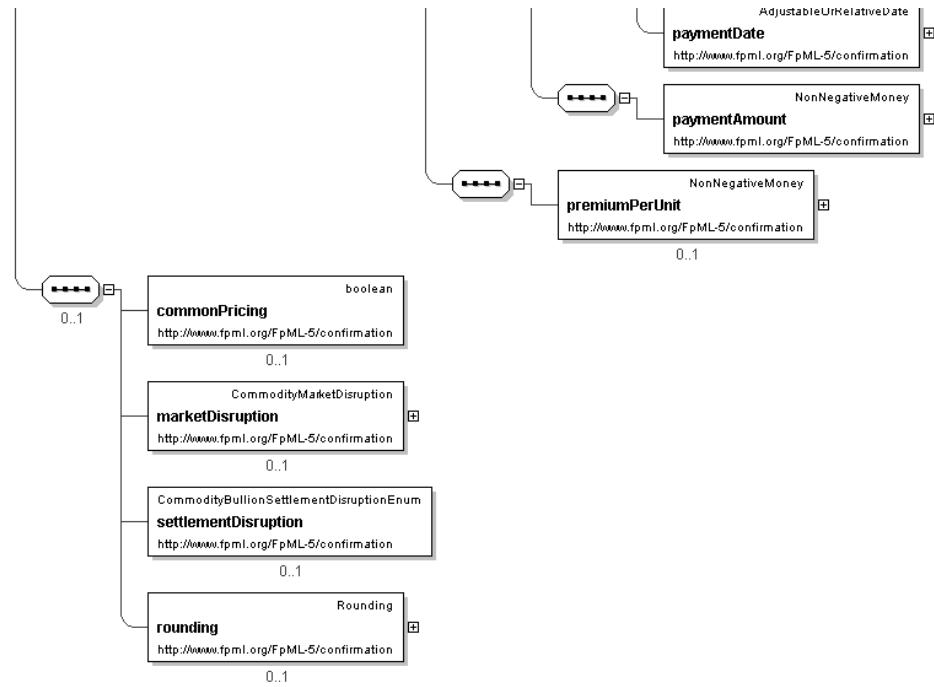
- This element can be used wherever the following element is referenced:
  - [product](#)

Name	commodityOption
Type	<a href="#">CommodityOption</a>
Nillable	no
Abstract	no
Documentation	Defines a commodity option product.

### Logical Diagram





**XML Instance Representation**

```

<commodityOption
id=" xsd:ID [0..1]">
  <productType> ProductType </productType> [0..*]
  'A classification of the type of product. FpML defines a simple product categorization using
  a coding scheme.'

  <productId> ProductId </productId> [0..*]
  'A product 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.'

  <buyerPartyReference> PartyReference </buyerPartyReference> [1]
  'A reference to the party that buys this instrument, i.e. pays for this instrument and
  receives the rights defined by it. See 2000 ISDA definitions Article 11.1 (b). In the case
  of FRAs this is the fixed rate payer.'

  <buyerAccountReference> AccountReference </buyerAccountReference> [0..1]
  'A reference to the account that buys this instrument.'

  <sellerPartyReference> PartyReference </sellerPartyReference> [1]
  'A reference to the party that sells ("writes") this instrument, i.e. that grants the
  rights defined by this instrument and in return receives a payment for it. See 2000
  ISDA definitions Article 11.1 (a). In the case of FRAs this is the floating rate payer.'

  <sellerAccountReference> AccountReference </sellerAccountReference> [0..1]
  'A reference to the account that sells this instrument.'

  <optionType> PutCallEnum </optionType> [1]
  'The type of option transaction.'

Start Choice [1]
  <commodity> Commodity </commodity> [1]
  'Specifies the underlying component. At the time of the initial schema design, only
  underlyers of type Commodity are supported; the choice group in the future could offer
  the possibility of adding other types later.'

```

Start Group: CommodityAsian.model [0..1]

'A group containing properties specific to Asian options.'

<effectiveDate> AdjustableOrRelativeDate </effectiveDate> [1]  
 'The effective date of the Commodity Option Transaction. Note that the Termination/Expiration Date should be specified in expirationDate within the CommodityAmericanExercise type or the CommodityEuropeanExercise type, as applicable.'

Start Choice [1]

- <calculationPeriodsSchedule> CommodityCalculationPeriodsSchedule </calculationPeriodsSchedule> [1]  
 'A parametric representation of the Calculation Periods of the Commodity Option Transaction.'
- <calculationPeriods> AdjustableDates </calculationPeriods> [1]  
 'An absolute representation of the Calculation Period start dates of the Commodity Option Transaction.'

End Choice

<pricingDates> CommodityPricingDates </pricingDates> [1]  
 'The dates on which the option will price.'

<averagingMethod> AveragingMethodEnum </averagingMethod> [0..1]  
 'The Method of Averaging if there is more than one Pricing Date.'

End Group: CommodityAsian.model

Start Choice [1]

Start Choice [1]

- <notionalQuantitySchedule> CommodityNotionalQuantitySchedule </notionalQuantitySchedule> [1]  
 'Allows the documentation of a shaped notional trade where the notional changes over the life of the transaction.'
- <notionalQuantity> CommodityNotionalQuantity </notionalQuantity> [1]  
 'The Notional Quantity.'
- <settlementPeriodsNotionalQuantity> CommoditySettlementPeriodsNotionalQuantity  
</settlementPeriodsNotionalQuantity> [1..\*]  
 'For an electricity transaction, the Notional Quantity for a one or more groups of Settlement Periods to which the Notional Quantity is based. If the schedule differs for different groups of Settlement Periods, this element should be repeated.'

End Choice

<totalNotionalQuantity> xsd:decimal </totalNotionalQuantity> [0..1]  
 'The Total Notional Quantity.'

<quantityReference> QuantityReference </quantityReference> [1]  
 'A pointer style reference to a quantity defined on another leg.'

End Choice

<exercise> CommodityExercise </exercise> [1]  
 'The parameters for defining how the commodity option can be exercised and how it is settled.'

Start Choice [1]

- <strikePricePerUnit> NonNegativeMoney </strikePricePerUnit> [1]  
 'The currency amount of the strike price per unit.'
- <strikePricePerUnitSchedule> CommodityStrikeSchedule </strikePricePerUnitSchedule> [1]

End Choice

Start Choice [1]

- <commoditySwap> ... </commoditySwap> [1]
- <commodityForward> ... </commodityForward> [1]

End Choice

<physicalExercise> CommodityPhysicalExercise </physicalExercise> [1]  
 'The parameters for defining how the commodity option can be exercised into a physical transaction.'

End Choice

<premium> CommodityPremium </premium> [1]  
 'The option premium payable by the buyer to the seller.'

Start Group: CommodityContent.model [0..1]

<commonPricing> xsd:boolean </commonPricing> [0..1]  
 'Common pricing may be relevant for a Transaction that references more than one

*Commodity Reference Price. If Common Pricing is not specified as applicable, it will be deemed not to apply.'*

<marketDisruption> [CommodityMarketDisruption](#) </marketDisruption> [0..1]

'Market disruption events as defined in the ISDA 1993 Commodity Definitions or in ISDA 2005 Commodity Definitions, as applicable.'

<settlementDisruption> [CommodityBullionSettlementDisruptionEnum](#) </settlementDisruption> [0..1]

'The consequences of Bullion Settlement Disruption Events.'

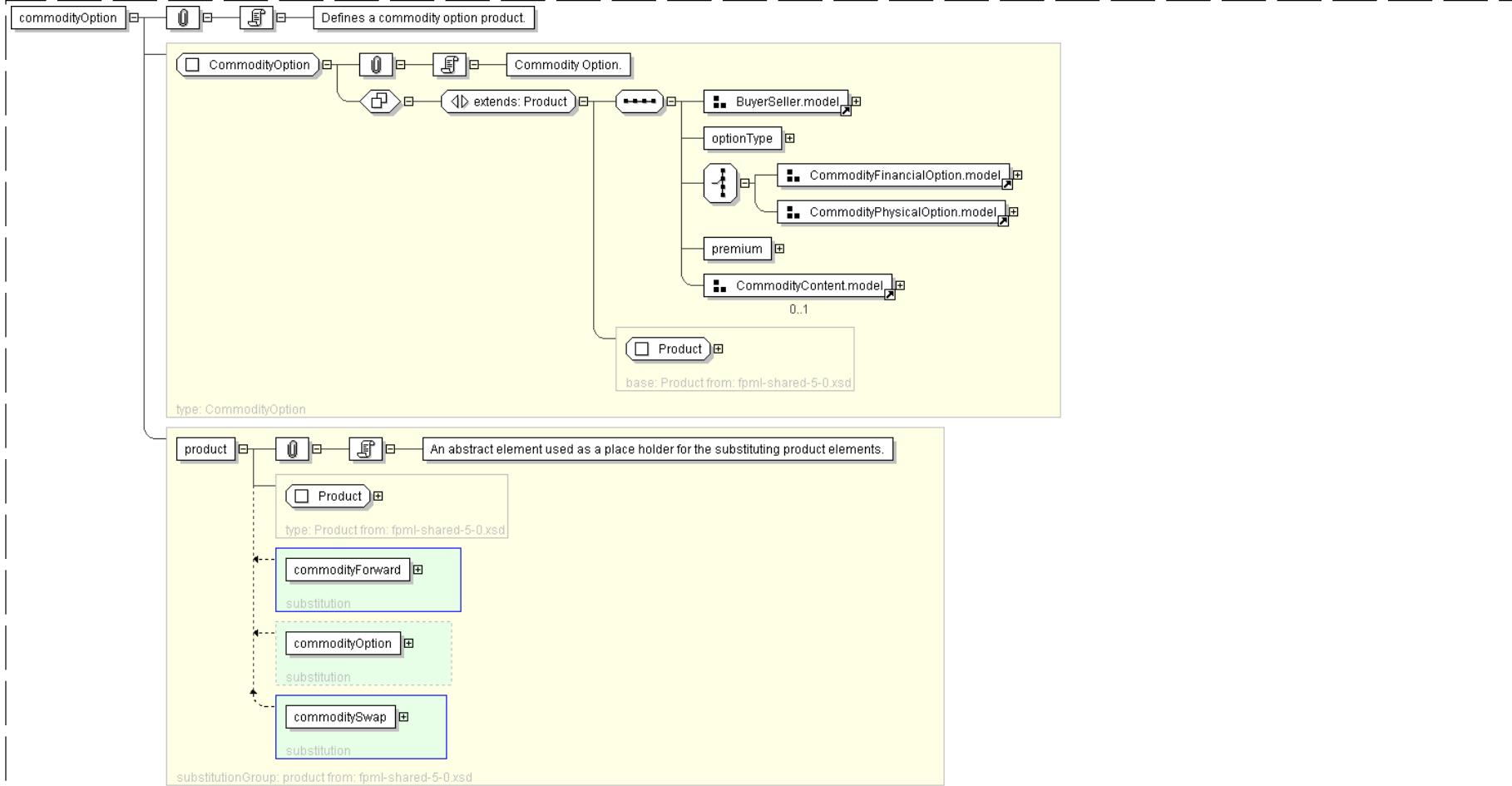
<rounding> [Rounding](#) </rounding> [0..1]

'Rounding direction and precision for amounts.'

End Group: [CommodityContent.model](#)

</commodityOption>

#### Diagram



#### Schema Component Representation

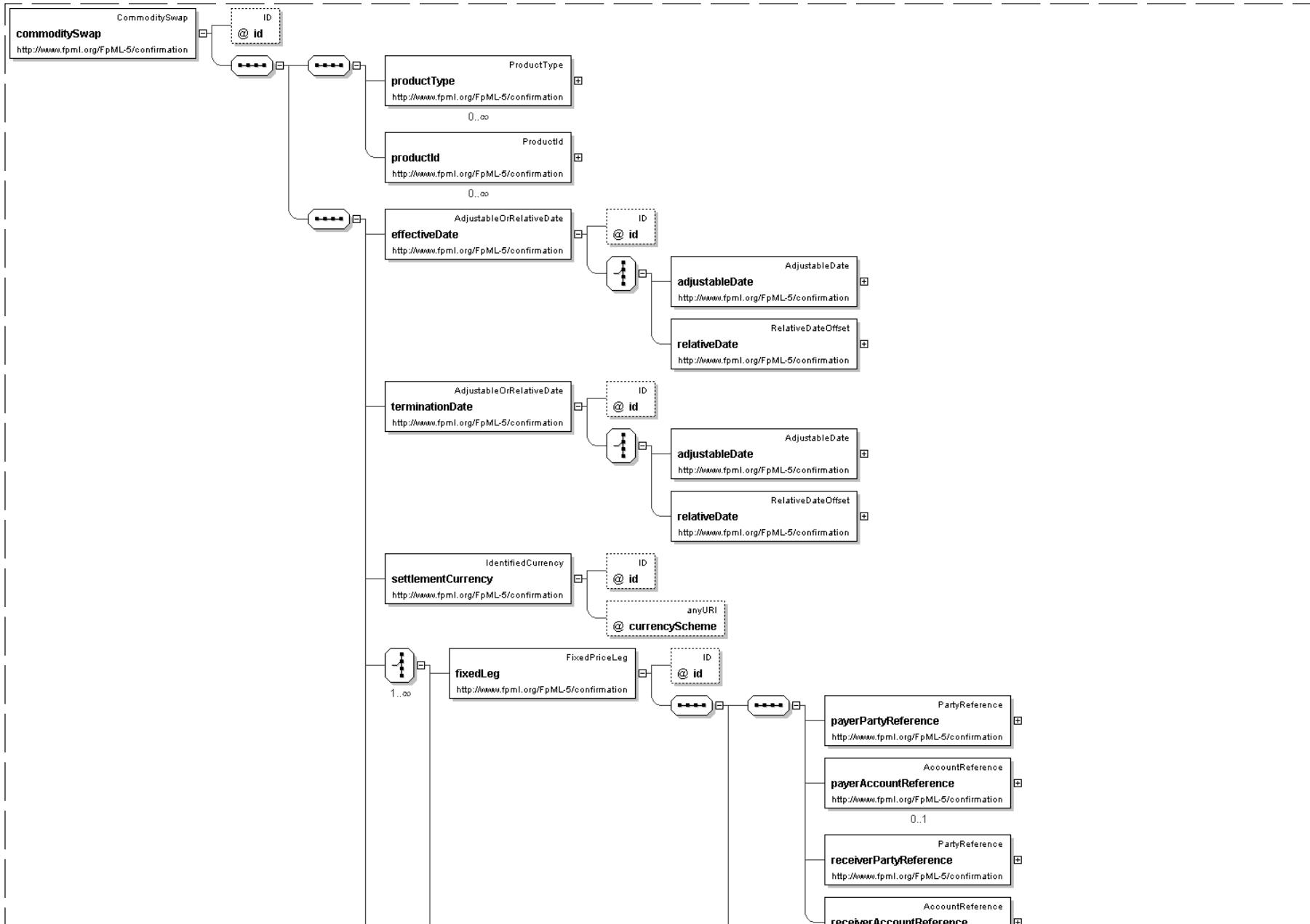
```
<xsd:element name="commodityOption" type="#CommodityOption" substitutionGroup="product"/>
```

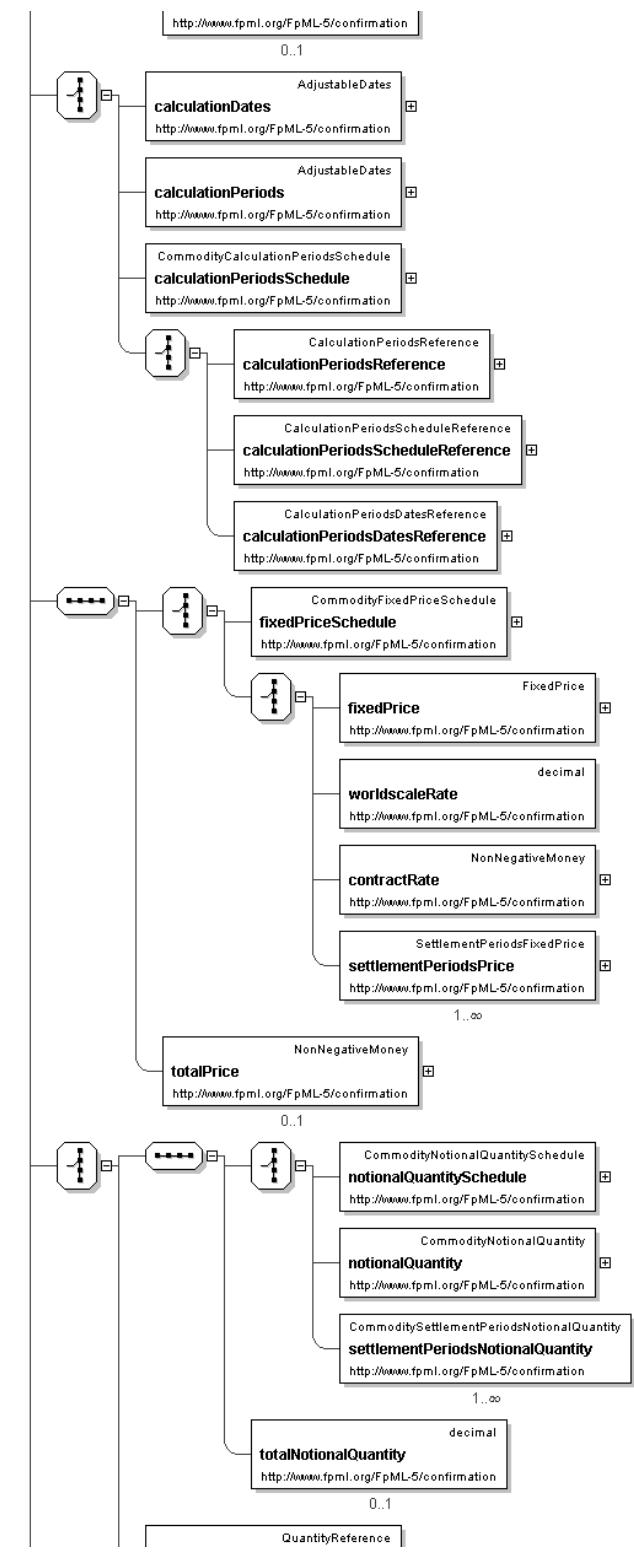
#### Element: `commoditySwap`

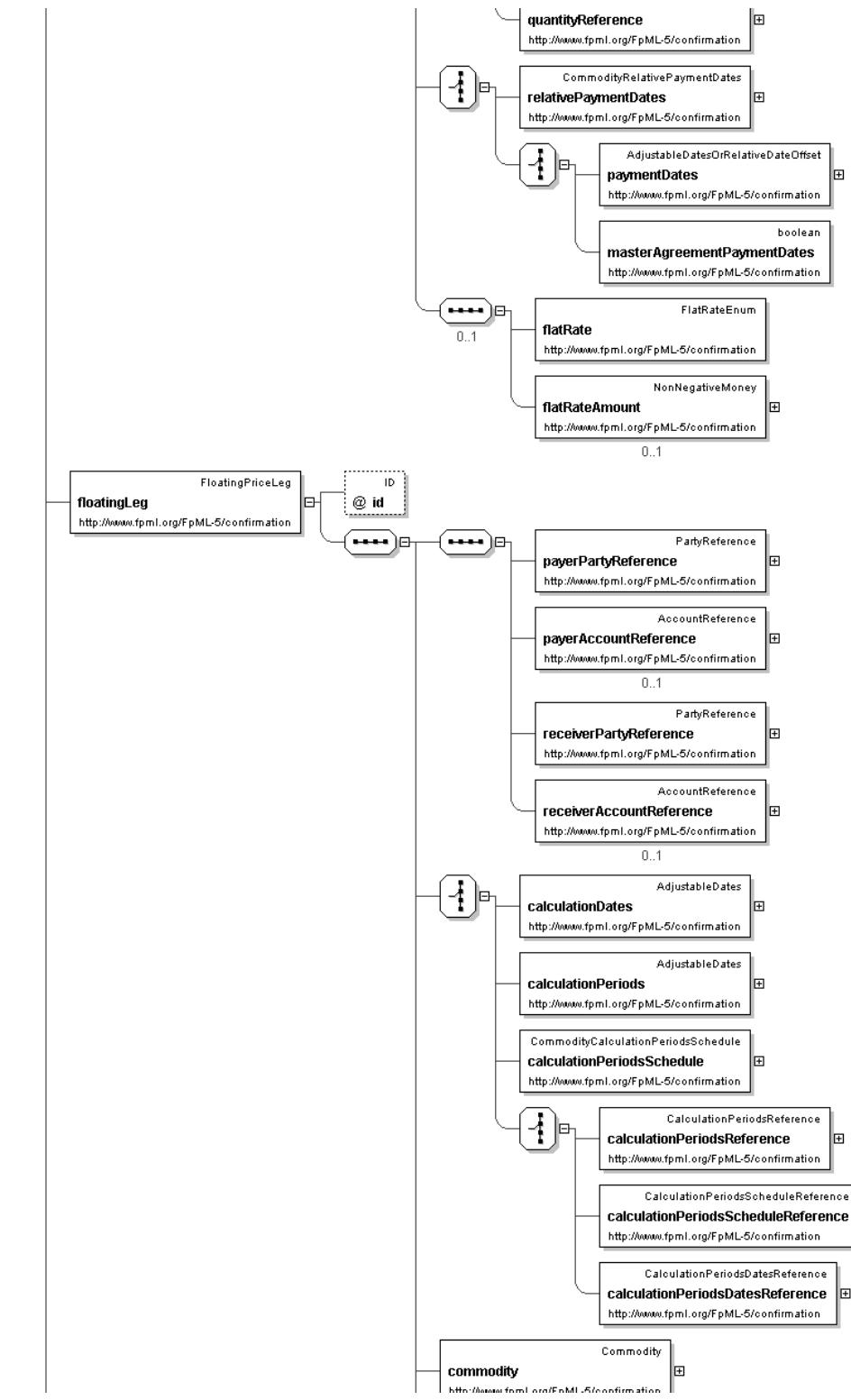
- This element can be used wherever the following element is referenced:
  - [product](#)

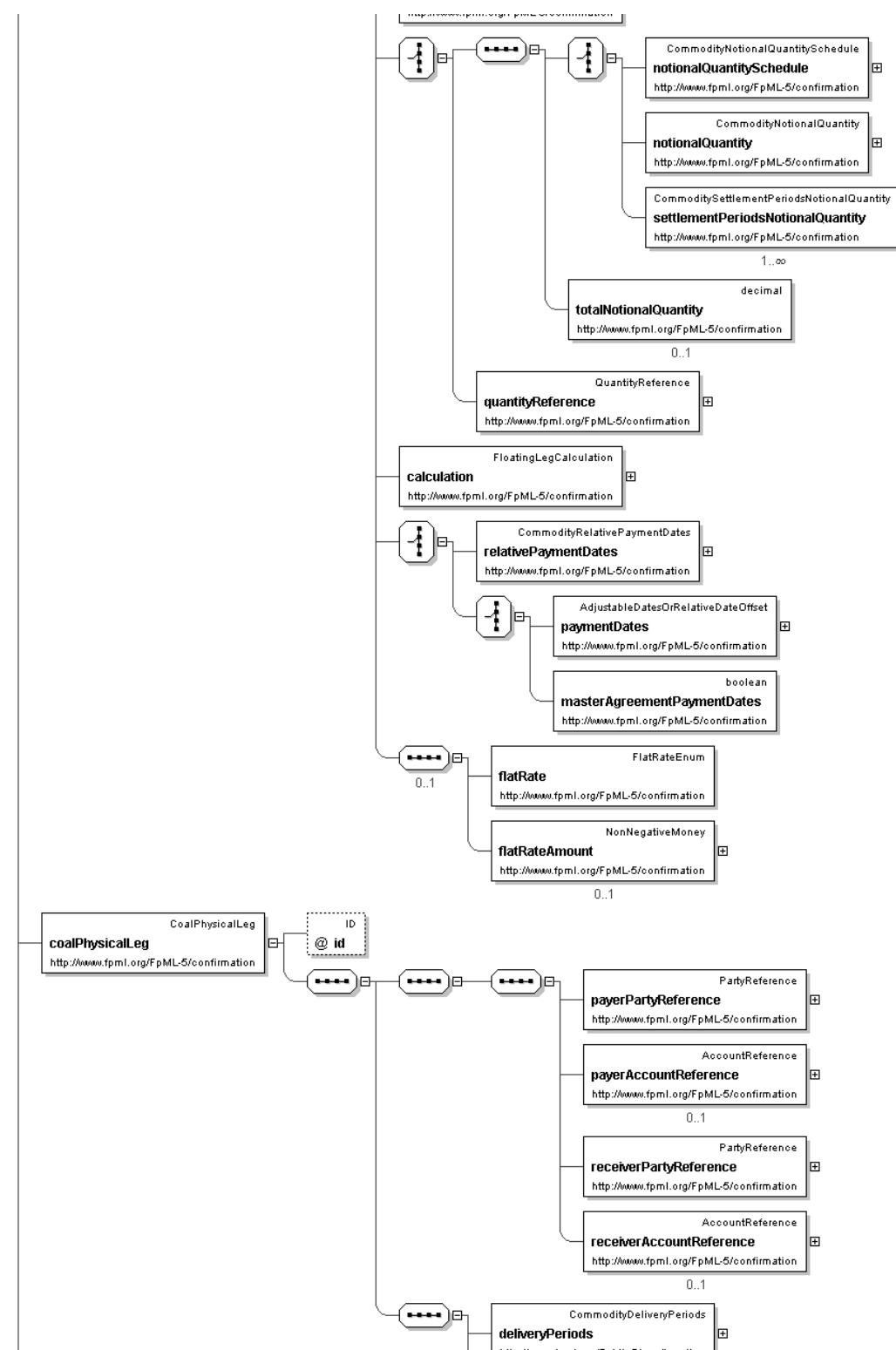
Name	commoditySwap
Used by (from the same schema document)	Model Group <a href="#">CommodityPhysicalOption.model</a>
Type	<a href="#">CommoditySwap</a>
Nillable	no
Abstract	no
Documentation	Defines a commodity swap product.

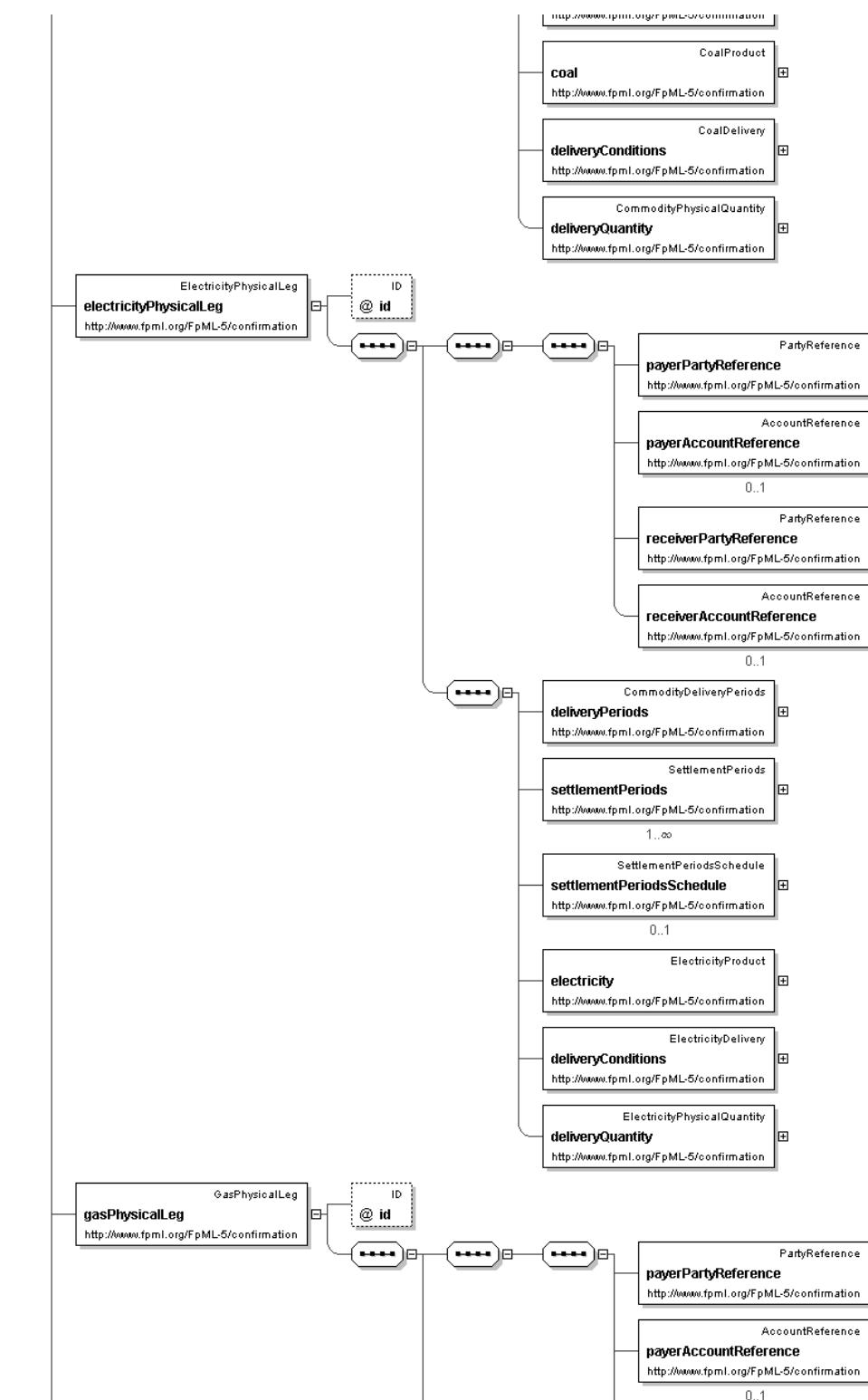
## Logical Diagram

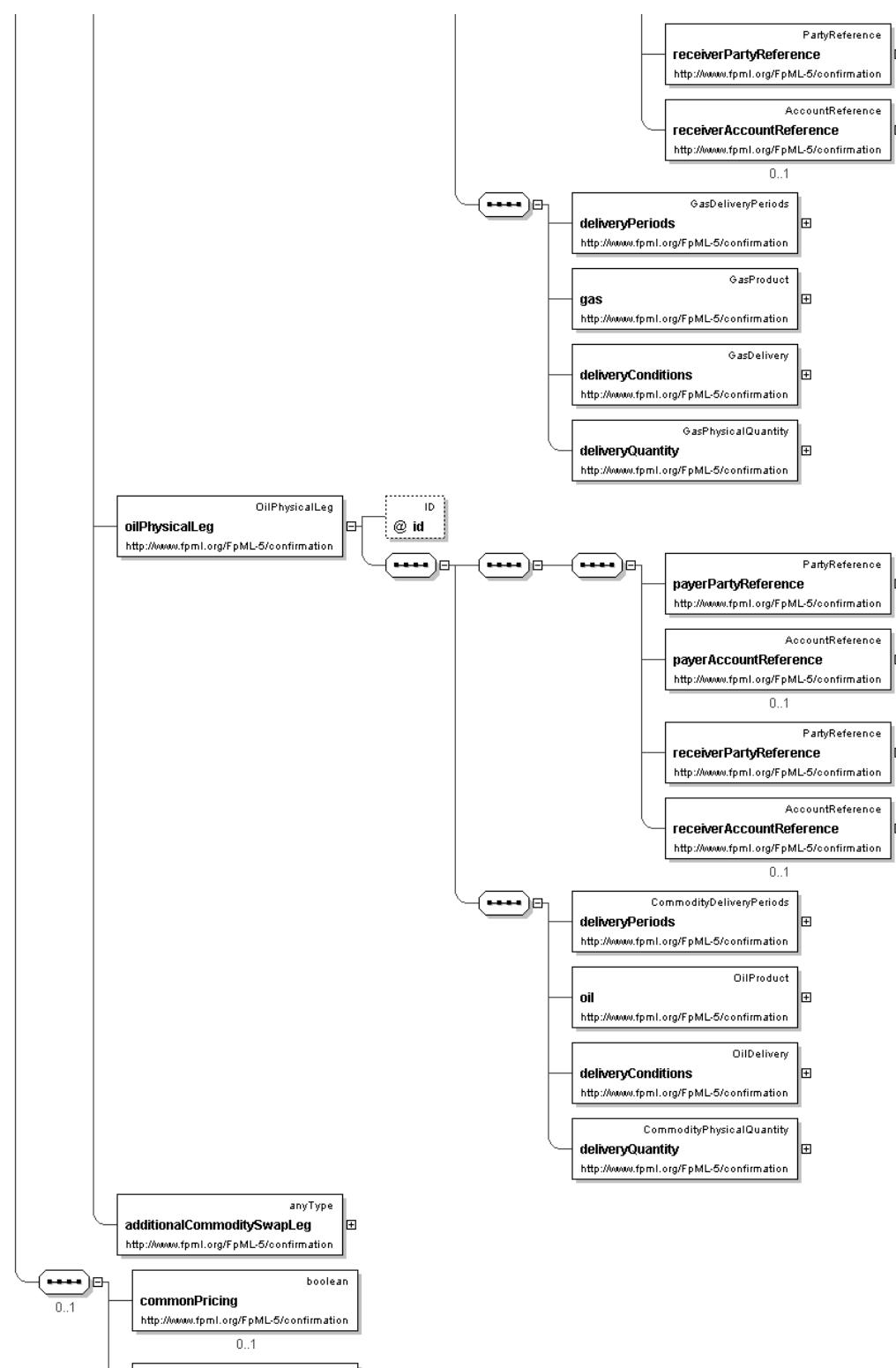


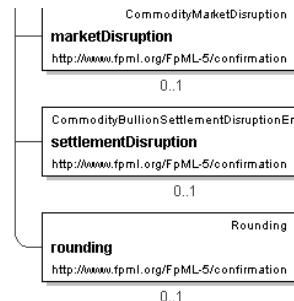










**XML Instance Representation**

```

<commoditySwap
id=" xsd:ID [0..1]">
  <productType> ProductType </productType> [0..*]
  'A classification of the type of product. FpML defines a simple product categorization using
  a coding scheme.'

  <productId> ProductId </productId> [0..*]
  'A product 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.'

  <effectiveDate> AdjustableOrRelativeDate </effectiveDate> [1]
  'Specifies the effective date of this leg of the swap. When defined in relation to a
  date specified somewhere else in the document (through the relativeDate component),
  this element will typically point to the effective date of the other leg of the swap.'

  <terminationDate> AdjustableOrRelativeDate </terminationDate> [1]
  'Specifies the termination date of this leg of the swap. When defined in relation to a
  date specified somewhere else in the document (through the relativeDate component),
  this element will typically point to the termination date of the other leg of the swap.'

  <settlementCurrency> IdentifiedCurrency </settlementCurrency> [1]
  'The currency into which the Commodity Swap Transaction will settle. If this is not the same
  as the currency in which the Commodity Reference Price is quoted on a given floating leg of
  the Commodity Swap Transaction, then an FX rate should also be specified for that leg.'

Start Choice [1..*]
  <fixedLeg> FixedPriceLeg </fixedLeg> [1]
  'Fixed Price Leg.'

  <floatingLeg> FloatingPriceLeg </floatingLeg> [1]
  'Floating Price leg.'

  <coalPhysicalLeg> CoalPhysicalLeg </coalPhysicalLeg> [1]
  'Physically settled coal leg.'

  <electricityPhysicalLeg> ElectricityPhysicalLeg </electricityPhysicalLeg> [1]
  'Physically settled electricity leg.'

  <gasPhysicalLeg> GasPhysicalLeg </gasPhysicalLeg> [1]
  'Physically settled natural gas leg.'

  <oilPhysicalLeg> OilPhysicalLeg </oilPhysicalLeg> [1]
  'Physically settled oil or refined products leg.'

  <additionalCommoditySwapLeg> ... </additionalCommoditySwapLeg> [1]
End Choice
Start Group: CommodityContent.model [0..1]
  <commonPricing> xsd:boolean </commonPricing> [0..1]
  'Common pricing may be relevant for a Transaction that references more than one
  Commodity Reference Price. If Common Pricing is not specified as applicable, it will be
  deemed not to apply.'

  <marketDisruption> CommodityMarketDisruption </marketDisruption> [0..1]
  'Market disruption events as defined in the ISDA 1993 Commodity Definitions or in ISDA
  
```

*2005 Commodity Definitions, as applicable.'*

```
<settlementDisruption> CommodityBullionSettlementDisruptionEnum </settlementDisruption> [0...]
```

'The consequences of Bullion Settlement Disruption Events.

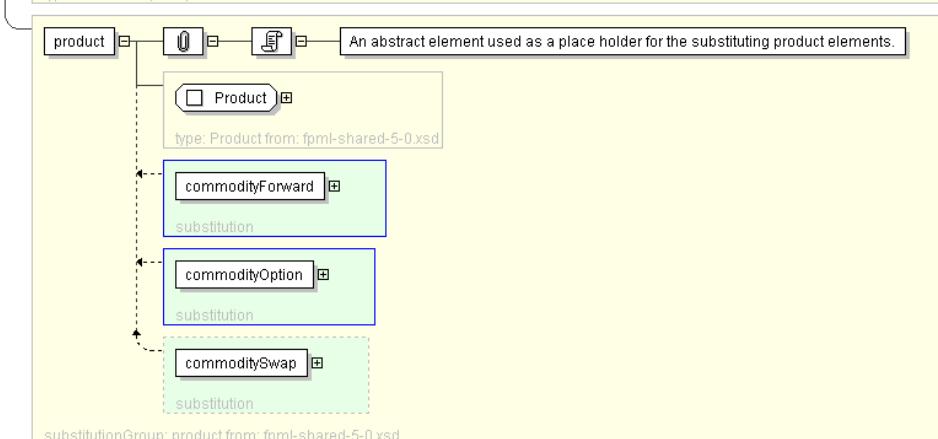
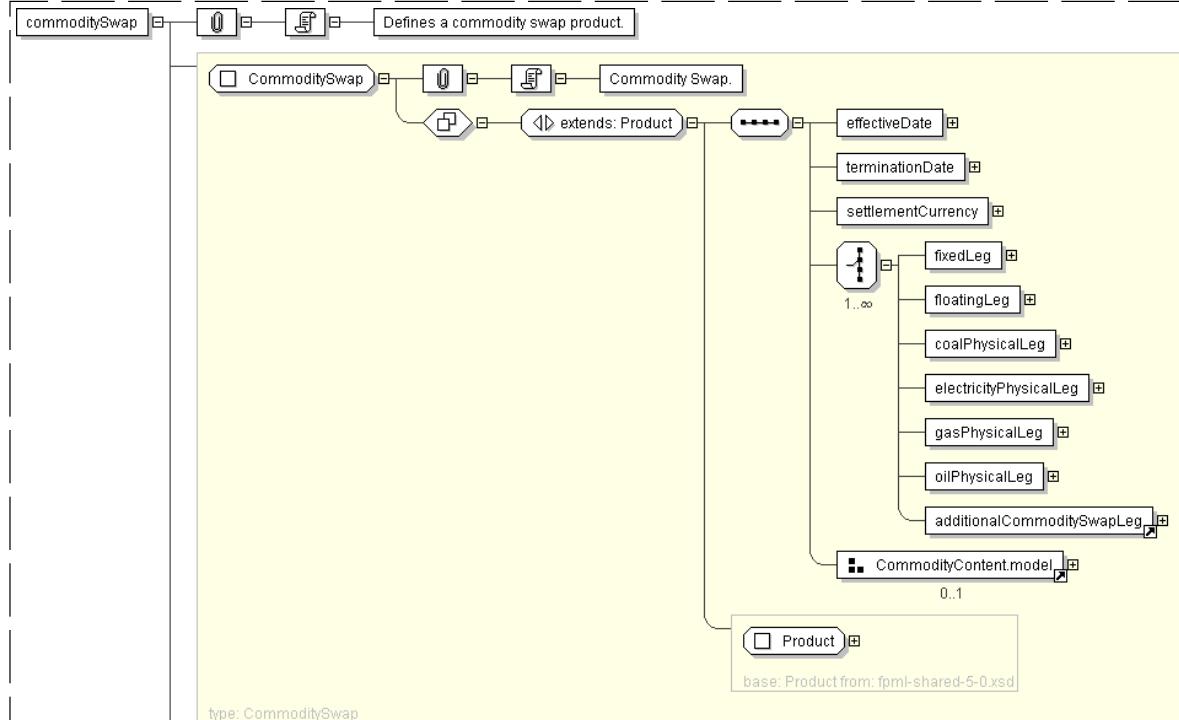
<rounding> Rounding </rounding> [0..1]

### *'Rounding direction and precision for amounts.'*

End Group: CommodityContent.model

</commoditySwap>

## Diagram



## Schema Component Representation

```
<xsd:element name="commoditySwap" type=" CommoditySwap " substitutionGroup="product"
```

## Global Definitions

### Complex Type: **AbsoluteTolerance**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None

<b>Name</b>	AbsoluteTolerance
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">OilDelivery</a>
<b>Abstract</b>	no
<b>Documentation</b>	The acceptable tolerance in the delivered quantity of a physical commodity product in terms of a number of units of that product.

#### XML Instance Representation

```

<...>
<positive> xsd:decimal </positive> [1]
'The maximum amount by which the quantity delivered can exceed the agreed quantity.'

<negative> xsd:decimal </negative> [1]
'The maximum amount by which the quantity delivered can be less than the agreed quantity.'

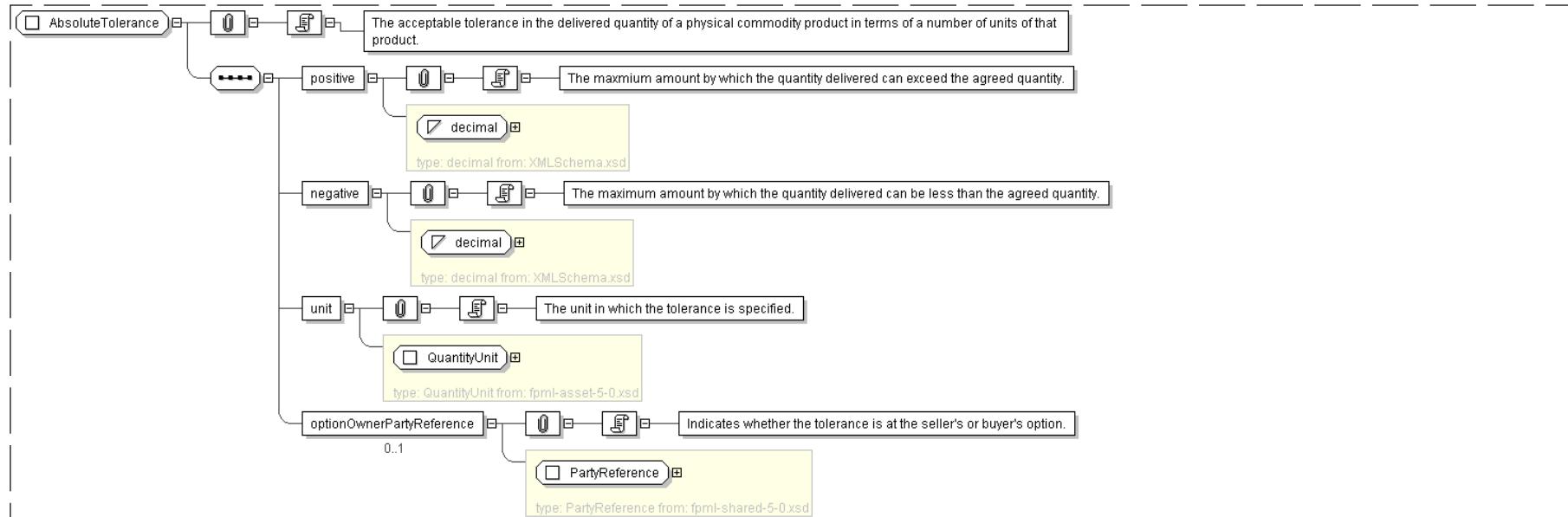
<unit> QuantityUnit </unit> [1]
'The unit in which the tolerance is specified.'

<optionOwnerPartyReference> PartyReference </optionOwnerPartyReference> [0..1]
'Indicates whether the tolerance is at the seller's or buyer's option.'

</...>

```

#### Diagram



#### Schema Component Representation

```

<xsd:complexType name="AbsoluteTolerance">
  <xsd:sequence>
    <xsd:element name="positive" type="xsd:decimal" />
    <xsd:element name="negative" type="xsd:decimal" />
    <xsd:element name="unit" type="QuantityUnit" />
    <xsd:element name="optionOwnerPartyReference" type="PartyReference" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>

```

&lt;/xsd:complexType&gt;

top

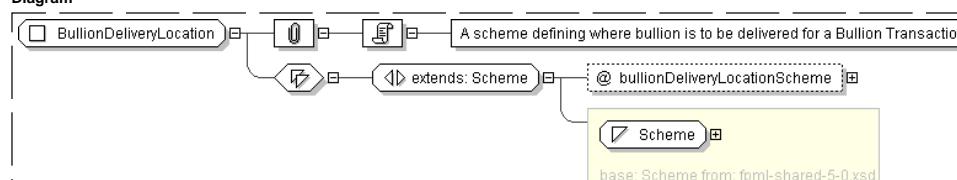
**Complex Type: BullionDeliveryLocation**

**Super-types:** [Scheme](#) < **BullionDeliveryLocation** (by extension)  
**Sub-types:** None

<b>Name</b>	BullionDeliveryLocation
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">BullionPhysicalLeg</a>
<b>Abstract</b>	no
<b>Documentation</b>	A scheme defining where bullion is to be delivered for a Bullion Transaction.

**XML Instance Representation**

```
<...>
  <bullionDeliveryLocationScheme="" type="xsd:anyURI" [0..1]>
    <Scheme>
  </...>
```

**Diagram****Schema Component Representation**

```
<xsd:complexType name="BullionDeliveryLocation">
  <xsd:simpleContent>
    <xsd:extension base="">
      <xsd:attribute name="bullionDeliveryLocationScheme" type="xsd:anyURI" default="http://www.fpml.org/coding-scheme/bullion-delivery-location"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
```

top

**Complex Type: BullionPhysicalLeg**

**Super-types:** [Leg](#) < [PhysicalLeg](#) (by extension) < **BullionPhysicalLeg** (by extension)  
**Sub-types:** None

<b>Name</b>	BullionPhysicalLeg
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">CommodityForward</a>
<b>Abstract</b>	no
<b>Documentation</b>	Physically settled leg of a physically settled Bullion Transaction.

**XML Instance Representation**

```
<...>
  <id="" type="ID" [0..1]>
    <payerPartyReference> <PartyReference> </PartyReference> [1]
    'A reference to the party responsible for making the payments defined by this structure.'
  <payerAccountReference> <AccountReference> </AccountReference> [0..1]
    'A reference to the account responsible for making the payments defined by this structure.'
  <receiverPartyReference> <PartyReference> </PartyReference> [1]
    'A reference to the party that receives the payments corresponding to this structure.'
  <receiverAccountReference> <AccountReference> </AccountReference> [0..1]
    'A reference to the account that receives the payments corresponding to this structure.'
```

```

<bullionType> BullionTypeEnum </bullionType> [1]
'The type of Bullion underlying a Bullion Transaction.'

<deliveryLocation> BullionDeliveryLocation </deliveryLocation> [1]
'The physical delivery location for the transaction.'

Start Choice [1]
<physicalQuantity> CommodityNotionalQuantity </physicalQuantity> [1]
'The Quantity per Delivery Period.'

<physicalQuantitySchedule> CommodityPhysicalQuantitySchedule </physicalQuantitySchedule> [1]
'Allows the documentation of a shaped quantity trade where the quantity changes over the life of the transaction.'

```

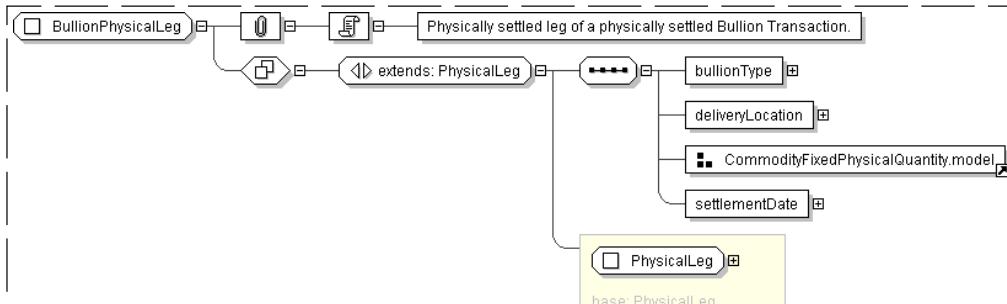
```

End Choice
<totalPhysicalQuantity> UnitQuantity </totalPhysicalQuantity> [0..1]
'The Total Quantity of the commodity to be delivered.'

<settlementDate> AdjustableOrRelativeDate </settlementDate> [1]
'Date on which the bullion will settle.'

</...>

```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="BullionPhysicalLeg">
  <xsd:complexContent>
    <xsd:extension base=" PhysicalLeg ">
      <xsd:sequence>
        <xsd:element name="bullionType" type=" BullionTypeEnum " />
        <xsd:element name="deliveryLocation" type=" BullionDeliveryLocation " />
        <xsd:group ref=" CommodityFixedPhysicalQuantity.model "/>
        <xsd:element name="settlementDate" type=" AdjustableOrRelativeDate " />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

top

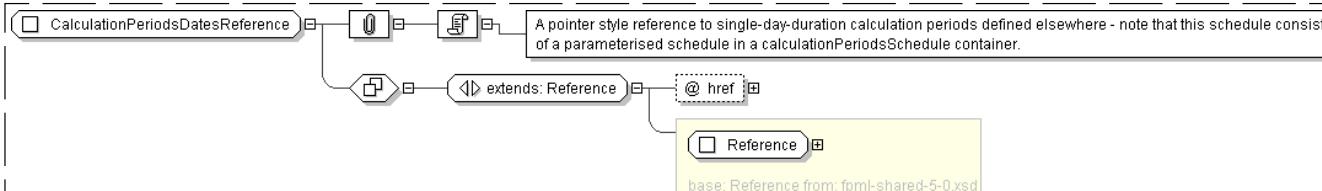
**Complex Type: CalculationPeriodsDatesReference**

Super-types:	<a href="#">Reference</a> < CalculationPeriodsDatesReference (by extension)
Sub-types:	None

Name	CalculationPeriodsDatesReference
Used by (from the same schema document)	Model Group <a href="#">CommodityCalculationPeriodsPointer.model</a>
Abstract	no
Documentation	A pointer style reference to single-day-duration calculation periods defined elsewhere - note that this schedule consists of a parameterised schedule in a calculationPeriodsSchedule container.

**XML Instance Representation**

```
<...>
  href=" xsd:IDREF [1]" />
```

**Diagram****Schema Component Representation**

```
<xsd:complexType name="CalculationPeriodsDatesReference">
  <xsd:complexContent>
    <xsd:extension base=" Reference ">
      <xsd:attribute name="href" type=" xsd:IDREF "
        use="required" reference="CommodityCalculationPeriodsSchedule" />
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

top

**Complex Type: CalculationPeriodsReference**

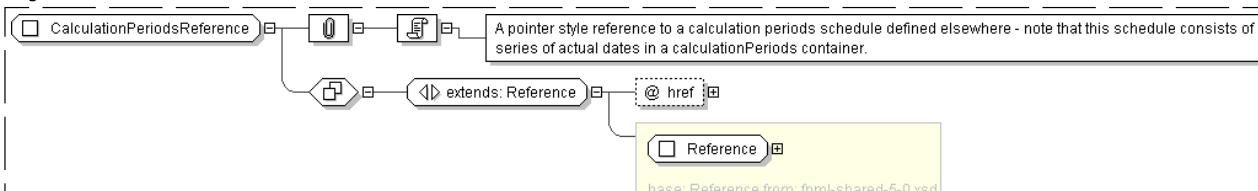
**Super-types:** [Reference](#) < CalculationPeriodsReference (by extension)

**Sub-types:** None

<b>Name</b>	CalculationPeriodsReference
<b>Used by (from the same schema document)</b>	Model Group <a href="#">CommodityDeliveryPeriodsPointer.model</a> , Model Group <a href="#">CommodityCalculationPeriodsPointer.model</a>
<b>Abstract</b>	no
<b>Documentation</b>	A pointer style reference to a calculation periods schedule defined elsewhere - note that this schedule consists of a series of actual dates in a calculationPeriods container.

**XML Instance Representation**

```
<...>
  href=" xsd:IDREF [1]" />
```

**Diagram****Schema Component Representation**

```
<xsd:complexType name="CalculationPeriodsReference">
  <xsd:complexContent>
    <xsd:extension base=" Reference ">
      <xsd:attribute name="href" type=" xsd:IDREF " use="required" reference="AdjustableDates" />
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

top

**Complex Type: CalculationPeriodsScheduleReference**

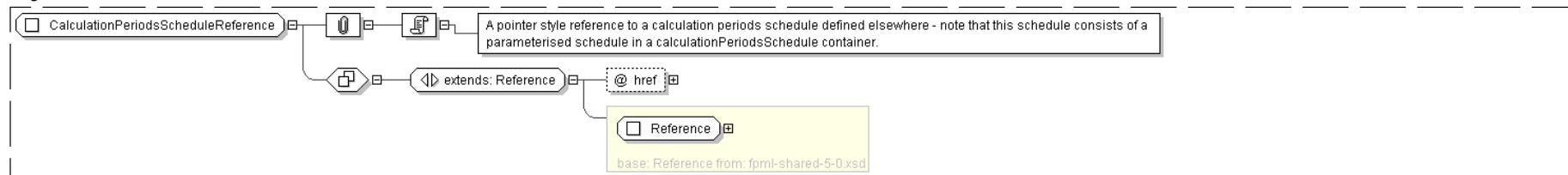
**Super-types:** [Reference](#) < CalculationPeriodsScheduleReference (by extension)

Sub-types:	None
------------	------

Name	CalculationPeriodsScheduleReference
Used by (from the same schema document)	Model Group <a href="#">CommodityDeliveryPeriodsPointer.model</a> , Model Group <a href="#">CommodityCalculationPeriodsPointer.model</a>
Abstract	no
Documentation	A pointer style reference to a calculation periods schedule defined elsewhere - note that this schedule consists of a parameterised schedule in a calculationPeriodsSchedule container.

**XML Instance Representation**

```
<...>
  href="# IDREF [1]" />
```

**Diagram****Schema Component Representation**

```
<xsd:complexType name="CalculationPeriodsScheduleReference">
  <xsd:complexContent>
    <xsd:extension base=" Reference ">
      <xsd:attribute name="href" type=" xsd:IDREF "
        use=" required" reference="CommodityCalculationPeriodsSchedule" />
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

top

**Complex Type: CoalAttributeDecimal**

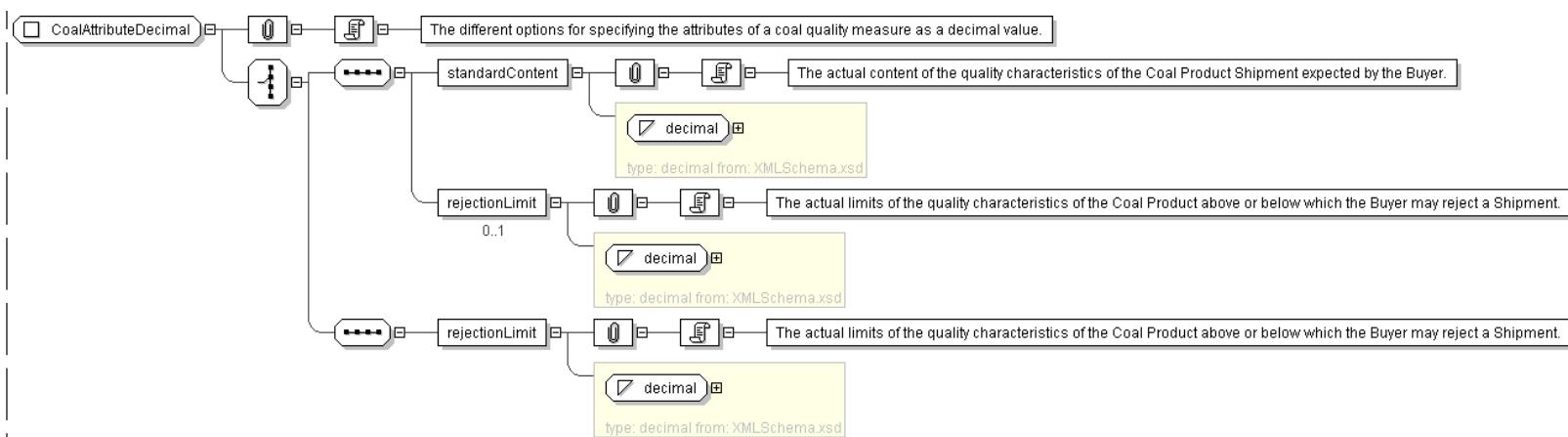
Super-types:	None
Sub-types:	None

Name	CoalAttributeDecimal
Used by (from the same schema document)	Model Group <a href="#">CommodityCoalProperties.model</a> , Model Group <a href="#">CommodityCoalReducingAtmosphere.model</a>
Abstract	no
Documentation	The different options for specifying the attributes of a coal quality measure as a decimal value.

**XML Instance Representation**

```
<...>
Start Choice [1]
  <standardContent> xsd:decimal </standardContent> [1]
  'The actual content of the quality characteristics of the Coal Product Shipment expected by the Buyer.'
  <rejectionLimit> xsd:decimal </rejectionLimit> [0..1]
  'The actual limits of the quality characteristics of the Coal Product above or below which the Buyer may reject a Shipment.'
  <rejectionLimit> xsd:decimal </rejectionLimit> [1]
  'The actual limits of the quality characteristics of the Coal Product above or below which the Buyer may reject a Shipment.'
End Choice
</...>
```

**Diagram**

**Schema Component Representation**

```
<xsd:complexType name="CoalAttributeDecimal">
  <xsd:choice>
    <xsd:sequence>
      <xsd:element name="standardContent" type="xsd:string" />
      <xsd:element name="rejectionLimit" type="xsd:decimal" minOccurs="0" />
    </xsd:sequence>
    <xsd:sequence>
      <xsd:element name="rejectionLimit" type="xsd:decimal" />
    </xsd:sequence>
  </xsd:choice>
</xsd:complexType>
```

[top](#)**Complex Type: CoalAttributePercentage**

Super-types:

None

Sub-types:

None

**Name**

CoalAttributePercentage

**Used by (from the same schema document)**

Model Group [CommodityCoalComposition.model](#), Model Group [CommodityCoalComposition.model](#), Model Group [CommodityCoalComposition.model](#), Model Group [CommodityCoalComposition.model](#), Model Group [CommodityCoalComposition.model](#)

**Abstract**

no

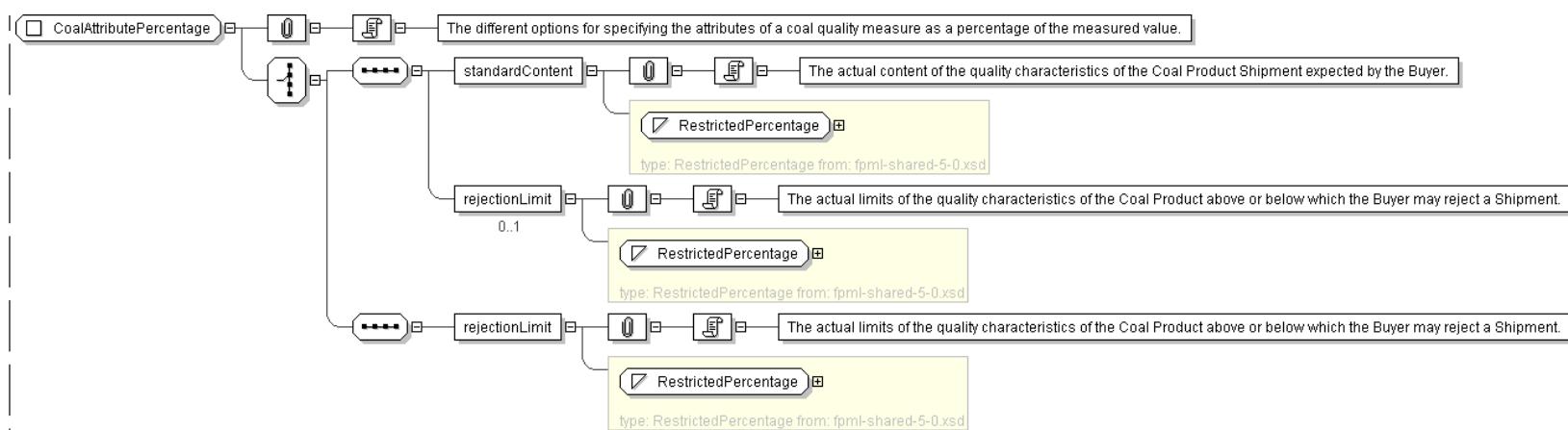
**Documentation**

The different options for specifying the attributes of a coal quality measure as a percentage of the measured value.

**XML Instance Representation**

```
<...>
Start Choice [1]
  <standardContent> RestrictedPercentage </standardContent> [1]
  'The actual content of the quality characteristics of the Coal Product Shipment expected by the Buyer.'
  <rejectionLimit> RestrictedPercentage </rejectionLimit> [0..1]
  'The actual limits of the quality characteristics of the Coal Product above or below which the Buyer may reject a Shipment.'
  <rejectionLimit> RestrictedPercentage </rejectionLimit> [1]
  'The actual limits of the quality characteristics of the Coal Product above or below which the Buyer may reject a Shipment.'
End Choice
<...>
```

**Diagram**

**Schema Component Representation**

```

<xsd:complexType name="CoalAttributePercentage">
  <xsd:choice>
    <xsd:sequence>
      <xsd:element name="standardContent" type="RestrictedPercentage" />
      <xsd:element name="rejectionLimit" type="RestrictedPercentage" minOccurs="0" />
    </xsd:sequence>
    <xsd:sequence>
      <xsd:element name="rejectionLimit" type="RestrictedPercentage" />
    </xsd:sequence>
  </xsd:choice>
</xsd:complexType>

```

[top](#)**Complex Type: CoalDelivery**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None

<b>Name</b>	CoalDelivery
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">CoalPhysicalLeg</a>
<b>Abstract</b>	no
<b>Documentation</b>	The physical delivery conditions for coal.

**XML Instance Representation**

```

<...>
Start Choice [1]
  <deliveryPoint> CoalDeliveryPoint </deliveryPoint> [1]
  'The point at which the Coal Product will be delivered and received.'

  <deliveryAtSource> xsd:boolean </deliveryAtSource> [1]
  'The point at which the Coal Product as a reference to the Source of the Coal Product.
  This should be a reference to the source element within product.'

End Choice
Start Group: CommodityUSCoalDelivery.model [0..1]
'Additional delivery details for U.S. Coal transactions.'

  <quantityVariationAdjustment> xsd:boolean </quantityVariationAdjustment> [1]
  'If true, indicates that QVA is applicable. If false, indicates that QVA is inapplicable.'

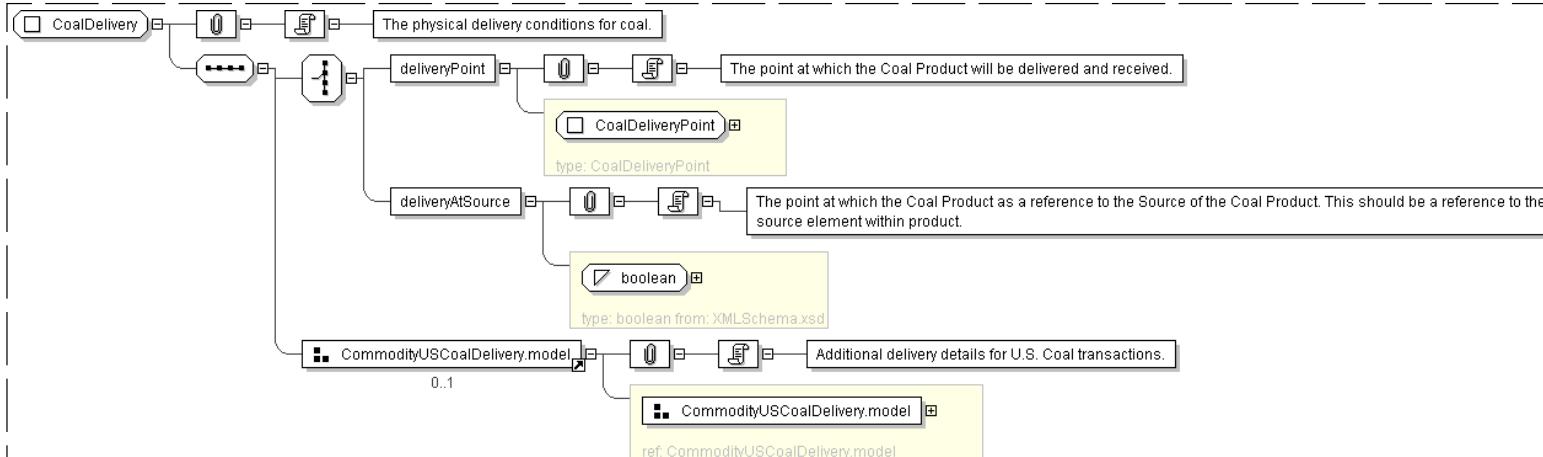
  <transportationEquipment> CoalTransportationEquipment </transportationEquipment> [0..1]
  'The transportation equipment with which the Coal Product will be delivered and received.'

  <risk> CommodityDeliveryRisk </risk> [0..1]
  'Specifies how the risk associated with the delivery is assigned.'

```

End Group: [CommodityUSCoalDelivery.model](#)

&lt;/...&gt;

**Diagram****Schema Component Representation**

```

<xsd:complexType name="CoalDelivery">
  <xsd:sequence>
    <xsd:choice>
      <xsd:element name="deliveryPoint" type=" CoalDeliveryPoint " />
      <xsd:element name="deliveryAtSource" type=" xsd:boolean " />
    </xsd:choice>
    <xsd:group ref=" CommodityUSCoalDelivery.model " minOccurs="0 " />
  </xsd:sequence>
</xsd:complexType>
  
```

top

**Complex Type: CoalDeliveryPoint**

Super-types:

Scheme &lt; CoalDeliveryPoint (by extension)

Sub-types:

None

**Name**

CoalDeliveryPoint

**Used by (from the same schema document)**Complex Type [CoalDelivery](#)**Abstract**

no

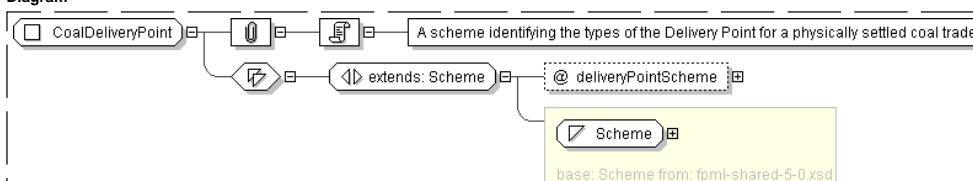
**Documentation**

A scheme identifying the types of the Delivery Point for a physically settled coal trade.

**XML Instance Representation**

```

<...
  deliveryPointScheme=" xsd:anyURI [0..1]">
  Scheme
</...>
  
```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="CoalDeliveryPoint">
  <xsd:simpleContent>
    <xsd:extension base=" Scheme ">
      <xsd:attribute name="deliveryPointsScheme" type=" xsd:anyURI " />
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>

```

top

## Complex Type: [CoalPhysicalLeg](#)

**Super-types:** [Leg](#) < [PhysicalLeg](#) (by extension) < [CoalPhysicalLeg](#) (by extension)

**Sub-types:** None

<b>Name</b>	CoalPhysicalLeg
-------------	-----------------

<b>Used by (from the same schema document)</b>	Complex Type <a href="#">CommoditySwap</a>
--	--

<b>Abstract</b>	no
-----------------	----

<b>Documentation</b>	Physically settled leg of a physically settled coal transaction.
----------------------	--

### XML Instance Representation

```

<...>
  id=" xsd:ID [0..1]>
  <payerPartyReference> PartyReference </payerPartyReference> [1]
  'A reference to the party responsible for making the payments defined by this structure.'

  <payerAccountReference> AccountReference </payerAccountReference> [0..1]
  'A reference to the account responsible for making the payments defined by this structure.'

  <receiverPartyReference> PartyReference </receiverPartyReference> [1]
  'A reference to the party that receives the payments corresponding to this structure.'

  <receiverAccountReference> AccountReference </receiverAccountReference> [0..1]
  'A reference to the account that receives the payments corresponding to this structure.'

  <deliveryPeriods> CommodityDeliveryPeriods </deliveryPeriods> [1]
  'The period during which delivery/deliveries of Coal Products may be scheduled. Equivalent to Nomination Period(s) for US Coal.'

  <coal> CoalProduct </coal> [1]
  'The specification of the Coal Product to be delivered.'

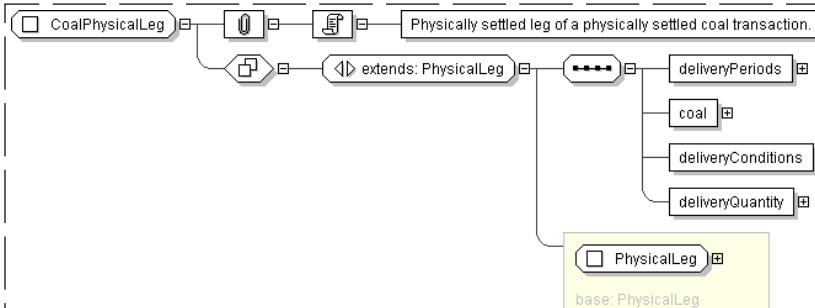
  <deliveryConditions> CoalDelivery </deliveryConditions> [1]
  'The physical delivery conditions for the transaction.'

  <deliveryQuantity> CommodityPhysicalQuantity </deliveryQuantity> [1]
  'The different options for specifying the quantity.'

</...>

```

### Diagram



### Schema Component Representation

```

<xsd:complexType name="CoalPhysicalLeg">
  <xsd:complexContent>
    <xsd:extension base=" PhysicalLeg ">
      <xsd:sequence>
        <xsd:element name="deliveryPeriods" type=" CommodityDeliveryPeriods " />
        <xsd:element name="coal" type=" CoalProduct " />
        <xsd:element name="deliveryConditions" type=" CoalDelivery " />
        <xsd:element name="deliveryQuantity" type=" CommodityPhysicalQuantity " />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

[top](#)

## Complex Type: [CoalProduct](#)

<b>Super-types:</b>	None
<b>Sub-types:</b>	None

<b>Name</b>	CoalProduct
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">CoalPhysicalLeg</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type defining the characteristics of the coal being traded in a physically settled gas transaction.

### XML Instance Representation

```

<...>
Start Choice [1]
  <type> CoalProductType </type> [1]
  'The type of coal product to be delivered by reference to a pre-defined specification.'

  <coalProductSpecifications> CoalProductSpecifications </coalProductSpecifications> [1]
  'The type of coal product to be delivered specified in full.'

End Choice
<source> CoalProductSource </source> [1..*]
'The mining region, mine(s), mining complex(es), loadout(s) or river dock(s) or other point
(s) of origin that Seller and Buyer agree are acceptable origins for the Coal Product.
For International Coal transactions, this is the Origin of the Coal Product.'

Start Group: CommodityUSCoalProduct.model [0..1]
'Additional product details for U.S. Coal transactions.'

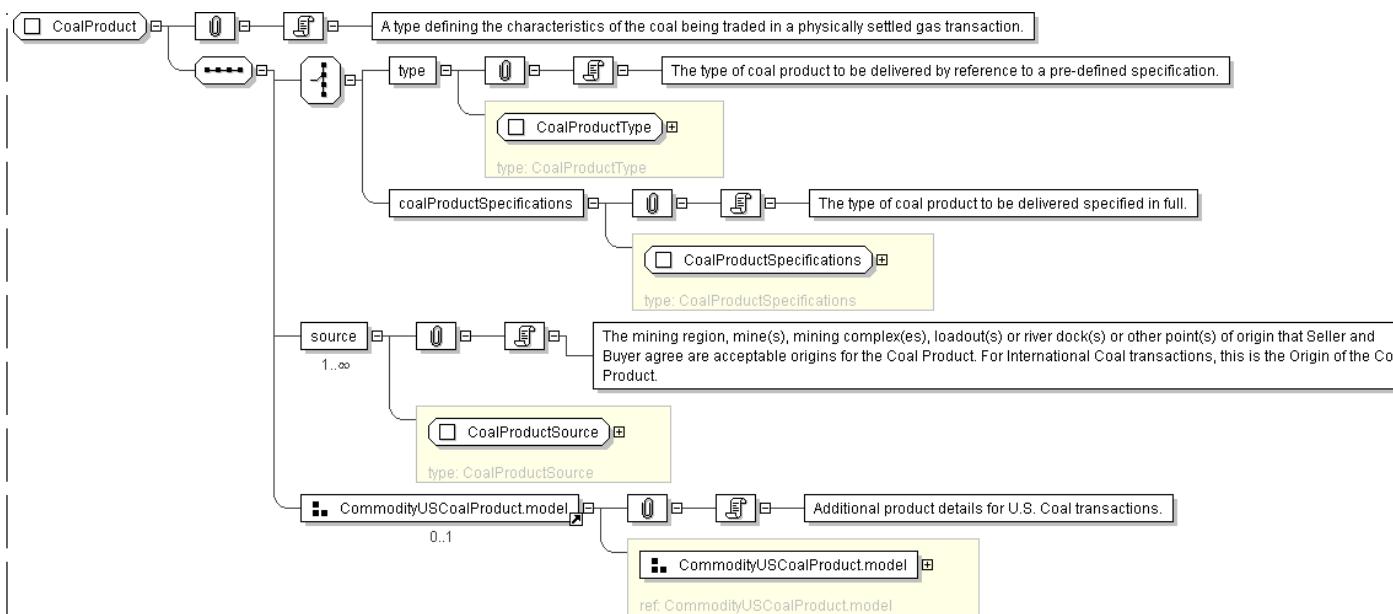
  <btuQualityAdjustment> CoalQualityAdjustments </btuQualityAdjustment> [1]
  'The Quality Adjustment formula to be used where the Actual Shipment BTU/Lb value differs
  from the Standard BTU/Lb value.'

  <so2QualityAdjustment> CoalQualityAdjustments </so2QualityAdjustment> [0..1]
  'The Quality Adjustment formula to be used where the Actual Shipment SO2/MMBTU value
  differs from the Standard SO2/MMBTU value.'

End Group: CommodityUSCoalProduct.model
</...>

```

### Diagram

**Schema Component Representation**

```

<xsd:complexType name="CoalProduct">
  <xsd:sequence>
    <xsd:choice>
      <xsd:element name="type" type="CoalProductType" />
      <xsd:element name="coalProductSpecifications" type="CoalProductSpecifications" />
    </xsd:choice>
    <xsd:element name="source" type="CoalProductSource" maxOccurs="unbounded" />
    <xsd:group ref=" CommodityUSCoalProduct.model " minOccurs="0" />
  </xsd:sequence>
</xsd:complexType>

```

top

**Complex Type: CoalProductSource**

<b>Super-types:</b>	<a href="#">Scheme</a> < <b>CoalProductSource</b> (by extension)
<b>Sub-types:</b>	None

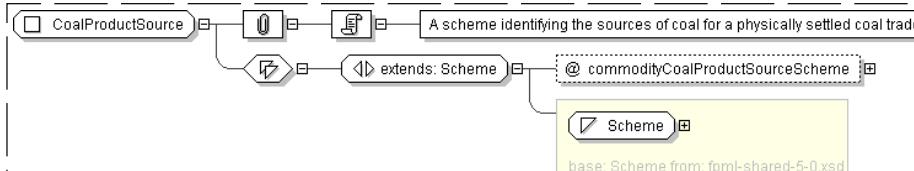
<b>Name</b>	CoalProductSource
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">CoalProduct</a>
<b>Abstract</b>	no
<b>Documentation</b>	A scheme identifying the sources of coal for a physically settled coal trade.

**XML Instance Representation**

```

<...
  commodityCoalProductSourceScheme=" xsd:anyURI [0..1]">
  Scheme
</...>

```

**Diagram**

**Schema Component Representation**

```
<xsd:complexType name="CoalProductSource">
  <xsd:simpleContent>
    <xsd:extension base="#_Scheme "#>
      <xsd:attribute name="commodityCoalProductSourceScheme" type="xsd:anyURI" default="http://www.fpml.org/coding-scheme/commodity-coal-product-source"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
```

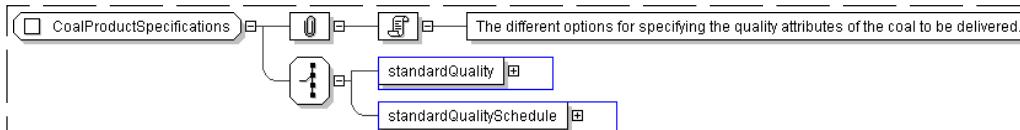
[top](#)**Complex Type: CoalProductSpecifications**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None

<b>Name</b>	CoalProductSpecifications
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">CoalProduct</a>
<b>Abstract</b>	no
<b>Documentation</b>	The different options for specifying the quality attributes of the coal to be delivered.

**XML Instance Representation**

```
<...>
Start Choice [1]
  <standardQuality> CoalStandardQuality </standardQuality> [1]
  <standardQualitySchedule> CoalStandardQualitySchedule </standardQualitySchedule> [1]
End Choice
<...>
```

**Diagram****Schema Component Representation**

```
<xsd:complexType name="CoalProductSpecifications">
  <xsd:choice>
    <xsd:element name="standardQuality" type="CoalStandardQuality" />
    <xsd:element name="standardQualitySchedule" type="CoalStandardQualitySchedule" />
  </xsd:choice>
</xsd:complexType>
```

[top](#)**Complex Type: CoalProductType**

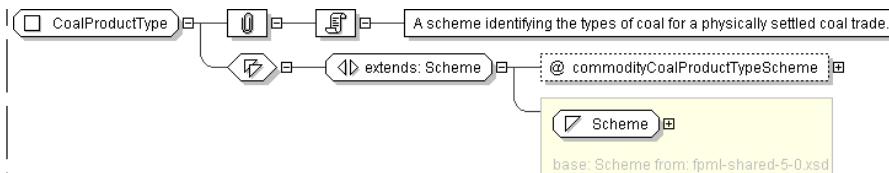
<b>Super-types:</b>	<a href="#">Scheme</a> < CoalProductType (by extension)
<b>Sub-types:</b>	None

<b>Name</b>	CoalProductType
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">CoalProduct</a>
<b>Abstract</b>	no
<b>Documentation</b>	A scheme identifying the types of coal for a physically settled coal trade.

**XML Instance Representation**

```
<...
  commodityCoalProductTypeScheme=" xsd:anyURI [0..1]">
  Scheme
</...>
```

**Diagram**

**Schema Component Representation**

```
<xsd:complexType name="CoalProductType">
  <xsd:simpleContent>
    <xsd:extension base="#Schema">
      <xsd:attribute name="commodityCoalProductTypeScheme" type="xsd:anyURI" default="http://www.fpml.org/coding-scheme/commodity-coal-product-type"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
```

top

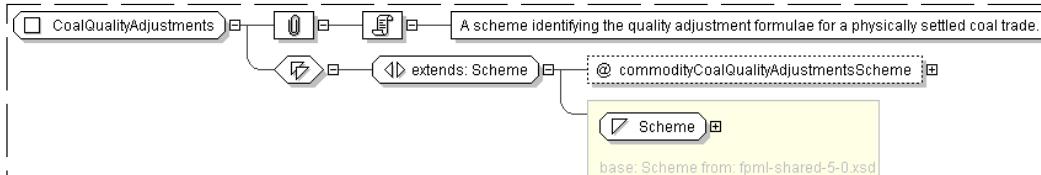
**Complex Type: CoalQualityAdjustments**

**Super-types:** [Scheme](#) < CoalQualityAdjustments (by extension)  
**Sub-types:** None

<b>Name</b>	CoalQualityAdjustments
<b>Used by (from the same schema document)</b>	Model Group <a href="#">CommodityUSCoalProduct.model</a> , Model Group <a href="#">CommodityUSCoalProduct.model</a>
<b>Abstract</b>	no
<b>Documentation</b>	A scheme identifying the quality adjustment formulae for a physically settled coal trade.

**XML Instance Representation**

```
<...>
  <xsd:element name="commodityCoalQualityAdjustments" type="commodityCoalQualityAdjustmentsScheme" minOccurs="0">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="Scheme" type="anyURI"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
<...>
```

**Diagram****Schema Component Representation**

```
<xsd:complexType name="CoalQualityAdjustments">
  <xsd:simpleContent>
    <xsd:extension base="#Schema">
      <xsd:attribute name="commodityCoalQualityAdjustmentsScheme" type="xsd:anyURI" default="http://www.fpml.org/coding-scheme/commodity-coal-quality-adjustments"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
```

top

**Complex Type: CoalStandardQuality**

**Super-types:** None  
**Sub-types:** None

<b>Name</b>	CoalStandardQuality
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">CoalProductSpecifications</a> , Complex Type <a href="#">CoalStandardQualitySchedule</a>

**Abstract**

no

**Documentation**

The quality attributes of the coal to be delivered.

**XML Instance Representation**

```
<...>
<moisture> CoalAttributePercentage </moisture> [0..1]
'The moisture content of the coal product.'

<ash> CoalAttributePercentage </ash> [0..1]
'The ash content of the coal product.'

<sulfur> CoalAttributePercentage </sulfur> [0..1]
'The sulfur/sulphur content of the coal product.'

<SO2> CoalAttributePercentage </SO2> [0..1]
'The sulfur/sulphur dioxide content of the coal product.'

<volatile> CoalAttributePercentage </volatile> [0..1]
'The volatile content of the coal product.'

<BTUpperLB> CoalAttributeDecimal </BTUpperLB> [0..1]
'The number of British Thermal Units per Pound of the coal product.'

<topSize> CoalAttributeDecimal </topSize> [0..1]
'The smallest sieve opening that will result in less than 5% of a sample of the coal product remaining.'

<finesPassingScreen> CoalAttributeDecimal </finesPassingScreen> [0..1]
<grindability> CoalAttributeDecimal </grindability> [0..1]
'The Hardgrove Grindability Index value of the coal to be delivered.'

<ashFusionTemperature> CoalAttributeDecimal </ashFusionTemperature> [0..1]
'The temperature at which the ash form of the coal product fuses completely in accordance with the ASTM International D1857 Standard Test Methodology.'

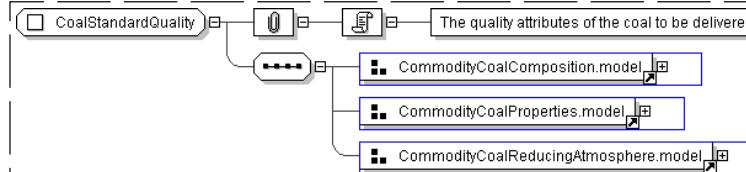
<initialDeformation> CoalAttributeDecimal </initialDeformation> [0..1]
'The temperature at which an ash cone shows evidence of deformation.'

<softeningHeightWidth> CoalAttributeDecimal </softeningHeightWidth> [0..1]
'The temperature at which the height of an ash cone equals its width. (Softening temperature).'

<softeningHeightHalfWidth> CoalAttributeDecimal </softeningHeightHalfWidth> [0..1]
'The temperature at which the height of an ash cone equals half its width. (Hemisphere temperature).'

<fluid> CoalAttributeDecimal </fluid> [0..1]
'The temperature at which the ash cone flattens.'

</...>
```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="CoalStandardQuality">
  <xsd:sequence>
    <xsd:group ref="#CommodityCoalComposition.model" />
    <xsd:group ref="#CommodityCoalProperties.model" />
    <xsd:group ref="#CommodityCoalReducingAtmosphere.model" />
  </xsd:sequence>
</xsd:complexType>
  
```

## Complex Type: CoalStandardQualitySchedule

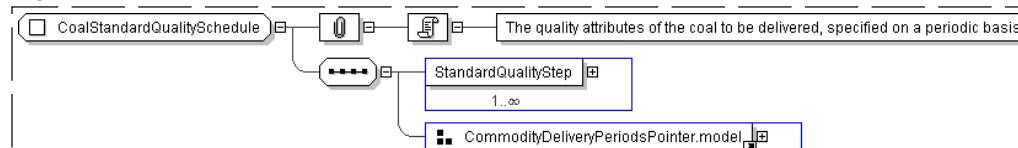
**Super-types:** None  
**Sub-types:** None

<b>Name</b>	CoalStandardQualitySchedule
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">CoalProductSpecifications</a>
<b>Abstract</b>	no
<b>Documentation</b>	The quality attributes of the coal to be delivered, specified on a periodic basis.

### XML Instance Representation

```
<...>
  <StandardQualityStep> CoalStandardQuality </StandardQualityStep> [1..*]
  Start Choice [1]
    <deliveryPeriodsReference> CalculationPeriodsReference </deliveryPeriodsReference> [1]
      'A pointer style reference to the Delivery Periods defined elsewhere.'
    <deliveryPeriodsScheduleReference> CalculationPeriodsScheduleReference
    </deliveryPeriodsScheduleReference> [1]
      'A pointer style reference to the Calculation Periods Schedule defined elsewhere.'
  End Choice
</...>
```

### Diagram



### Schema Component Representation

```

<xsd:complexType name="CoalStandardQualitySchedule">
  <xsd:sequence>
    <xsd;element name="StandardQualityStep" type=" CoalStandardQuality " maxOccurs="unbounded" />
    <xsd:group ref=" CommodityDeliveryPeriodsPointer.model " />
  </xsd:sequence>
</xsd:complexType>
  
```

## Complex Type: CoalTransportationEquipment

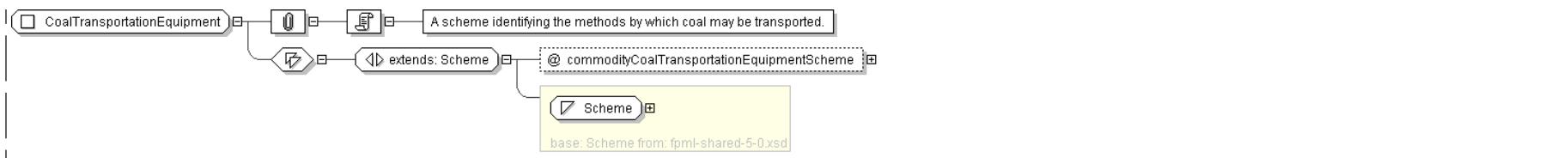
**Super-types:** [Scheme](#) < CoalTransportationEquipment (by extension)  
**Sub-types:** None

<b>Name</b>	CoalTransportationEquipment
<b>Used by (from the same schema document)</b>	Model Group <a href="#">CommodityUSCoalDelivery.model</a>
<b>Abstract</b>	no
<b>Documentation</b>	A scheme identifying the methods by which coal may be transported.

### XML Instance Representation

```
<...
  commodityCoalTransportationEquipmentScheme=" xsd:anyURI [0..1]">
  Scheme
</...>
```

### Diagram

**Schema Component Representation**

```

<xsd:complexType name="CoalTransportationEquipment">
  <xsd:simpleContent>
    <xsd:extension base="#commodityCoalTransportationEquipmentScheme">
      <xsd:attribute name="commodityCoalTransportationEquipmentScheme" type="xsd:anyURI"
        " default="http://www.fpml.org/coding-scheme/commodity-coal-transportation-equipment"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
  
```

top

**Complex Type: CommodityAmericanExercise**

**Super-types:** [SharedAmericanExercise](#) < **CommodityAmericanExercise** (by extension)  
**Sub-types:** None

<b>Name</b>	CommodityAmericanExercise
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">CommodityExercise</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type for defining exercise procedures associated with an American style exercise of a commodity option. This entity inherits from the type SharedAmericanExercise.

**XML Instance Representation**

```

<...
  id="#ID [0..1]">
  <commencementDate> AdjustableOrRelativeDate </commencementDate> [1]
  'The first day of the exercise period for an American style option.'

  <expirationDate> AdjustableOrRelativeDate </expirationDate> [1]
  'The last day within an exercise period for an American style option. For a European
  style option it is the only day within the exercise period.'

  Start Choice [0..1]
  'Choice between latest exercise time expressed as literal time, or using a
  determination method.'

    <latestExerciseTime> BusinessCenterTime </latestExerciseTime> [1]
    'For a Bermuda or American style option, the latest time on an exercise business day
    (excluding the expiration date) within the exercise period that notice can be given by
    the 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.'

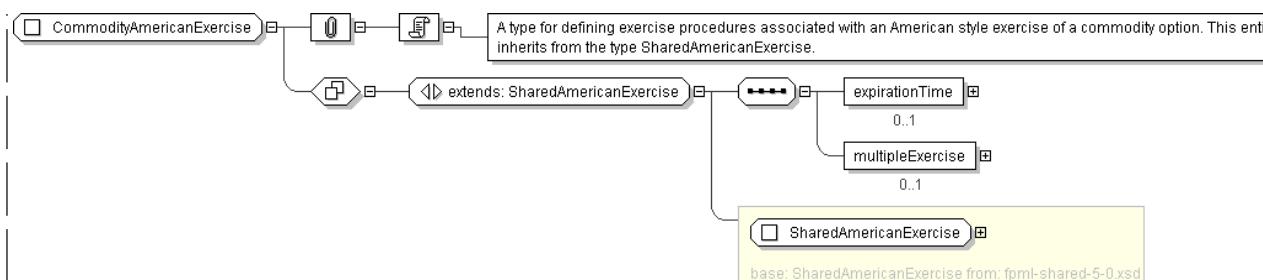
    <latestExerciseTimeDetermination> DeterminationMethod </latestExerciseTimeDetermination> [1]
    'Latest exercise time determination method.'

  End Choice
  <expirationTime> BusinessCenterTime </expirationTime> [0..1]
  'The specific time of day on which the option expires.'

  <multipleExercise> CommodityMultipleExercise </multipleExercise> [0..1]
  'The presence of this element indicates that the option may be partially exercised. It is
  not applicable to European or Asian options.'

</...>
  
```

**Diagram**



## Schema Component Representation

```

<xsd:complexType name="CommodityAmericanExercise">
  <xsd:complexContent>
    <xsd:extension base=" SharedAmericanExercise ">
      <xsd:sequence>
        <xsd:element name="expirationTime" type=" BusinessCenterTime " minOccurs="0" />
        <xsd:element name="multipleExercise" type=" CommodityMultipleExercise " minOccurs="0" />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

[top](#)

## Complex Type: CommodityCalculationPeriodsSchedule

Super-types:	<a href="#">Frequency</a> < CommodityCalculationPeriodsSchedule (by extension)
Sub-types:	None

Name	CommodityCalculationPeriodsSchedule
Used by (from the same schema document)	Complex Type <a href="#">CommodityDeliveryPeriods</a> , Model Group <a href="#">CommodityAsian.model</a> , Model Group <a href="#">CommodityCalculationPeriods.model</a>
Abstract	no
Documentation	A parametric representation of the Calculation Periods for an Asian option or a leg of a swap. In case the calculation frequency is of value T (term), the period is defined by the commoditySwapEffectiveDate and the commoditySwapTerminationDate.

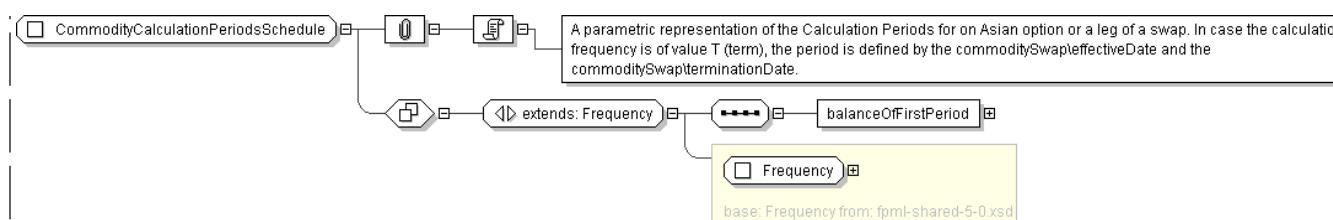
## XML Instance Representation

```

<...
  id=" xsd:ID [0..1]">
  <periodMultiplier> xsd:integer </periodMultiplier> [1]
  '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> PeriodExtendedEnum </period> [1]
  '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).'
  <balanceOfFirstPeriod> xsd:boolean </balanceOfFirstPeriod> [1]
  'If true, indicates that that the first Calculation Period should run from the Effective
  Date to the end of the calendar period in which the Effective Date falls, e.g. Jan 15 - Jan
  31 if the calculation periods are one month long and Effective Date is Jan 15. If false,
  the first Calculation Period should run from the Effective Date for one whole period, e.g.
  Jan 15 to Feb 14 if the calculation periods are one month long and Effective Date is Jan 15.'
</...>

```

## Diagram

**Schema Component Representation**

```

<xsd:complexType name="CommodityCalculationPeriodsSchedule">
  <xsd:complexContent>
    <xsd:extension base=" Frequency ">
      <xsd:sequence>
        <xsd:element name="balanceOfFirstPeriod" type=" xsd:boolean " />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

top

**Complex Type: CommodityDeliveryPeriods**

<b>Super-types:</b>	None
<b>Sub-types:</b>	<ul style="list-style-type: none"> <li>• <a href="#">ElectricityDeliveryPeriods</a> (by extension)</li> <li>• <a href="#">GasDeliveryPeriods</a> (by extension)</li> </ul>

<b>Name</b>	CommodityDeliveryPeriods
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">CoalPhysicalLeg</a> , Complex Type <a href="#">ElectricityPhysicalLeg</a> , Complex Type <a href="#">OilPhysicalLeg</a>
<b>Abstract</b>	no
<b>Documentation</b>	The different options for specifying the Delivery Periods of a physical leg.

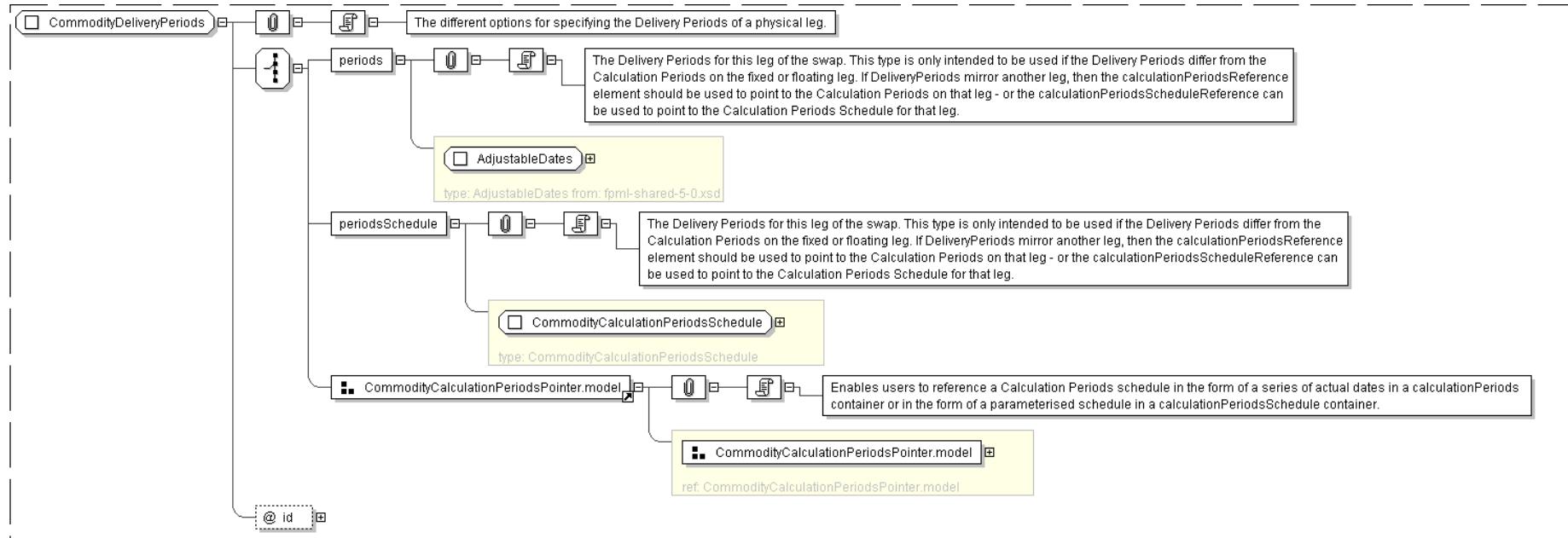
**XML Instance Representation**

```

<...
  id=" xsd:ID [0..1]">
Start Choice [1]
  <periods> AdjustableDates </periods> [1]
    'The Delivery Periods for this leg of the swap. This type is only intended to be used if
    the Delivery Periods differ from the Calculation Periods on the fixed or floating leg.
    If DeliveryPeriods mirror another leg, then the calculationPeriodsReference element should
    be used to point to the Calculation Periods on that leg - or
    the calculationPeriodsScheduleReference can be used to point to the Calculation
    Periods Schedule for that leg.'
  <periodsSchedule> CommodityCalculationPeriodsSchedule </periodsSchedule> [1]
    'The Delivery Periods for this leg of the swap. This type is only intended to be used if
    the Delivery Periods differ from the Calculation Periods on the fixed or floating leg.
    If DeliveryPeriods mirror another leg, then the calculationPeriodsReference element should
    be used to point to the Calculation Periods on that leg - or
    the calculationPeriodsScheduleReference can be used to point to the Calculation
    Periods Schedule for that leg.'
Start Choice [1]
  <calculationPeriodsReference> CalculationPeriodsReference </calculationPeriodsReference> [1]
    'A pointer style reference to the Calculation Periods defined on another leg.'
  <calculationPeriodsScheduleReference> CalculationPeriodsScheduleReference
  </calculationPeriodsScheduleReference> [1]
    'A pointer style reference to the Calculation Periods Schedule defined on another leg.'
  <calculationPeriodsDatesReference> CalculationPeriodsDatesReference
  </calculationPeriodsDatesReference> [1]
    'A pointer style reference to single-day-duration Calculation Periods defined on another leg.'
End Choice
End Choice
</...>

```

## Diagram



## Schema Component Representation

```
<xsd:complexType name="CommodityDeliveryPeriods">
  <xsd:choice>
    <xsd:element name="periods" type="AdjustableDates" />
    <xsd:element name="periodsSchedule" type="CommodityCalculationPeriodsSchedule" />
    <xsd:group ref="CommodityCalculationPeriodsPointer.model" />
  </xsd:choice>
  <xsd:attribute name="id" type="xsd:ID" />
</xsd:complexType>
```

top

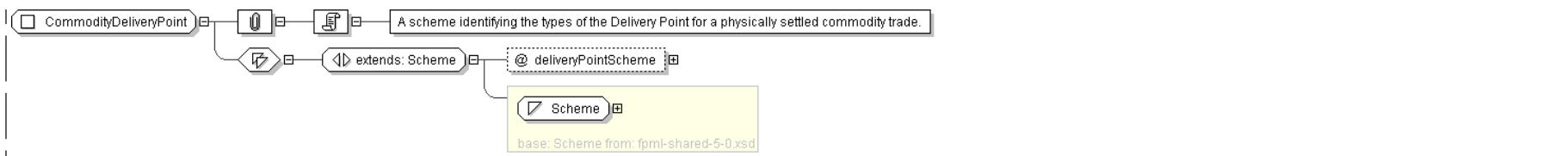
## **Complex Type: CommodityDeliveryPoint**

<b>Super-types:</b>	<a href="#">Scheme</a> < <b>CommodityDeliveryPoint</b> (by extension)
<b>Sub-types:</b>	None
<b>Name</b>	CommodityDeliveryPoint
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">ElectricityDelivery</a> , Complex Type <a href="#">ElectricityDeliverySystemFirm</a> , Complex Type <a href="#">ElectricityDeliveryUnitFirm</a> , Complex Type <a href="#">OilPipelineDelivery</a> , Complex Type <a href="#">OilPipelineDelivery</a> , Complex Type <a href="#">OilTransferDelivery</a> , Model Group <a href="#">CommodityDeliveryPoints.model</a> , Model Group <a href="#">CommodityDeliveryPoints.model</a>
<b>Abstract</b>	no
<b>Documentation</b>	A schema identifying the types of the Delivery Point for a physically settled commodity trade.

## XML Instance Representation

```
<...  
deliveryPointScheme=" xsd:anyURI [0..1]">  
Scheme  
</ ... >
```

## Diagram

**Schema Component Representation**

```

<xsd:complexType name="CommodityDeliveryPoint">
  <xsd:simpleContent>
    <xsd:extension base="#Scheme">
      <xsd:attribute name="deliveryPointScheme" type="xsd:anyURI" />
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
  
```

[top](#)**Complex Type: CommodityDeliveryRisk**

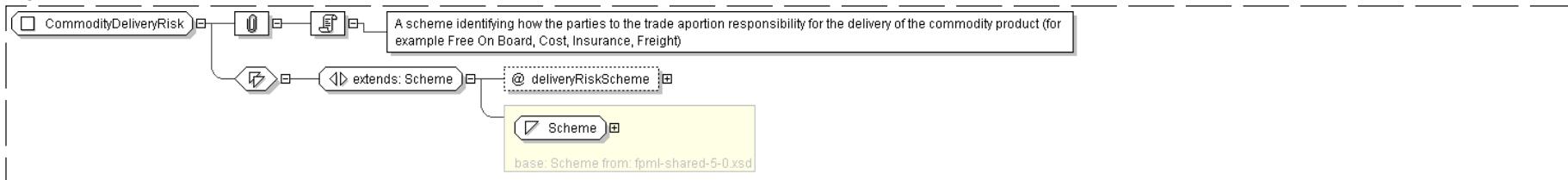
<b>Super-types:</b>	<a href="#">Scheme</a> < <b>CommodityDeliveryRisk</b> (by extension)
<b>Sub-types:</b>	None

<b>Name</b>	CommodityDeliveryRisk
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">OilPipelineDelivery</a> , Model Group <a href="#">CommodityUSCoalDelivery.model</a>
<b>Abstract</b>	no
<b>Documentation</b>	A scheme identifying how the parties to the trade apportion responsibility for the delivery of the commodity product (for example Free On Board, Cost, Insurance, Freight)

**XML Instance Representation**

```

<...
  deliveryRiskScheme="xsd:anyURI [0..1]">
  <!--
  Scheme
  -->
<...>
  
```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="CommodityDeliveryRisk">
  <xsd:simpleContent>
    <xsd:extension base="#Scheme">
      <xsd:attribute name="deliveryRiskScheme" type="xsd:anyURI" default="http://www.fpml.org/coding-scheme/external/incoterms"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
  
```

[top](#)**Complex Type: CommodityEuropeanExercise**

<b>Super-types:</b>	<a href="#">Exercise</a> < <b>CommodityEuropeanExercise</b> (by extension)
<b>Sub-types:</b>	None

<b>Name</b>	CommodityEuropeanExercise
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">CommodityExercise</a>

<b>Abstract</b>	no
<b>Documentation</b>	A type for defining exercise procedures associated with a European style exercise of a commodity option.
<b>XML Instance Representation</b>	
<pre>&lt;...&gt;   id="xsd:ID [0..1]"*   &lt;expirationDate&gt; AdjustableOrRelativeDate &lt;/expirationDate&gt; [1]   'The last day within an exercise period for an American style option. For a European   style option it is the only day within the exercise period. For an averaging option this   is equivalent to the Termination Date..'    &lt;expirationTime&gt; BusinessCenterTime &lt;/expirationTime&gt; [0..1]   'The specific time of day on which the option expires.'  &lt;/...&gt;</pre>	
<b>Diagram</b>	
<pre> classDiagram     class CommodityEuropeanExercise {         &lt;&gt; expirationDate : AdjustableOrRelativeDate         &lt;&gt; expirationTime : BusinessCenterTime     }     class Exercise {         &lt;&gt; extends : Exercise     }     CommodityEuropeanExercise &lt; -- Exercise     &lt;&gt; extends &lt; -- CommodityEuropeanExercise   </pre> <p>base: Exercise from: fpml-shared-5-0.xsd</p>	
<b>Schema Component Representation</b>	
<pre> &lt;xsd:complexType name="CommodityEuropeanExercise"&gt;   &lt;xsd:complexContent&gt;     &lt;xsd:extension base=" Exercise "&gt;       &lt;xsd:sequence&gt;         &lt;xsd:element name="expirationDate" type=" AdjustableOrRelativeDate "/&gt;         &lt;xsd:element name="expirationTime" type=" BusinessCenterTime " minOccurs="0"/&gt;       &lt;/xsd:sequence&gt;     &lt;/xsd:extension&gt;   &lt;/xsd:complexContent&gt; &lt;/xsd:complexType&gt;   </pre>	

[top](#)

## Complex Type: CommodityExercise

<b>Super-types:</b>	None
<b>Sub-types:</b>	None
<b>Name</b>	CommodityExercise
<b>Used by (from the same schema document)</b>	Model Group <a href="#">CommodityFinancialOption.model</a>
<b>Abstract</b>	no
<b>Documentation</b>	The parameters for defining how the commodity option can be exercised, how it is priced and how it is settled.

<b>XML Instance Representation</b>
<pre>&lt;...&gt; Start Choice [1]   &lt;americanExercise&gt; CommodityAmericanExercise &lt;/americanExercise&gt; [1]   'The parameters for defining the exercise period for an American style option together with   the rules governing the quantity of the commodity that can be exercised on any given   exercise date..'    &lt;europeanExercise&gt; CommodityEuropeanExercise &lt;/europeanExercise&gt; [1]   'The parameters for defining the expiration date and time for a European or Asian style   option. For an Asian style option the expiration date is equivalent to the termination date..'  End Choice &lt;automaticExercise&gt; xsd:boolean &lt;/automaticExercise&gt; [0..1]   'Specifies whether or not Automatic Exercise applies to a Commodity Option Transaction.'</pre>

```

<writtenConfirmation> xsd:boolean </writtenConfirmation> [0..1]
'Specifies whether or not Written Confirmation applies to a Commodity Option Transaction.'

<settlementCurrency> IdentifiedCurrency </settlementCurrency> [1]
'The currency into which the Commodity Option Transaction will settle. If this is not the same as the currency in which the Commodity Reference Price is quoted, then an FX determination method should also be specified.'

<fx> CommodityFx </fx> [0..1]
'FX observations to be used to convert the observed Commodity Reference Price to the Settlement Currency.'

<conversionFactor> xsd:decimal </conversionFactor> [0..1]
>If the Notional Quantity is specified in a unit that does not match the unit in which the Commodity Reference Price is quoted, the scaling or conversion factor used to convert the Commodity Reference Price unit into the Notional Quantity unit should be stated here. If there is no conversion, this element is not intended to be used.

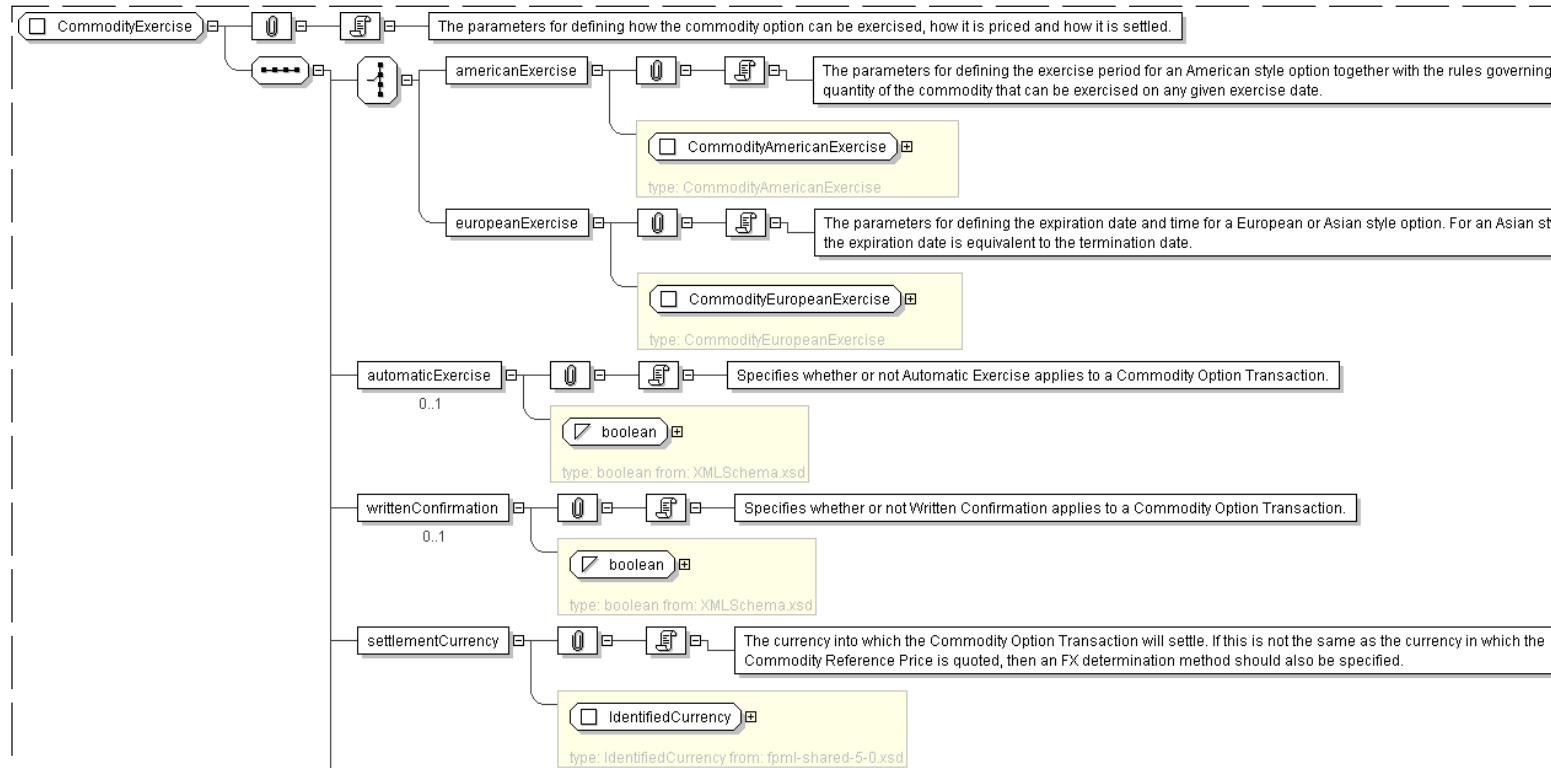
Start Choice [1]
<relativePaymentDates> CommodityRelativePaymentDates </relativePaymentDates> [1]
'The Payment Dates of the trade relative to the Calculation Periods.'

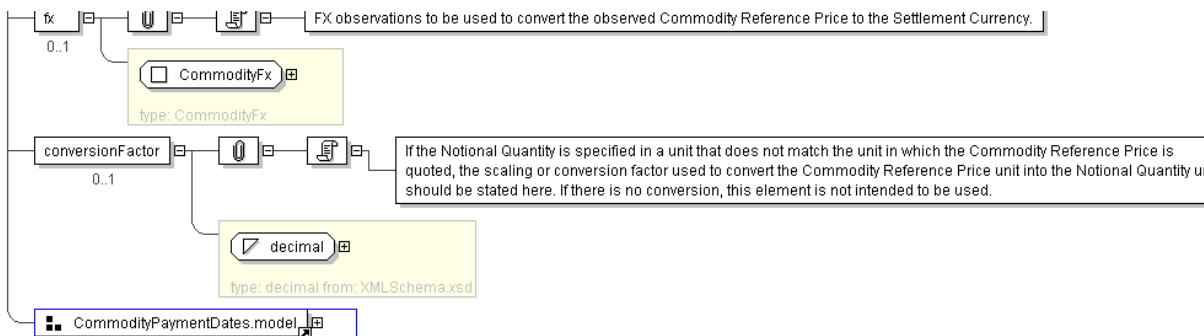
Start Choice [1]
<paymentDates> AdjustableDatesOrRelativeDateOffset </paymentDates> [1]
'Dates on which payments will be made.'

<masterAgreementPaymentDates> xsd:boolean </masterAgreementPaymentDates> [1]
'If present and true indicates that the Payment Date(s) are specified in the relevant master agreement.'

End Choice
End Choice
</...>

```

**Diagram**

**Schema Component Representation**

```
<xsd:complexType name="CommodityExercise">
  <xsd:sequence>
    <xsd:choice>
      <xsd:element name="americanExercise" type="CommodityAmericanExercise" />
      <xsd:element name="europeanExercise" type="CommodityEuropeanExercise" />
    </xsd:choice>
    <xsd:element name="automaticExercise" type="xsd:boolean" minOccurs="0"/>
    <xsd:element name="writtenConfirmation" type="xsd:boolean" minOccurs="0"/>
    <xsd:element name="settlementCurrency" type="IdentifiedCurrency" />
    <xsd:element name="fx" type="CommodityFx" minOccurs="0"/>
    <xsd:element name="conversionFactor" type="xsd:decimal" minOccurs="0"/>
    <xsd:group ref="CommodityPaymentDates.model" />
  </xsd:sequence>
</xsd:complexType>
```

top

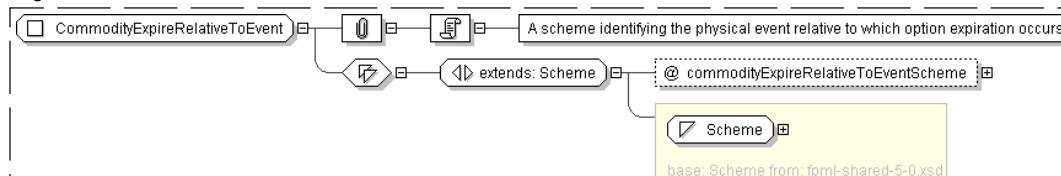
**Complex Type: CommodityExireRelativeToEvent**

Super-types:	<a href="#">Scheme</a> < CommodityExireRelativeToEvent (by extension)
Sub-types:	None

Name	CommodityExireRelativeToEvent
Used by (from the same schema document)	Complex Type <a href="#">CommodityRelativeExpirationDates</a>
Abstract	no
Documentation	A scheme identifying the physical event relative to which option expiration occurs.

**XML Instance Representation**

```
<...
  commodityExireRelativeToEventScheme="xsd:anyURI [0..1]">
  Scheme
<...>
```

**Diagram****Schema Component Representation**

```
<xsd:complexType name="CommodityExireRelativeToEvent">
  <xsd:simpleContent>
    <xsd:extension base="Scheme">
      <xsd:attribute name="commodityExireRelativeToEventScheme" type="xsd:anyURI"
        default="http://www.fpml.org/coding-scheme/commodity-expire-relative-to-event"/>
    </xsd:extension>
  </xsd:simpleContent>
```

**Complex Type: CommodityFixedPriceSchedule**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None
<b>Name</b>	CommodityFixedPriceSchedule
<b>Used by (from the same schema document)</b>	Model Group <a href="#">CommodityFixedPrice.model</a>
<b>Abstract</b>	no
<b>Documentation</b>	The Fixed Price for a given Calculation Period during the life of the trade. There must be a Fixed Price step specified for each Calculation Period, regardless of whether the Fixed Price changes or remains the same between periods.

**XML Instance Representation**

```
<...>
Start Choice [1]
<fixedPriceStep> FixedPrice </fixedPriceStep> [1..*]
'The Fixed Price for a given Calculation Period during the life of the trade. There must be
a Fixed Price step specified for each Calculation Period, regardless of whether the Fixed
Price changes or remains the same between periods.'

<worldscaleRateStep> xsd:decimal </worldscaleRateStep> [1..*]
'For a Wet Voyager Charter Freight Swap, the number of Worldscale Points for purposes of
the calculation of a Fixed Amount for a given Calculation Period during the life of the
trade. There must be Worldscale Rate Step specified for each Calculation Period, regardless
of whether the Worldscale Rate Step changes or remains the same between periods.'

<contractRateStep> NonNegativeMoney </contractRateStep> [1..*]
'For a DRY Voyage Charter or Time Charter Freight Swap, the price per relevant unit
for purposes of the calculation of a Fixed Amount for a given Calculation Period during
the life of the trade. There must be Worldscale Rate Step specified for each
Calculation Period, regardless of whether the Worldscale Rate Step changes or remains the
same between periods.'

<settlementPeriodsPriceSchedule> CommoditySettlementPeriodsPriceSchedule
</settlementPeriodsPriceSchedule> [1..*]
'For an electricity transaction, the fixed price schedule for one or more groups of
Settlement Periods on which fixed payments are based. If the schedule differs for
different groups of Settlement Periods, this element should be repeated.'

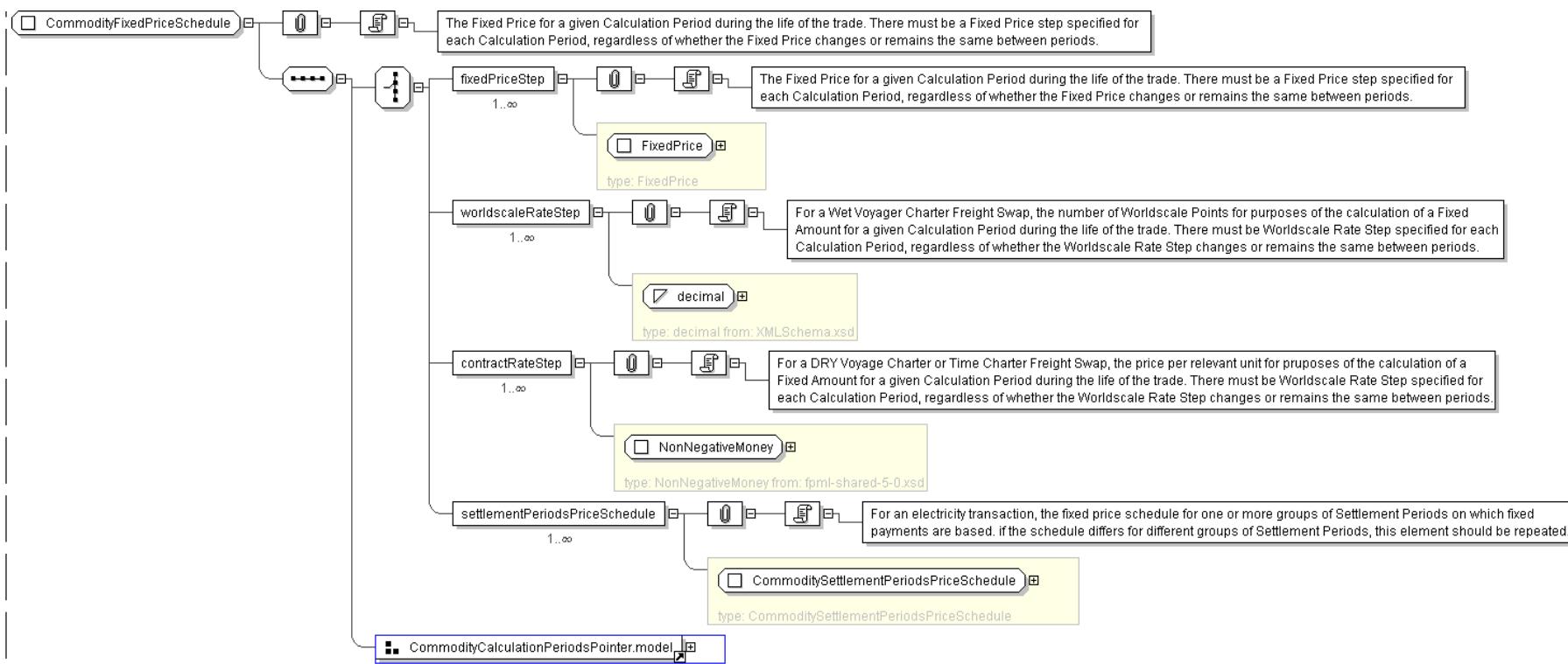
End Choice
Start Choice [1]
<calculationPeriodsReference> CalculationPeriodsReference </calculationPeriodsReference> [1]
'A pointer style reference to the Calculation Periods defined on another leg.'

<calculationPeriodsScheduleReference> CalculationPeriodsScheduleReference
</calculationPeriodsScheduleReference> [1]
'A pointer style reference to the Calculation Periods Schedule defined on another leg.'

<calculationPeriodsDatesReference> CalculationPeriodsDatesReference
</calculationPeriodsDatesReference> [1]
'A pointer style reference to single-day-duration Calculation Periods defined on another leg.'

End Choice
</...>
```

**Diagram**



#### Schema Component Representation

```

<xsd:complexType name="CommodityFixedPriceSchedule">
  <xsd:sequence>
    <xsd:choice>
      <xsd:element name="fixedPriceStep" type="#FixedPrice" maxOccurs="unbounded"/>
      <xsd:element name="worldscaleRateStep" type="xsd:decimal" maxOccurs="unbounded"/>
      <xsd:element name="contractRateStep" type="#NonNegativeMoney" maxOccurs="unbounded"/>
      <xsd:element name="settlementPeriodsPriceSchedule" type="CommoditySettlementPeriodsPriceSchedule" maxOccurs="unbounded"/>
    </xsd:choice>
    <xsd:group ref="CommodityCalculationPeriodsPointer.model"/>
  </xsd:sequence>
</xsd:complexType>
  
```

top

#### Complex Type: CommodityForward

**Super-types:** Product < CommodityForward (by extension)

**Sub-types:** None

<b>Name</b>	CommodityForward
<b>Used by (from the same schema document)</b>	Element <a href="#">commodityForward</a>
<b>Abstract</b>	no
<b>Documentation</b>	Commodity Forward

#### XML Instance Representation

```

<...
  id="#ID [0..1]>
  <productType> ProductType </productType> [0..*]
  'A classification of the type of product. FpML defines a simple product categorization using
  a coding scheme.'
  
```

```
<productId> ProductId </productId> [0..*]
'A product 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.'
```

```
<valueDate> AdjustableOrRelativeDate </valueDate> [0..1]
'Specifies the value date of the Commodity Forward Transaction. This is the day on which both the cash and the physical commodity settle.'
```

```
<fixedLeg> NonPeriodicFixedPriceLeg </fixedLeg> [1]
'The fixed leg of a Commodity Forward Transaction'
```

Start Choice [1]

'This choice group is intended to allow legs based on different classes of commodity to be added to the schema as this becomes necessary.'

```
<bullionPhysicalLeg> BullionPhysicalLeg </bullionPhysicalLeg> [1]
'The physical leg of a Commodity Forward Transaction for which the underlyer is Bullion.'
```

```
<additionalCommodityForwardLeg> ... </additionalCommodityForwardLeg> [1]
```

End Choice

Start Group: CommodityContent.model [0..1]

```
<commonPricing> xsd:boolean </commonPricing> [0..1]
'Common pricing may be relevant for a Transaction that references more than one Commodity Reference Price. If Common Pricing is not specified as applicable, it will be deemed not to apply.'
```

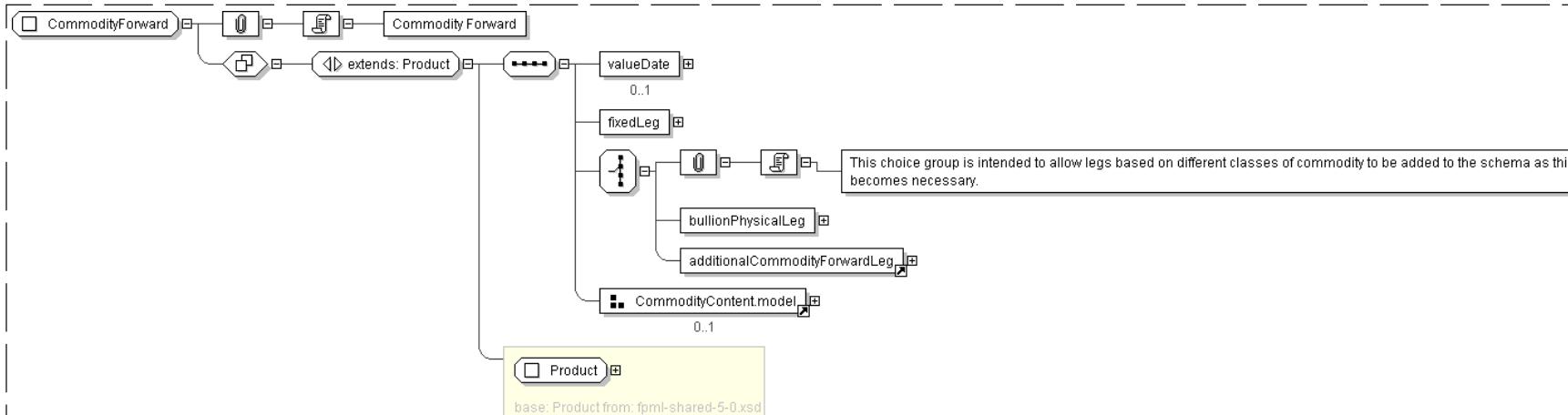
```
<marketDisruption> CommodityMarketDisruption </marketDisruption> [0..1]
'Market disruption events as defined in the ISDA 1993 Commodity Definitions or in ISDA 2005 Commodity Definitions, as applicable.'
```

```
<settlementDisruption> CommodityBullionSettlementDisruptionEnum </settlementDisruption> [0..1]
'The consequences of Bullion Settlement Disruption Events.'
```

```
<rounding> Rounding </rounding> [0..1]
'Rounding direction and precision for amounts.'
```

End Group: CommodityContent.model

## Diagram



## Schema Component Representation

```
<xsd:complexType name="CommodityForward">
  <xsd:complexContent>
    <xsd:extension base=" Product ">
      <xsd:sequence>
        <xsd:element name="valueDate" type=" AdjustableOrRelativeDate " minOccurs="0"/>
        <xsd:element name="fixedLeg" type=" NonPeriodicFixedPriceLeg " />
```

```

<xsd:choice>
  <xsd:element name="bullionPhysicalLeg" type=" BullionPhysicalLeg " />
  <xsd:element ref=" additionalCommodityForwardLeg " />
</xsd:choice>
<xsd:group ref=" CommodityContent.model " minOccurs="0 " />
</xsd:sequence>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>

```

[top](#)

## Complex Type: CommodityFrequencyType

Super-types:	<a href="#">Scheme</a> < CommodityFrequencyType (by extension)
Sub-types:	None

Name	CommodityFrequencyType
Used by (from the same schema document)	Model Group <a href="#">PricingDays.model</a>
Abstract	no
Documentation	Frequency Type for use in Pricing Date specifications.

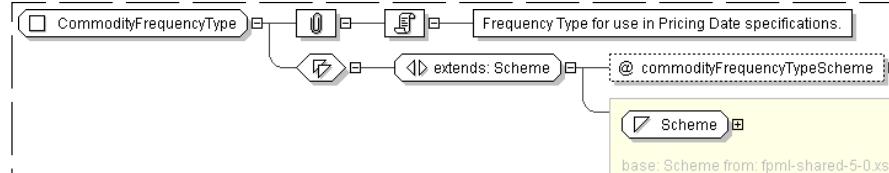
### XML Instance Representation

```

<...
  commodityFrequencyTypeScheme=" xsd:anyURI [0..1]">
  Scheme
</...>

```

### Diagram



### Schema Component Representation

```

<xsd:complexType name="CommodityFrequencyType">
  <xsd:simpleContent>
    <xsd:extension base=" Scheme ">
      <xsd:attribute name="commodityFrequencyTypeScheme" type=" xsd:anyURI " default="http://www.
        fpml.org/coding-scheme/commodity-frequency-type"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>

```

[top](#)

## Complex Type: CommodityFx

Super-types:	None
Sub-types:	None

Name	CommodityFx
Used by (from the same schema document)	Complex Type <a href="#">CommodityExercise</a> , Complex Type <a href="#">FloatingLegCalculation</a>
Abstract	no
Documentation	A type defining the FX observations to be used to convert the observed Commodity Reference Price to the Settlement Currency. The rate source must be specified. Additionally, a time for the spot price to be observed on that source may be specified, or else an averaging schedule for trades priced using an average FX rate.

### XML Instance Representation

```

<...
  <primaryRateSource> InformationSource </primaryRateSource> [1]
  'The primary source for where the rate observation will occur. Will typically be either a
  page or a reference bank published rate.'

```

```

<secondaryRateSource> InformationSource </secondaryRateSource> [0..1]
'An alternative, or secondary, source for where the rate observation will occur. Will
typically be either a page or a reference bank published rate.'

<fxType> CommodityFxType </fxType> [0..1]
'A type to identify how the FX rate will be applied. This is intended to differentiate
between the various methods for applying FX to the floating price such as a daily
calculation, or averaging the FX and applying the average at the end of each CalculationPeriod.'

<averagingMethod> AveragingMethodEnum </averagingMethod> [0..1]
'The parties may specify a Method of Averaging when averaging of the FX rate is applicable.'

Start Choice [0..1]
<fixingTime> BusinessCenterTime </fixingTime> [1]
'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. 11:00am London time.'

Start Choice [1]
<fxObservationDates> AdjustableDates </fxObservationDates> [1..*]
'A list of the fx observation dates for a given Calculation Period.'

Start Sequence [0..1]
<dayType> CommodityDayTypeEnum </dayType> [1]
'The type of day on which pricing occurs.'

Start Choice [1]
<dayDistribution> CommodityFrequencyType </dayDistribution> [1]
'The method by which the pricing days are distributed across the pricing period.'

<dayCount> xsd:positiveInteger </dayCount> [0..1]
'The number of days over which pricing should take place.'

<dayOfWeek> DayOfWeekEnum </dayOfWeek> [1..7]
'The day(s) of the week on which pricing will take place during the pricing period.'

<dayNumber> xsd:integer </dayNumber> [0..1]
'The occurrence of the dayOfWeek within the pricing period on which pricing will take place,
e.g. the 3rd Friday within each Calculation Period. If omitted, every dayOfWeek will be
a pricing day.'

End Choice
Start Group: LagOrReference.model [0..1]
Start Choice [1]
<lag> Lag </lag> [1]
'The pricing period per calculation period if the pricing days do not wholly fall within
the respective calculation period.'

<lagReference> LagReference </lagReference> [1]
Allows a lag to reference one already defined elsewhere in the trade.'

End Choice
End Group: LagOrReference.model
End Sequence
Start Choice [1]
<calculationPeriodsReference> CalculationPeriodsReference </calculationPeriodsReference> [1]
'A pointer style reference to the Calculation Periods defined on another leg.'

<calculationPeriodsScheduleReference> CalculationPeriodsScheduleReference
</calculationPeriodsScheduleReference> [1]
'A pointer style reference to the Calculation Periods Schedule defined on another leg.'

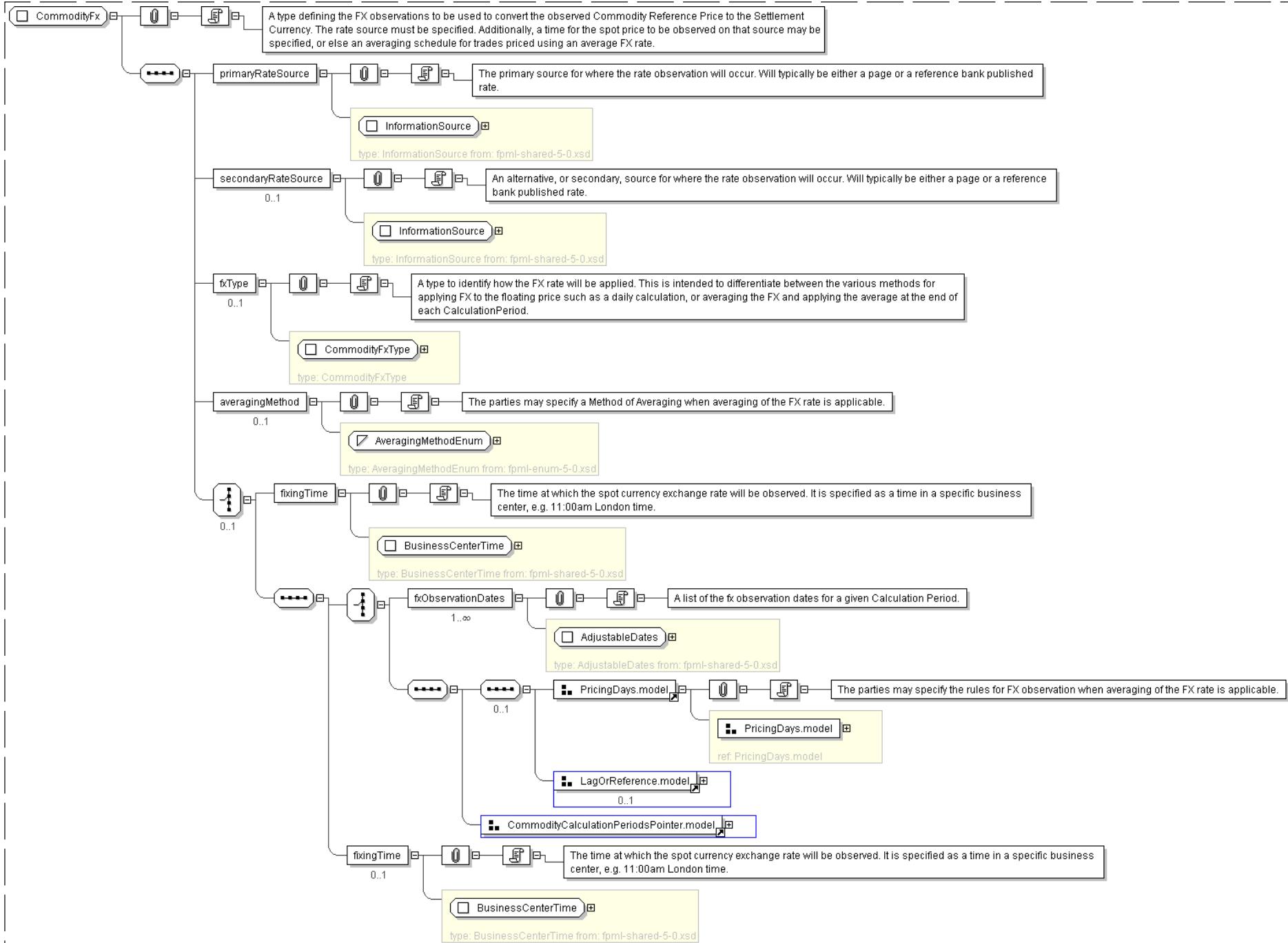
<calculationPeriodsDatesReference> CalculationPeriodsDatesReference
</calculationPeriodsDatesReference> [1]
'A pointer style reference to single-day-duration Calculation Periods defined on another leg.'

End Choice
End Choice
<fixingTime> BusinessCenterTime </fixingTime> [0..1]
'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. 11:00am London time.'

```

| End Choice  
 </...>

## Diagram



## Schema Component Representation

```
<xsd:complexType name="CommodityFx">
```

```

<xsd:sequence>
  <xsd:element name="primaryRateSource" type=" InformationSource " />
  <xsd:element name="secondaryRateSource" type=" InformationSource " minOccurs="0" />
  <xsd:element name="fxType" type=" CommodityFxType " minOccurs="0" />
  <xsd:element name="averagingMethod" type=" AveragingMethodEnum " minOccurs="0" />
  <xsd:choice minOccurs="0">
    <xsd:element name="fixingTime" type=" BusinessCenterTime " />
    <xsd:sequence>
      <xsd:choice>
        <xsd:element name="fxObservationDates" type=" AdjustableDates " maxOccurs="unbounded" />
        <xsd:sequence>
          <xsd:sequence minOccurs="0">
            <xsd:group ref=" PricingDays.model " />
            <xsd:group ref=" LagOrReference.model " minOccurs="0" />
          </xsd:sequence>
        </xsd:group ref=" CommodityCalculationPeriodsPointer.model " />
      </xsd:choice>
    </xsd:sequence>
    <xsd:element name="fixingTime" type=" BusinessCenterTime " minOccurs="0" />
  </xsd:choice>
</xsd:sequence>
</xsd:complexType>

```

[top](#)

## Complex Type: CommodityFxType

Super-types:	<a href="#">Scheme</a> < <b>CommodityFxType</b> (by extension)
Sub-types:	None

Name	CommodityFxType
Used by (from the same schema document)	Complex Type <a href="#">CommodityFx</a>
Abstract	no
Documentation	Identifies how the FX rate will be applied. This is intended to differentiate between the various methods for applying FX to the floating price such as a daily calculation, or averaging the FX and applying the average at the end of each CalculationPeriod.

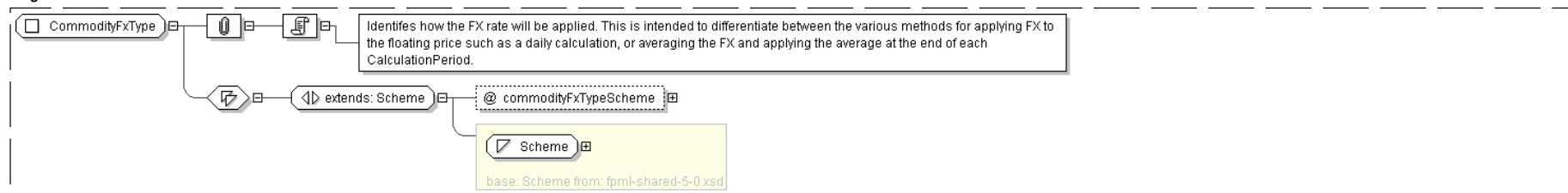
### XML Instance Representation

```

<...
  commodityFxTypeScheme=" xsd:anyURI [ 0..1 ] "gt;
  Scheme
</...>

```

### Diagram



### Schema Component Representation

```

<xsd:complexType name="CommodityFxType">
  <xsd:simpleContent>
    <xsd:extension base=" Scheme ">
      <xsd:attribute name="commodityFxTypeScheme" type=" xsd:anyURI " default="http://www.fpml.org/coding-scheme/commodity-fx-type" />
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>

```

[top](#)

## Complex Type: CommodityHub

<b>Super-types:</b>	None
<b>Sub-types:</b>	None

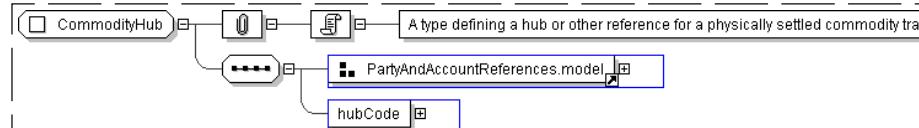
<b>Name</b>	CommodityHub
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">GasDelivery</a> , Complex Type <a href="#">GasDelivery</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type defining a hub or other reference for a physically settled commodity trade.

**XML Instance Representation**

```
<...>
<partyReference> PartyReference </partyReference> [1]
'Reference to a party.'

<accountReference> AccountReference </accountReference> [0..1]
'Reference to an account.'

<hubCode> CommodityHubCode </hubCode> [1]
</...>
```

**Diagram****Schema Component Representation**

```
<xsd:complexType name="CommodityHub">
  <xsd:sequence>
    <xsd:group ref="#PartyAndAccountReferences.model" />
    <xsd:element name="hubCode" type="CommodityHubCode" />
  </xsd:sequence>
</xsd:complexType>
```

top

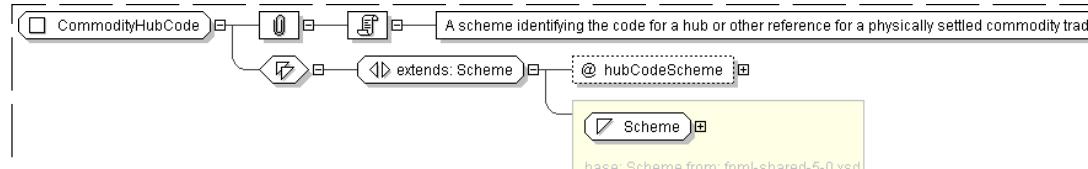
**Complex Type: CommodityHubCode**

<b>Super-types:</b>	<a href="#">Scheme</a> < CommodityHubCode (by extension)
<b>Sub-types:</b>	None

<b>Name</b>	CommodityHubCode
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">CommodityHub</a>
<b>Abstract</b>	no
<b>Documentation</b>	A scheme identifying the code for a hub or other reference for a physically settled commodity trade.

**XML Instance Representation**

```
<...
  hubCodeScheme="xsd:anyURI [1]">
  Scheme
</...>
```

**Diagram****Schema Component Representation**

```
<xsd:complexType name="CommodityHubCode">
  <xsd:simpleContent>
```

```

<xsd:extension base="Scheme">
  <xsd:attribute name="hubCodeScheme" type="xsd:anyURI" use="required"/>
</xsd:extension>
</xsd:simpleContent>
</xsd:complexType>

```

## Complex Type: CommodityMarketDisruption

<b>Super-types:</b>	None
<b>Sub-types:</b>	None
<b>Name</b>	CommodityMarketDisruption
<b>Used by (from the same schema document)</b>	Model Group <a href="#">CommodityContent.model</a>
<b>Abstract</b>	no
<b>Documentation</b>	ISDA 1993 or 2005 commodity market disruption elements.

### XML Instance Representation

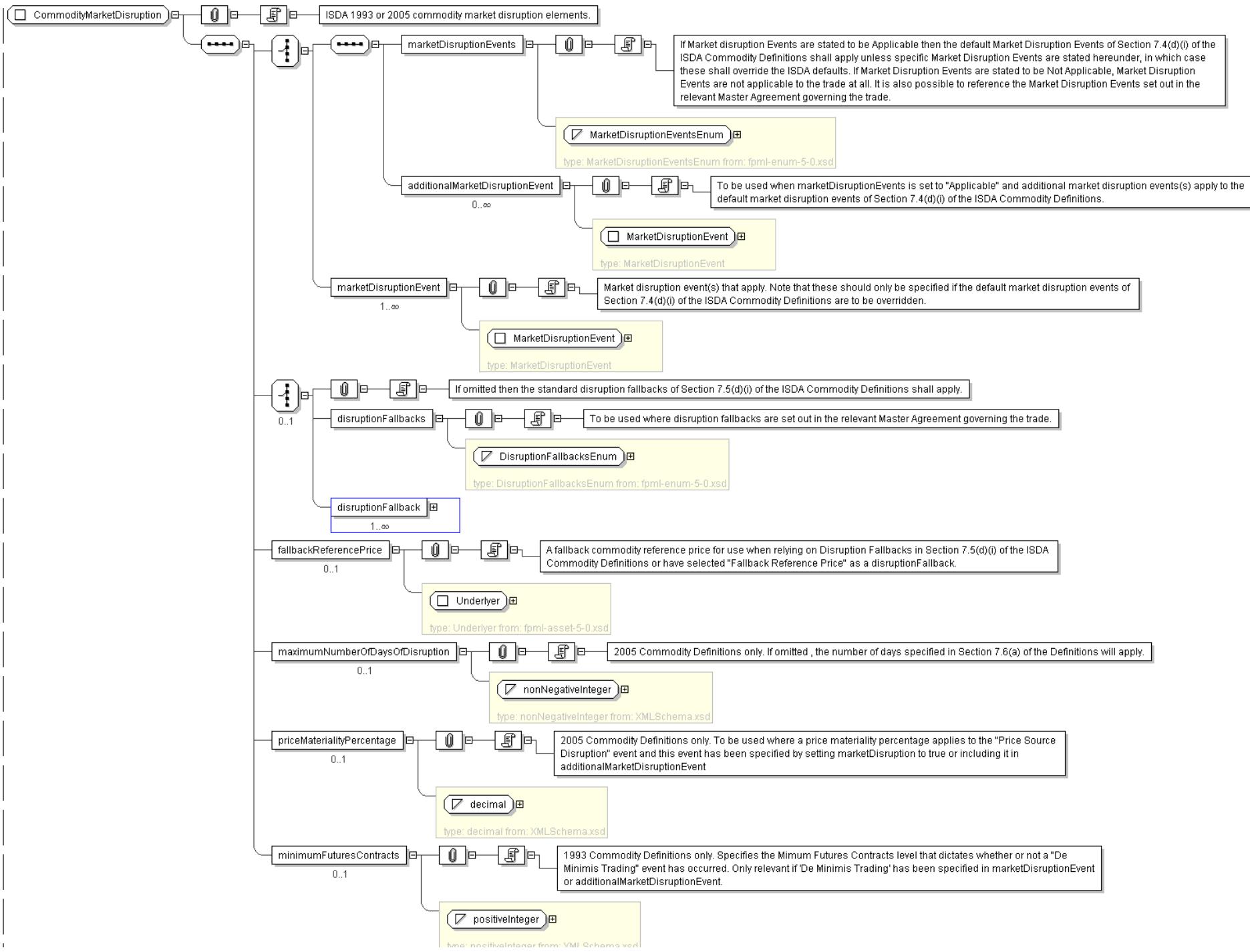
```

<...>
Start Choice [1]
  <marketDisruptionEvents> MarketDisruptionEventsEnum </marketDisruptionEvents> [1]
    'If Market disruption Events are stated to be Applicable then the default Market Disruption Events of Section 7.4(d)(i) of the ISDA Commodity Definitions shall apply unless specific Market Disruption Events are stated hereunder, in which case these shall override the ISDA defaults. If Market Disruption Events are stated to be Not Applicable, Market Disruption Events are not applicable to the trade at all. It is also possible to reference the Market Disruption Events set out in the relevant Master Agreement governing the trade.'
  <additionalMarketDisruptionEvent> MarketDisruptionEvent </additionalMarketDisruptionEvent>
  [0..*]
    'To be used when marketDisruptionEvents is set to \"Applicable\" and additional market disruption events(s) apply to the default market disruption events of Section 7.4(d)(i) of the ISDA Commodity Definitions.'
  <marketDisruptionEvent> MarketDisruptionEvent </marketDisruptionEvent> [1..*]
    'Market disruption event(s) that apply. Note that these should only be specified if the default market disruption events of Section 7.4(d)(i) of the ISDA Commodity Definitions are to be overridden.'
End Choice
Start Choice [0..1]
  'If omitted then the standard disruption fallbacks of Section 7.5(d)(i) of the ISDA Commodity Definitions shall apply.'
  <disruptionFallbacks> DisruptionFallbacksEnum </disruptionFallbacks> [1]
    'To be used where disruption fallbacks are set out in the relevant Master Agreement governing the trade.'
  <disruptionFallback> SequencedDisruptionFallback </disruptionFallback> [1..*]
End Choice
<fallbackReferencePrice> Underlyer </fallbackReferencePrice> [0..1]
  'A fallback commodity reference price for use when relying on Disruption Fallbacks in Section 7.5(d)(i) of the ISDA Commodity Definitions or have selected \"Fallback Reference Price\" as a disruptionFallback.'
<maximumNumberOfDaysOfDisruption> xsd:nonNegativeInteger </maximumNumberOfDaysOfDisruption> [0..1]
  '2005 Commodity Definitions only. If omitted , the number of days specified in Section 7.6 (a) of the Definitions will apply.'
<priceMaterialityPercentage> xsd:decimal </priceMaterialityPercentage> [0..1]
  '2005 Commodity Definitions only. To be used where a price materiality percentage applies to the \"Price Source Disruption\" event and this event has been specified by setting marketDisruption to true or including it in additionalMarketDisruptionEvent'
<minimumFuturesContracts> xsd:positiveInteger </minimumFuturesContracts> [0..1]
  '1993 Commodity Definitions only. Specifies the Mimimum Futures Contracts level that dictates whether or not a \"De Minimis Trading\" event has occurred. Only relevant if 'De Minimis Trading' has been specified in marketDisruptionEvent'

```

or additionalMarketDisruptionEvent.'

&lt;/...&gt;

**Diagram**

**Schema Component Representation**

```

<xsd:complexType name="CommodityMarketDisruption">
  <xsd:sequence>
    <xsd:choice>
      <xsd:sequence>
        <xsd:element name="marketDisruptionEvents" type=" MarketDisruptionEventsEnum " />
        <xsd:element name="additionalMarketDisruptionEvent" type=" MarketDisruptionEvent "
          " minOccurs="0" maxOccurs="unbounded"/>
      </xsd:sequence>
      <xsd:element name="marketDisruptionEvent" type=" MarketDisruptionEvent " maxOccurs="unbounded"/>
    </xsd:choice>
    <xsd:choice minOccurs="0">
      <xsd:element name="disruptionFallbacks" type=" DisruptionFallbacksEnum " />
      <xsd:element name="disruptionFallback" type=" SequencedDisruptionFallback "
        " maxOccurs="unbounded"/>
    </xsd:choice>
    <xsd:element name="fallbackReferencePrice" type=" Underlyer " minOccurs="0"/>
    <xsd:element name="maximumNumberOfDaysOfDisruption" type=" xsd:nonNegativeInteger "
      " minOccurs="0"/>
    <xsd:element name="priceMaterialityPercentage" type=" xsd:decimal " minOccurs="0"/>
    <xsd:element name="minimumFuturesContracts" type=" xsd:positiveInteger " minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>

```

top

**Complex Type: CommodityMultipleExercise**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None

<b>Name</b>	CommodityMultipleExercise
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">CommodityAmericanExercise</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type for defining the multiple exercise provisions of an American style commodity option.

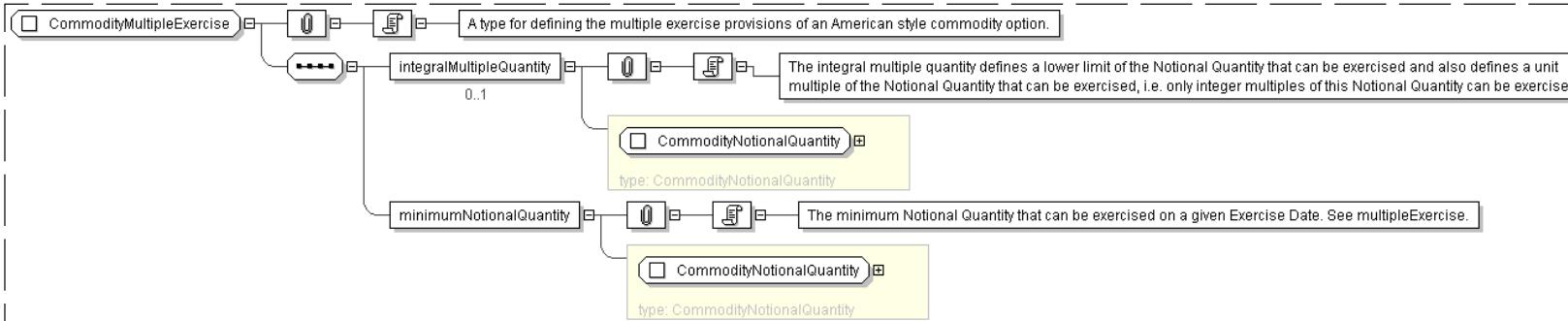
**XML Instance Representation**

```

<...>
<integralMultipleQuantity> CommodityNotionalQuantity </integralMultipleQuantity> [0..1]
'The integral multiple quantity defines a lower limit of the Notional Quantity that can
be exercised and also defines a unit multiple of the Notional Quantity that can be exercised,
i.e. only integer multiples of this Notional Quantity can be exercised.'

<minimumNotionalQuantity> CommodityNotionalQuantity </minimumNotionalQuantity> [1]
'The minimum Notional Quantity that can be exercised on a given Exercise Date.
See multipleExercise.'
</...>

```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="CommodityMultipleExercise">
  <xsd:sequence>
    <xsd:element name="integralMultipleQuantity" type=" CommodityNotionalQuantity " minOccurs="0"/>
    <xsd:element name="minimumNotionalQuantity" type=" CommodityNotionalQuantity " />
  </xsd:sequence>
</xsd:complexType>

```

top

**Complex Type: CommodityNotionalQuantity**

**Super-types:** None  
**Sub-types:**

- [CommoditySettlementPeriodsNotionalQuantity](#) (by extension)
- [ElectricityPhysicalDeliveryQuantity](#) (by extension)

<b>Name</b>	CommodityNotionalQuantity
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">CommodityMultipleExercise</a> , Complex Type <a href="#">CommodityMultipleExercise</a> , Complex Type <a href="#">CommodityNotionalQuantitySchedule</a> , Complex Type <a href="#">CommodityPhysicalQuantitySchedule</a> , Complex Type <a href="#">CommoditySettlementPeriodsNotionalQuantitySchedule</a> , Complex Type <a href="#">GasPhysicalQuantity</a> , Complex Type <a href="#">GasPhysicalQuantity</a> , Model Group <a href="#">CommodityFixedPhysicalQuantity.model</a> , Model Group <a href="#">CommodityNotionalQuantity.model</a>
<b>Abstract</b>	no
<b>Documentation</b>	Commodity Notional.

**XML Instance Representation**

```

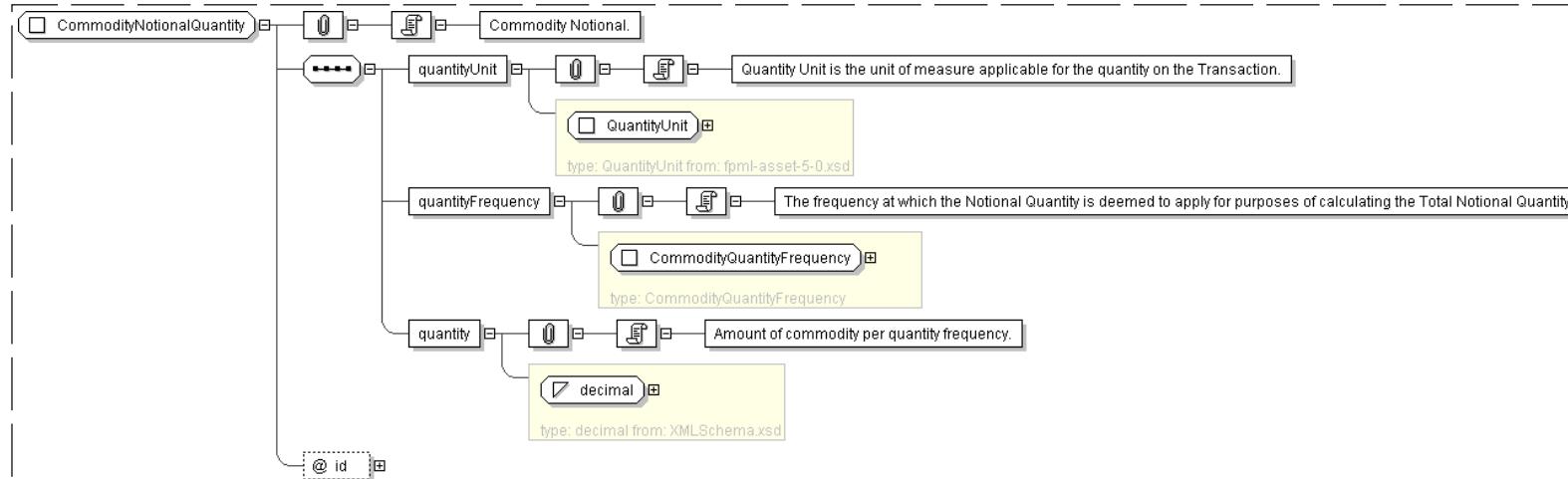
<...>
  id=" xsd:ID [0..1]">
  <quantityUnit> QuantityUnit </quantityUnit> [1]
  'Quantity Unit is the unit of measure applicable for the quantity on the Transaction.'

  <quantityFrequency> CommodityQuantityFrequency </quantityFrequency> [1]
  'The frequency at which the Notional Quantity is deemed to apply for purposes of calculating the Total Notional Quantity.'

  <quantity> xsd:decimal </quantity> [1]
  'Amount of commodity per quantity frequency.'

</...>

```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="CommodityNotionalQuantity">
  <xsd:sequence>
    <xsd:element name="quantityUnit" type=" QuantityUnit "/>
    <xsd:element name="quantityFrequency" type=" CommodityQuantityFrequency "/>

```

```

<xsd:element name="quantity" type="xsd:decimal" />
</xsd:sequence>
<xsd:attribute name="id" type="xsd:ID" />
</xsd:complexType>

```

## Complex Type: CommodityNotionalQuantitySchedule

**Super-types:** None  
**Sub-types:** None

<b>Name</b>	CommodityNotionalQuantitySchedule
<b>Used by (from the same schema document)</b>	Model Group <a href="#">CommodityNotionalQuantity.model</a>
<b>Abstract</b>	no
<b>Documentation</b>	The Notional Quantity per Calculation Period. There must be a Notional Quantity step specified for each Calculation Period, regardless of whether the Notional Quantity changes or remains the same between periods.

### XML Instance Representation

```

<...
  id="xsd:ID [0..1]">
  Start Choice [1]
    <notionalStep> CommodityNotionalQuantity </notionalStep> [1..*]
      'The Notional Quantity per Calculation Period. There must be a Notional Quantity specified
      for each Calculation Period, regardless of whether the quantity changes or remains the
      same between periods.'

    <settlementPeriodsNotionalQuantitySchedule>
      CommoditySettlementPeriodsNotionalQuantitySchedule </
      settlementPeriodsNotionalQuantitySchedule> [1..*]
        'For an electricity transaction, the Notional Quantity schedule for a one or more groups
        of Settlement Periods to which the Notional Quantity is based. If the schedule differs
        for different groups of Settlement Periods, this element should be repeated.'

    End Choice
    Start Choice [1]
      <calculationPeriodsReference> CalculationPeriodsReference </calculationPeriodsReference> [1]
        'A pointer style reference to the Calculation Periods defined on another leg.'

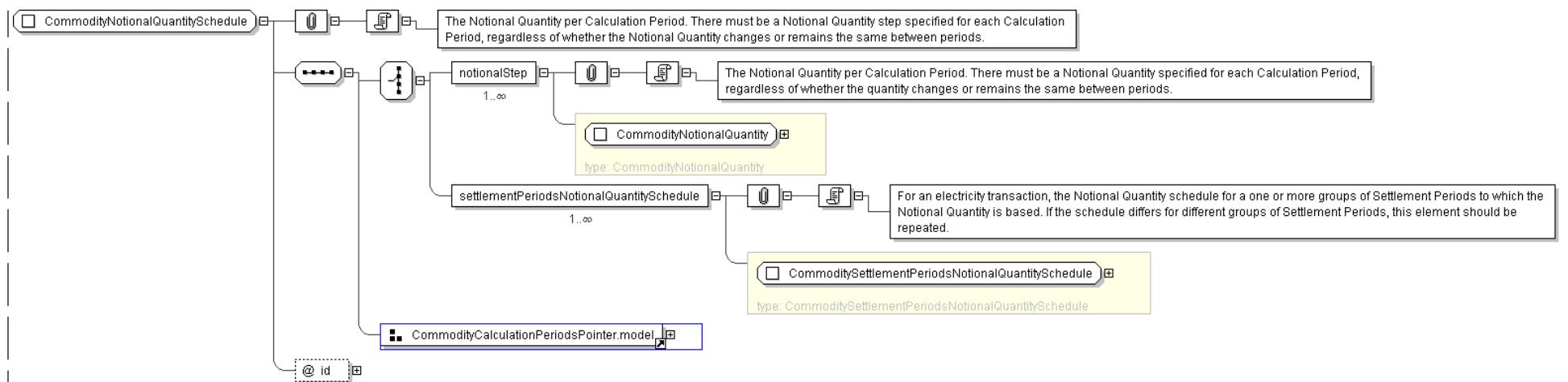
      <calculationPeriodsScheduleReference> CalculationPeriodsScheduleReference
      </calculationPeriodsScheduleReference> [1]
        'A pointer style reference to the Calculation Periods Schedule defined on another leg.'

      <calculationPeriodsDatesReference> CalculationPeriodsDatesReference
      </calculationPeriodsDatesReference> [1]
        'A pointer style reference to single-day-duration Calculation Periods defined on another leg.'

    End Choice
  </...>

```

### Diagram

**Schema Component Representation**

```

<xsd:complexType name="CommodityNotionalQuantitySchedule">
  <xsd:sequence>
    <xsd:choice>
      <xsd:element name="notionalStep" type="CommodityNotionalQuantity" maxOccurs="unbounded"/>
      <xsd:element name="settlementPeriodsNotionalQuantitySchedule"
        type="CommoditySettlementPeriodsNotionalQuantitySchedule" maxOccurs="unbounded"/>
    </xsd:choice>
    <xsd:group ref="CommodityCalculationPeriodsPointer.model"/>
  </xsd:sequence>
  <xsd:attribute name="id" type="xsd:ID"/>
</xsd:complexType>

```

top

**Complex Type: CommodityOption**

<b>Super-types:</b>	<code>Product</code> < <code>CommodityOption</code> (by extension)
<b>Sub-types:</b>	None
<b>Name</b>	CommodityOption
<b>Used by (from the same schema document)</b>	Element <code>commodityOption</code>
<b>Abstract</b>	no
<b>Documentation</b>	Commodity Option.

**XML Instance Representation**

```

<...
  id="xsd:ID [0..1]">
  <productType> ProductType </productType> [0..*]
  'A classification of the type of product. FpML defines a simple product categorization using
  a coding scheme.'
  <productId> ProductId </productId> [0..*]
  'A product 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.'
  <buyerPartyReference> PartyReference </buyerPartyReference> [1]
  'A reference to the party that buys this instrument, ie. pays for this instrument and
  receives the rights defined by it. See 2000 ISDA definitions Article 11.1 (b). In the case
  of FRAs this the fixed rate payer.'
  <buyerAccountReference> AccountReference </buyerAccountReference> [0..1]
  'A reference to the account that buys this instrument.'
  <sellerPartyReference> PartyReference </sellerPartyReference> [1]

```

'A reference to the party that sells ("writes") this instrument, i.e. that grants the rights defined by this instrument and in return receives a payment for it. See 2000 ISDA definitions Article 11.1 (a). In the case of FRAs this is the floating rate payer.'

<sellerAccountReference> [AccountReference](#) </sellerAccountReference> [0..1]

'A reference to the account that sells this instrument.'

<optionType> [PutCallEnum](#) </optionType> [1]

'The type of option transaction.'

Start [Choice](#) [1]

<commodity> [Commodity](#) </commodity> [1]

'Specifies the underlying component. At the time of the initial schema design, only underlyers of type Commodity are supported; the choice group in the future could offer the possibility of adding other types later.'

Start Group: [CommodityAsian.model](#) [0..1]

'A group containing properties specific to Asian options.'

<effectiveDate> [AdjustableOrRelativeDate](#) </effectiveDate> [1]

'The effective date of the Commodity Option Transaction. Note that the Termination/Expiration Date should be specified in expirationDate within the CommodityAmericanExercise type or the CommodityEuropeanExercise type, as applicable.'

Start [Choice](#) [1]

<calculationPeriodsSchedule> [CommodityCalculationPeriodsSchedule](#) </calculationPeriodsSchedule> [1]

'A parametric representation of the Calculation Periods of the Commodity Option Transaction.'

<calculationPeriods> [AdjustableDates](#) </calculationPeriods> [1]

'An absolute representation of the Calculation Period start dates of the Commodity Option Transaction.'

End Choice

<pricingDates> [CommodityPricingDates](#) </pricingDates> [1]

'The dates on which the option will price.'

<averagingMethod> [AveragingMethodEnum](#) </averagingMethod> [0..1]

'The Method of Averaging if there is more than one Pricing Date.'

End Group: [CommodityAsian.model](#)

Start [Choice](#) [1]

Start [Choice](#) [1]

<notionalQuantitySchedule> [CommodityNotionalQuantitySchedule](#) </notionalQuantitySchedule> [1]

'Allows the documentation of a shaped notional trade where the notional changes over the life of the transaction.'

<notionalQuantity> [CommodityNotionalQuantity](#) </notionalQuantity> [1]

'The Notional Quantity.'

<settlementPeriodsNotionalQuantity> [CommoditySettlementPeriodsNotionalQuantity](#) </settlementPeriodsNotionalQuantity> [1..\*]

'For an electricity transaction, the Notional Quantity for a one or more groups of Settlement Periods to which the Notional Quantity is based. If the schedule differs for different groups of Settlement Periods, this element should be repeated.'

End Choice

<totalNotionalQuantity> xsd:decimal </totalNotionalQuantity> [0..1]

'The Total Notional Quantity.'

<quantityReference> [QuantityReference](#) </quantityReference> [1]

'A pointer style reference to a quantity defined on another leg.'

End Choice

<exercise> [CommodityExercise](#) </exercise> [1]

'The parameters for defining how the commodity option can be exercised and how it is settled.'

Start [Choice](#) [1]

<strikePricePerUnit> [NonNegativeMoney](#) </strikePricePerUnit> [1]

'The currency amount of the strike price per unit.'

```

<strikePricePerUnitSchedule> CommodityStrikeSchedule </strikePricePerUnitSchedule> [1]
End Choice
Start Choice [1]
    <commoditySwap> ... </commoditySwap> [1]
    <commodityForward> ... </commodityForward> [1]
End Choice
<physicalExercise> CommodityPhysicalExercise </physicalExercise> [1]
'The parameters for defining how the commodity option can be exercised into a
physical transaction.'

End Choice
<premium> CommodityPremium </premium> [1]
'The option premium payable by the buyer to the seller.'

Start Group: CommodityContent.model [0..1]
<commonPricing> xsd:boolean </commonPricing> [0..1]
'Common pricing may be relevant for a Transaction that references more than one
Commodity Reference Price. If Common Pricing is not specified as applicable, it will be
deemed not to apply.'

<marketDisruption> CommodityMarketDisruption </marketDisruption> [0..1]
'Market disruption events as defined in the ISDA 1993 Commodity Definitions or in ISDA
2005 Commodity Definitions, as applicable.'

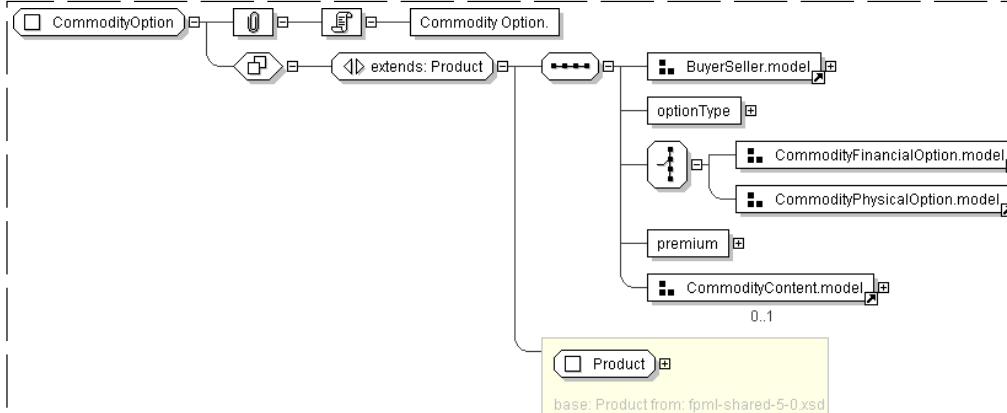
<settlementDisruption> CommodityBullionSettlementDisruptionEnum </settlementDisruption> [0..1]
'The consequences of Bullion Settlement Disruption Events.'

<rounding> Rounding </rounding> [0..1]
'Rounding direction and precision for amounts.'

End Group: CommodityContent.model
</...>

```

## Diagram



## Schema Component Representation

```

<xsd:complexType name="CommodityOption">
    <xsd:complexContent>
        <xsd:extension base=" Product ">
            <xsd:sequence>
                <xsd:group ref=" BuyerSeller.model " />
                <xsd:element name="optionType" type=" PutCallEnum " />
                <xsd:choice>
                    <xsd:group ref=" CommodityFinancialOption.model " />
                    <xsd:group ref=" CommodityPhysicalOption.model " />
                </xsd:choice>
                <xsd:element name="premium" type=" CommodityPremium " />
                <xsd:group ref=" CommodityContent.model " minOccurs="0" />
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

```

## Complex Type: CommodityPayRelativeToEvent

Super-types:	<a href="#">Scheme</a> < <b>CommodityPayRelativeToEvent</b> (by extension)
Sub-types:	None
Name	<b>CommodityPayRelativeToEvent</b>
Used by (from the same schema document)	Complex Type <a href="#">CommodityRelativePaymentDates</a>
Abstract	no
Documentation	A scheme identifying the physical event relative to which payment occurs.
<b>XML Instance Representation</b> <pre>&lt;...&gt;   commodityPayRelativeToEventScheme=" xsd:anyURI [0..1]"&gt;   Scheme &lt;/...&gt;</pre>	
<b>Diagram</b> <pre> classDiagram     class CommodityPayRelativeToEvent {         &lt;&lt;A scheme identifying the physical event relative to which payment occurs.&gt;&gt;     }     class Scheme {         &lt;&lt;@ commodityPayRelativeToEventScheme &gt;&gt;     }     CommodityPayRelativeToEvent &lt; -- Scheme     &lt;&lt;base: Scheme from: fpml-shared-5-0.xsd&gt;&gt;   </pre>	
<b>Schema Component Representation</b> <pre> &lt;xsd:complexType name="CommodityPayRelativeToEvent"&gt;   &lt;xsd:simpleContent&gt;     &lt;xsd:extension base=" Scheme "&gt;       &lt;xsd:attribute name="commodityPayRelativeToEventScheme" type=" xsd:anyURI " default="http://www.fpml.org/coding-scheme/commodity-pay-relative-to-event"/&gt;     &lt;/xsd:extension&gt;   &lt;/xsd:simpleContent&gt; &lt;/xsd:complexType&gt;   </pre>	

## Complex Type: CommodityPhysicalAmericanExercise

Super-types:	<a href="#">Exercise</a> < <b>CommodityPhysicalAmericanExercise</b> (by extension)
Sub-types:	None
Name	<b>CommodityPhysicalAmericanExercise</b>
Used by (from the same schema document)	Complex Type <a href="#">CommodityPhysicalExercise</a>
Abstract	no
Documentation	The parameters for defining the expiration date(s) and time(s) for an American style option.
<b>XML Instance Representation</b> <pre>&lt;...&gt;   id=" xsd:ID [0..1]"&gt;   Start Choice [1]     &lt;commencementDates&gt; AdjustableOrRelativeDates &lt;/commencementDates&gt; [1]     'The first day(s) of the exercise period(s) for an American-style option.'     &lt;expirationDates&gt; AdjustableOrRelativeDates &lt;/expirationDates&gt; [1]     'The Expiration Date(s) of an American-style option.'     &lt;relativeCommencementDates&gt; CommodityRelativeExpirationDates &lt;/relativeCommencementDates&gt; [1]     'The first day(s) of the exercise period(s) for an American-style option where it is     relative to the occurrence of an external event.'     &lt;relativeExpirationDates&gt; CommodityRelativeExpirationDates &lt;/relativeExpirationDates&gt; [1]</pre>	

'The Expiration Date(s) of an American-style option where it is relative to the occurrence of an external event.'

End Choice

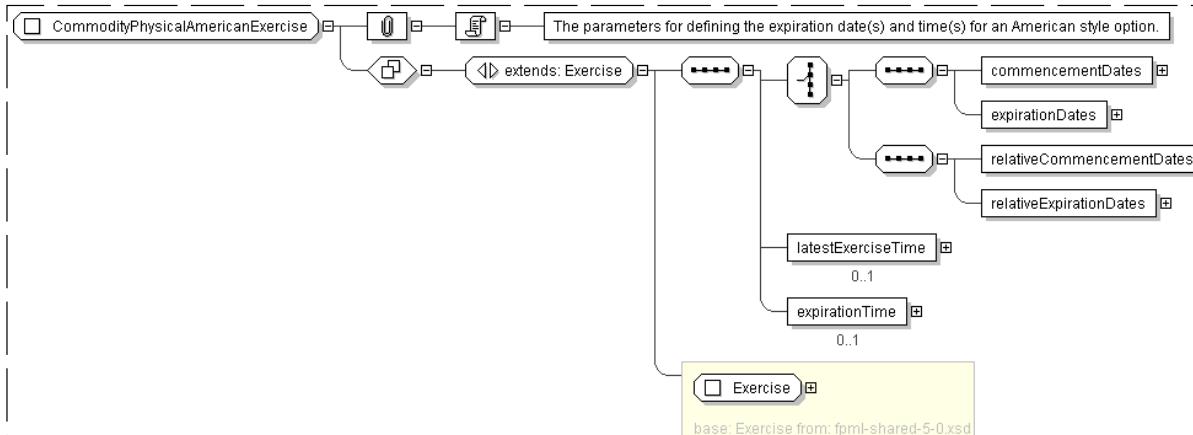
<latestExerciseTime> PrevailingTime </latestExerciseTime> [0..1]

'For a Bermuda or American style option, the latest time on an exercise business day (excluding the expiration date) within the exercise period that notice can be given by the 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> PrevailingTime </expirationTime> [0..1]

'The specific time of day at which the option expires.'

&lt;/...&gt;

**Diagram****Schema Component Representation**

```

<xsd:complexType name="CommodityPhysicalAmericanExercise">
  <xsd:complexContent>
    <xsd:extension base="#Exercise">
      <xsd:sequence>
        <xsd:choice>
          <xsd:sequence>
            <xsd:element name="commencementDates" type="AdjustableOrRelativeDates" />
            <xsd:element name="expirationDates" type="AdjustableOrRelativeDates" />
          </xsd:sequence>
          <xsd:sequence>
            <xsd:element name="relativeCommencementDates" type="CommodityRelativeExpirationDates" />
            <xsd:element name="relativeExpirationDates" type="CommodityRelativeExpirationDates" />
          </xsd:sequence>
        </xsd:choice>
        <xsd:element name="latestExerciseTime" type="#PrevailingTime" minOccurs="0" />
        <xsd:element name="expirationTime" type="#PrevailingTime" minOccurs="0" />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
  
```

top

**Complex Type: CommodityPhysicalEuropeanExercise**

Super-types:

Exercise &lt; CommodityPhysicalEuropeanExercise (by extension)

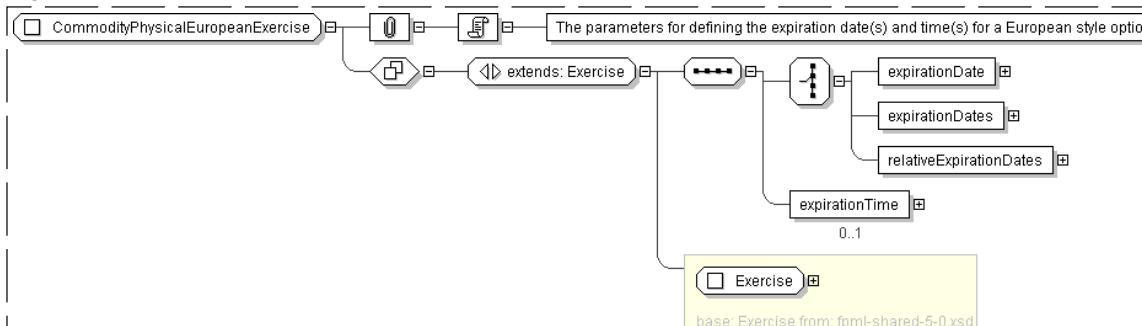
Sub-types:

None

Name	CommodityPhysicalEuropeanExercise
Used by (from the same schema document)	Complex Type <a href="#">CommodityPhysicalExercise</a>
Abstract	no
Documentation	The parameters for defining the expiration date(s) and time(s) for a European style option.

**XML Instance Representation**

```
<...>
  id="xsd:ID [0..1]">
Start Choice [1]
  <expirationDate> AdjustableOrRelativeDate </expirationDate> [1]
  'The Expiration Date of a single expiry European-style option or the first Expiration Date of
  a multiple expiry or daily expiring option.'
  <expirationDates> AdjustableRelativeOrPeriodicDates2 </expirationDates> [1]
  'The Expiration Date(s) of a European-style option.'
  <relativeExpirationDates> CommodityRelativeExpirationDates </relativeExpirationDates> [1]
  'The Expiration Date(s) of a European-style option where it is relative to the occurrence of
  an external event.'
End Choice
  <expirationTime> PrevailingTime </expirationTime> [0..1]
  'The specific time of day at which the option expires.'
</...>
```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="CommodityPhysicalEuropeanExercise">
  <xsd:complexContent>
    <xsd:extension base=" Exercise ">
      <xsd:sequence>
        <xsd:choice>
          <xsd;element name="expirationDate" type=" AdjustableOrRelativeDate "/>
          <xsd;element name="expirationDates" type=" AdjustableRelativeOrPeriodicDates2 "/>
          <xsd;element name="relativeExpirationDates" type=" CommodityRelativeExpirationDates "/>
        </xsd:choice>
        <xsd;element name="expirationTime" type=" PrevailingTime " minOccurs="0"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

top

**Complex Type: CommodityPhysicalExercise**

Super-types:	None
Sub-types:	None

Name	CommodityPhysicalExercise
Used by (from the same schema document)	Model Group <code>CommodityPhysicalOption.model</code>
Abstract	no
Documentation	The parameters for defining how the physically-settled commodity option can be exercised and how it is settled.

**XML Instance Representation**

```
<...>
  |
```

```

Start Choice [1]
  <americanExercise> CommodityPhysicalAmericanExercise </americanExercise> [1]
    'The parameters for defining the expiration date(s) and time(s) for an American style option.'

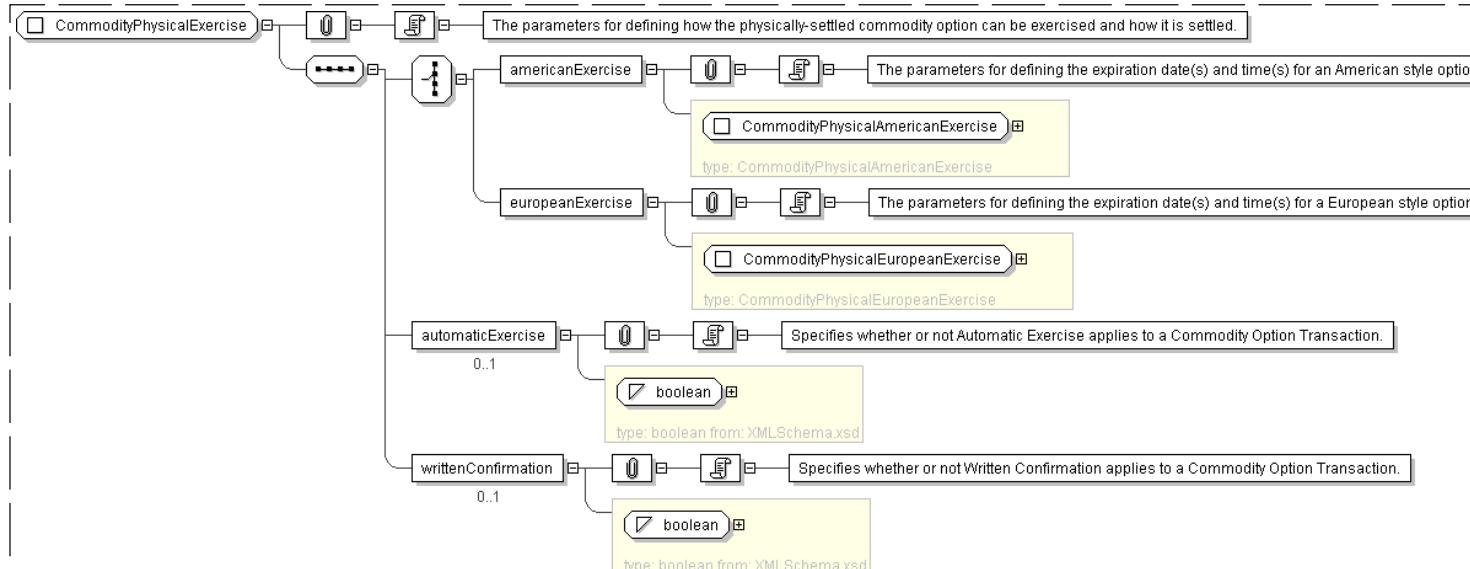
  <europeanExercise> CommodityPhysicalEuropeanExercise </europeanExercise> [1]
    'The parameters for defining the expiration date(s) and time(s) for a European style option.'

End Choice
<automaticExercise> xsd:boolean </automaticExercise> [0..1]
  'Specifies whether or not Automatic Exercise applies to a Commodity Option Transaction.'

<writtenConfirmation> xsd:boolean </writtenConfirmation> [0..1]
  'Specifies whether or not Written Confirmation applies to a Commodity Option Transaction.'

</...>

```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="CommodityPhysicalExercise">
  <xsd:sequence>
    <xsd:choice>
      <xsd:element name="americanExercise" type="CommodityPhysicalAmericanExercise" />
      <xsd:element name="europeanExercise" type="CommodityPhysicalEuropeanExercise" />
    </xsd:choice>
    <xsd:element name="automaticExercise" type="xsd:boolean" minOccurs="0"/>
    <xsd:element name="writtenConfirmation" type="xsd:boolean" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>

```

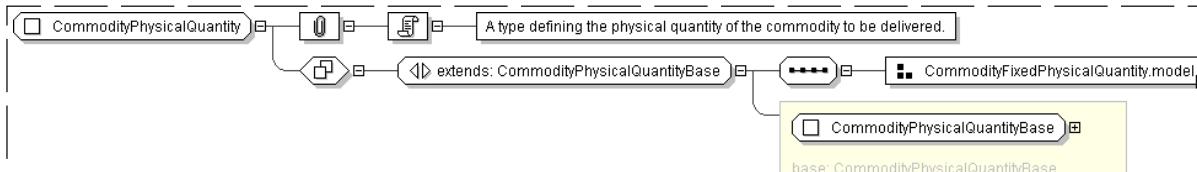
top

**Complex Type: CommodityPhysicalQuantity**

<b>Super-types:</b>	<a href="#">CommodityPhysicalQuantityBase</a> < <b>CommodityPhysicalQuantity</b> (by extension)
<b>Sub-types:</b>	None
<b>Name</b>	<b>CommodityPhysicalQuantity</b>
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">CoalPhysicalLeg</a> , Complex Type <a href="#">OilPhysicalLeg</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type defining the physical quantity of the commodity to be delivered.

**XML Instance Representation**

```
<...  
id="xsd:ID [0..1]">  
Start Choice [1]  
  <physicalQuantity> CommodityNotionalQuantity </physicalQuantity> [1]  
    'The Quantity per Delivery Period.'  
  
  <physicalQuantitySchedule> CommodityPhysicalQuantitySchedule </physicalQuantitySchedule> [1]  
    'Allows the documentation of a shaped quantity trade where the quantity changes over the  
    life of the transaction.'  
  
End Choice  
  <totalPhysicalQuantity> UnitQuantity </totalPhysicalQuantity> [0..1]  
    'The Total Quantity of the commodity to be delivered.'  
  
</...>
```

**Diagram****Schema Component Representation**

```
<xsd:complexType name="CommodityPhysicalQuantity">  
  <xsd:complexContent>  
    <xsd:extension base=" CommodityPhysicalQuantityBase ">  
      <xsd:sequence>  
        <xsd:group ref=" CommodityFixedPhysicalQuantity.model "/>  
      </xsd:sequence>  
    </xsd:extension>  
  </xsd:complexContent>  
</xsd:complexType>
```

top

**Complex Type: CommodityPhysicalQuantityBase****Super-types:**

None

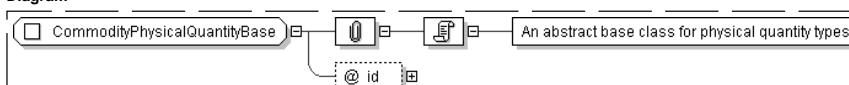
**Sub-types:**

- CommodityPhysicalQuantity (by extension)
- ElectricityPhysicalQuantity (by extension)
- GasPhysicalQuantity (by extension)

<b>Name</b>	CommodityPhysicalQuantityBase
<b>Abstract</b>	yes
<b>Documentation</b>	An abstract base class for physical quantity types.

**XML Instance Representation**

```
<...  
id="xsd:ID [0..1]">
```

**Diagram****Schema Component Representation**

```
<xsd:complexType name="CommodityPhysicalQuantityBase" abstract="true">  
  <xsd:attribute name="id" type="xsd:ID "/>  
</xsd:complexType>
```

top

## Complex Type: CommodityPhysicalQuantitySchedule

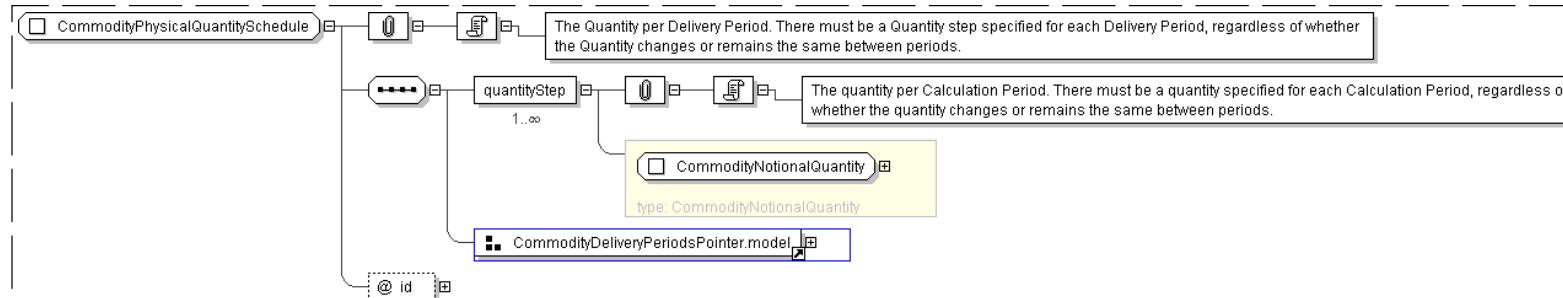
Super-types:	None
Sub-types:	<ul style="list-style-type: none"> <li>• <a href="#">ElectricityPhysicalDeliveryQuantitySchedule</a> (by extension)</li> </ul>

Name	CommodityPhysicalQuantitySchedule
Used by (from the same schema document)	Model Group <a href="#">CommodityFixedPhysicalQuantity.model</a>
Abstract	no
Documentation	The Quantity per Delivery Period. There must be a Quantity step specified for each Delivery Period, regardless of whether the Quantity changes or remains the same between periods.

### XML Instance Representation

```
<...>
<id="xsd:ID [0..1]">
  <quantityStep> CommodityNotionalQuantity </quantityStep> [1..*]
  'The quantity per Calculation Period. There must be a quantity specified for each Calculation Period, regardless of whether the quantity changes or remains the same between periods.'
</id>
Start Choice [1]
  <deliveryPeriodsReference> CalculationPeriodsReference </deliveryPeriodsReference> [1]
  'A pointer style reference to the Delivery Periods defined elsewhere.'
  <deliveryPeriodsScheduleReference> CalculationPeriodsScheduleReference
  </deliveryPeriodsScheduleReference> [1]
  'A pointer style reference to the Calculation Periods Schedule defined elsewhere.'
End Choice
</...>
```

### Diagram



### Schema Component Representation

```
<xsd:complexType name="CommodityPhysicalQuantitySchedule">
  <xsd:sequence>
    <xsd:element name="quantityStep" type=" CommodityNotionalQuantity " maxOccurs="unbounded" />
    <xsd:group ref=" CommodityDeliveryPeriodsPointer.model " />
  </xsd:sequence>
  <xsd:attribute name="id" type=" xsd:ID " />
</xsd:complexType>
```

top

## Complex Type: CommodityPipeline

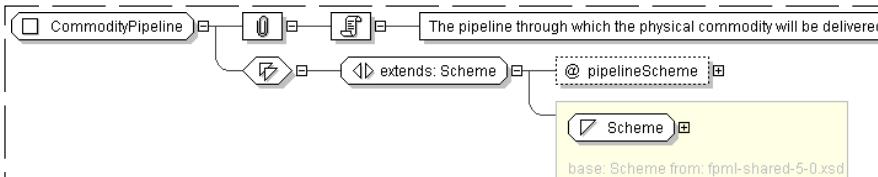
Super-types:	<a href="#">Scheme</a> < CommodityPipeline (by extension)
Sub-types:	None

Name	CommodityPipeline
Used by (from the same schema document)	Complex Type <a href="#">OilPipelineDelivery</a>
Abstract	no
Documentation	The pipeline through which the physical commodity will be delivered.

**XML Instance Representation**

```
<...>


```

**Diagram****Schema Component Representation**

```
<xsd:complexType name="CommodityPipeline">
  <xsd:simpleContent>
    <xsd:extension base=" Scheme ">
      <xsd:attribute name="pipelineScheme" type=" xsd:anyURI " />
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
```

top

**Complex Type: CommodityPipelineCycle**

**Super-types:** [Scheme](#) < CommodityPipelineCycle (by extension)

**Sub-types:** None

**Name** CommodityPipelineCycle

**Used by (from the same schema document)** Complex Type [OilPipelineDelivery](#)

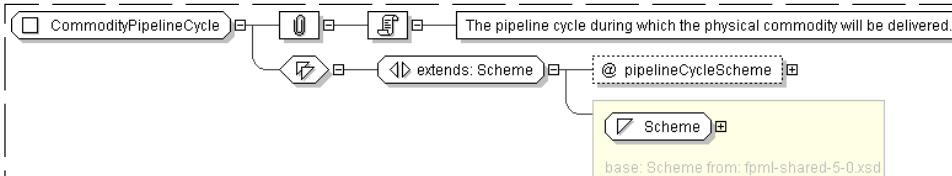
**Abstract** no

**Documentation** The pipeline cycle during which the physical commodity will be delivered.

**XML Instance Representation**

```
<...>


```

**Diagram****Schema Component Representation**

```
<xsd:complexType name="CommodityPipelineCycle">
  <xsd:simpleContent>
    <xsd:extension base=" Scheme ">
      <xsd:attribute name="pipelineCycleScheme" type=" xsd:anyURI " use="required"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
```

top

**Complex Type: CommodityPremium**

**Super-types:** [NonNegativePayment](#) < **CommodityPremium** (by extension)

**Sub-types:** None

**Name** CommodityPremium

**Used by (from the same schema document)** Complex Type [CommodityOption](#)

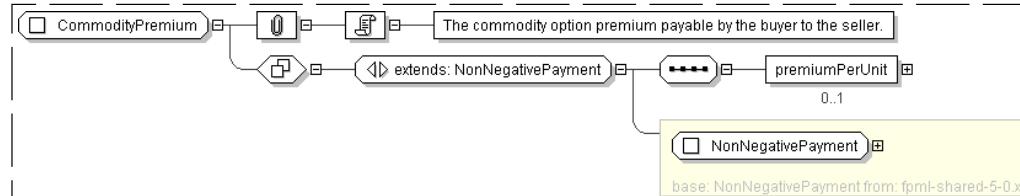
**Abstract** no

**Documentation** The commodity option premium payable by the buyer to the seller.

#### XML Instance Representation

```
<...>
  id="xsd:ID [0..1]">
    <payerPartyReference> PartyReference </payerPartyReference> [1]
      'A reference to the party responsible for making the payments defined by this structure.'
    <payerAccountReference> AccountReference </payerAccountReference> [0..1]
      'A reference to the account responsible for making the payments defined by this structure.'
    <receiverPartyReference> PartyReference </receiverPartyReference> [1]
      'A reference to the party that receives the payments corresponding to this structure.'
    <receiverAccountReference> AccountReference </receiverAccountReference> [0..1]
      'A reference to the account that receives the payments corresponding to this structure.'
    <paymentDate> AdjustableOrRelativeDate </paymentDate> [1]
      'The payment date, which can be expressed as either an adjustable or relative date.'
    <paymentAmount> NonNegativeMoney </paymentAmount> [1]
      'Non negative payment amount.'
    <premiumPerUnit> NonNegativeMoney </premiumPerUnit> [0..1]
      'The currency amount of premium to be paid per Unit of the Total Notional Quantity.'
  </...>
```

#### Diagram



#### Schema Component Representation

```
<xsd:complexType name="CommodityPremium">
  <xsd:complexContent>
    <xsd:extension base=" NonNegativePayment ">
      <xsd:sequence>
        <xsd:element name="premiumPerUnit" type=" NonNegativeMoney " minOccurs="0"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

top

#### Complex Type: CommodityPricingDates

**Super-types:** None

**Sub-types:** None

**Name** CommodityPricingDates

**Used by (from the same schema document)** Complex Type [FloatingLegCalculation](#), Model Group [CommodityAsian.model](#)

**Abstract** no

**Documentation** The dates on which prices are observed for the underlyer.

**XML Instance Representation**

```

<...
id=" xsd:ID [0..1]">
Start Choice [1]
 <calculationPeriodsReference> CalculationPeriodsReference </calculationPeriodsReference> [1]
  'A pointer style reference to the Calculation Periods defined on another leg.'

 <calculationPeriodsScheduleReference> CalculationPeriodsScheduleReference
 </calculationPeriodsScheduleReference> [1]
  'A pointer style reference to the Calculation Periods Schedule defined on another leg.'

 <calculationPeriodsDatesReference> CalculationPeriodsDatesReference
 </calculationPeriodsDatesReference> [1]
  'A pointer style reference to single-day-duration Calculation Periods defined on another leg.'

End Choice
Start Choice [1]
 <lag> Lag </lag> [0..1]
  'The pricing period per calculation period if the pricing days do not wholly fall within
the respective calculation period.'

Start Choice [1]
 <dayType> CommodityDayTypeEnum </dayType> [1]
  'The type of day on which pricing occurs.'

Start Choice [1]
 <dayDistribution> CommodityFrequencyType </dayDistribution> [1]
  'The method by which the pricing days are distributed across the pricing period.'

 <dayCount> xsd:positiveInteger </dayCount> [0..1]
  'The number of days over which pricing should take place.'

 <dayOfWeek> DayOfWeekEnum </dayOfWeek> [1..7]
  'The day(s) of the week on which pricing will take place during the pricing period.'

 <dayNumber> xsd:integer </dayNumber> [0..1]
  'The occurrence of the dayOfWeek within the pricing period on which pricing will take place,
e.g. the 3rd Friday within each Calculation Period. If omitted, every dayOfWeek will be
a pricing day.'

End Choice
 <businessCalendar> CommodityBusinessCalendar </businessCalendar> [0..1]
  'Identifies a commodity business day calendar from which the pricing dates will be generated.'

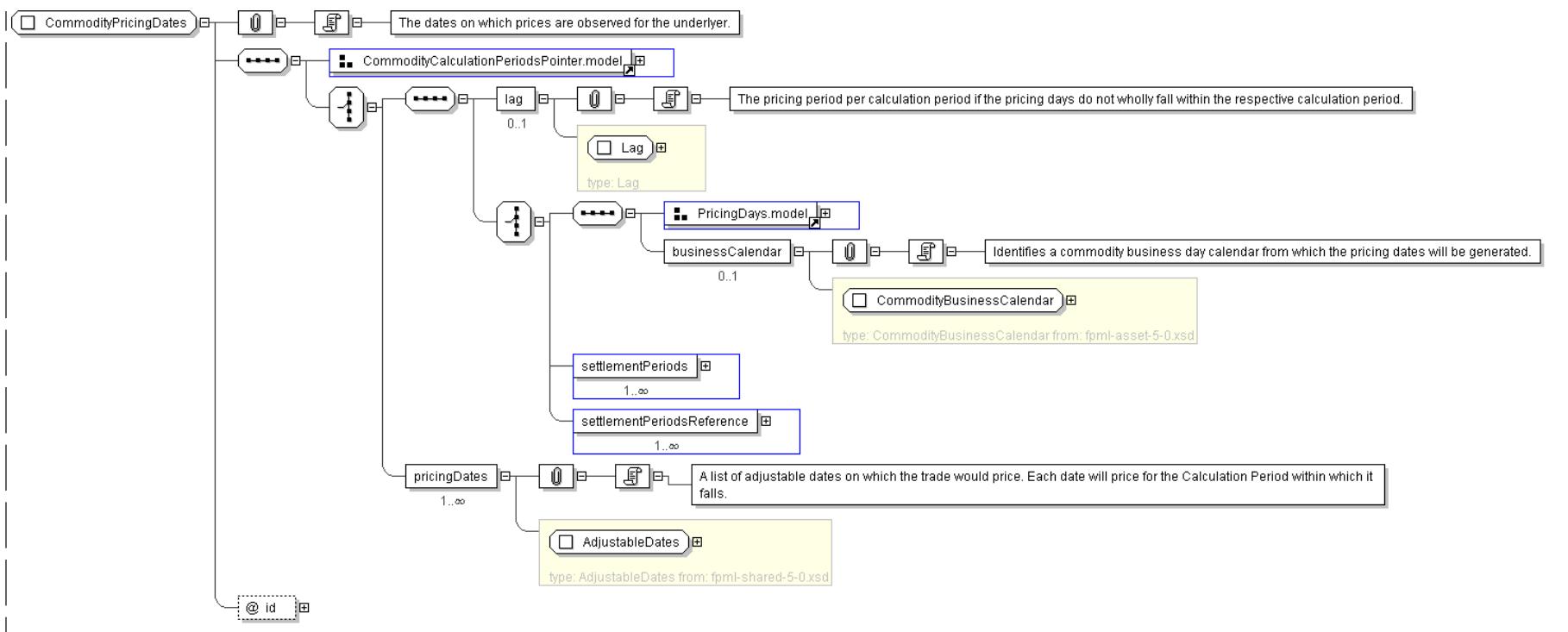
 <settlementPeriods> SettlementPeriods </settlementPeriods> [1..*]
 <settlementPeriodsReference> SettlementPeriodsReference </settlementPeriodsReference> [1..*]

End Choice
 <pricingDates> AdjustableDates </pricingDates> [1..*]
  'A list of adjustable dates on which the trade would price. Each date will price for
the Calculation Period within which it falls.'

End Choice
</...>

```

**Diagram**

**Schema Component Representation**

```

<xsd:complexType name="CommodityPricingDates">
  <xsd:sequence>
    <xsd:group ref=" CommodityCalculationPeriodsPointer.model " />
    <xsd:choice>
      <xsd:sequence>
        <xsd:element name="lag" type=" Lag " minOccurs="0"/>
        <xsd:choice>
          <xsd:sequence>
            <xsd:group ref=" PricingDays.model " />
            <xsd:element name="businessCalendar" type=" CommodityBusinessCalendar " minOccurs="0" />
          </xsd:sequence>
          <xsd:element name="settlementPeriods" type=" SettlementPeriods " maxOccurs="unbounded" />
          <xsd:element name="settlementPeriodsReference" type=" SettlementPeriodsReference "
            " maxOccurs="unbounded" />
        </xsd:choice>
      </xsd:sequence>
      <xsd:element name="pricingDates" type=" AdjustableDates " maxOccurs="unbounded" />
    </xsd:choice>
  </xsd:sequence>
  <xsd:attribute name="id" type=" xsd:ID " />
</xsd:complexType>

```

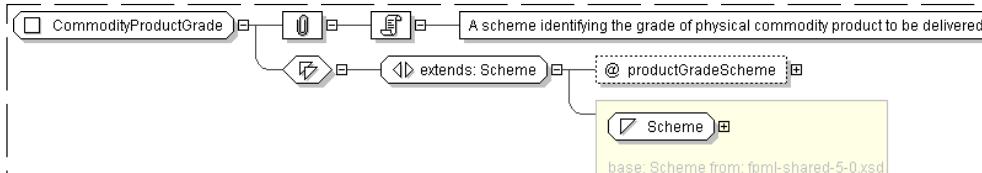
top

**Complex Type: CommodityProductGrade**

<b>Super-types:</b>	<a href="#">Scheme</a> < <b>CommodityProductGrade</b> (by extension)
<b>Sub-types:</b>	None
<b>Name</b>	CommodityProductGrade
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">OilProduct</a>
<b>Abstract</b>	no
<b>Documentation</b>	A scheme identifying the grade of physical commodity product to be delivered.

**XML Instance Representation**

```
<...
  productGradeScheme=" xsd:anyURI [ 0..1 ]">
  Scheme
</...>
```

**Diagram****Schema Component Representation**

```
<xsd:complexType name="CommodityProductGrade">
  <xsd:simpleContent>
    <xsd:extension base=" Scheme ">
      <xsd:attribute name="productGradeScheme" type=" xsd:anyURI " />
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
```

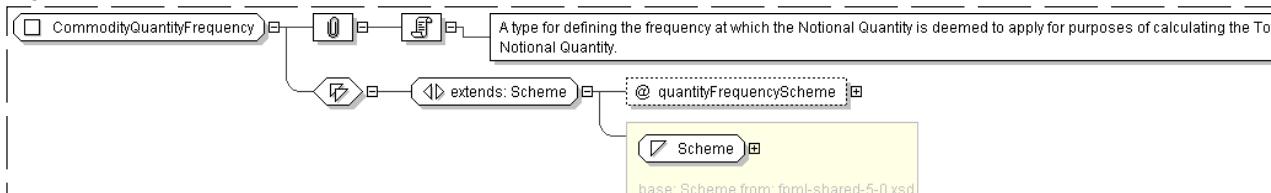
[top](#)**Complex Type: CommodityQuantityFrequency**

<b>Super-types:</b>	<a href="#">Scheme</a> < <b>CommodityQuantityFrequency</b> (by extension)
<b>Sub-types:</b>	None

<b>Name</b>	CommodityQuantityFrequency
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">CommodityNotionalQuantity</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type for defining the frequency at which the Notional Quantity is deemed to apply for purposes of calculating the Total Notional Quantity.

**XML Instance Representation**

```
<...
  quantityFrequencyScheme=" xsd:anyURI [ 0..1 ]">
  Scheme
</...>
```

**Diagram****Schema Component Representation**

```
<xsd:complexType name="CommodityQuantityFrequency">
  <xsd:simpleContent>
    <xsd:extension base=" Scheme ">
      <xsd:attribute name="quantityFrequencyScheme" type=" xsd:anyURI " default="http://www.fpml.org/coding-scheme/commodity-quantity-frequency"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
```

[top](#)**Complex Type: CommodityRelativeExpirationDates**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None
<b>Name</b>	CommodityRelativeExpirationDates
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">CommodityPhysicalAmericanExercise</a> , Complex Type <a href="#">CommodityPhysicalExercise</a> , Complex Type <a href="#">CommodityPhysicalEuropeanExercise</a>
<b>Abstract</b>	no
<b>Documentation</b>	The Expiration Dates of the trade relative to the Calculation Periods.

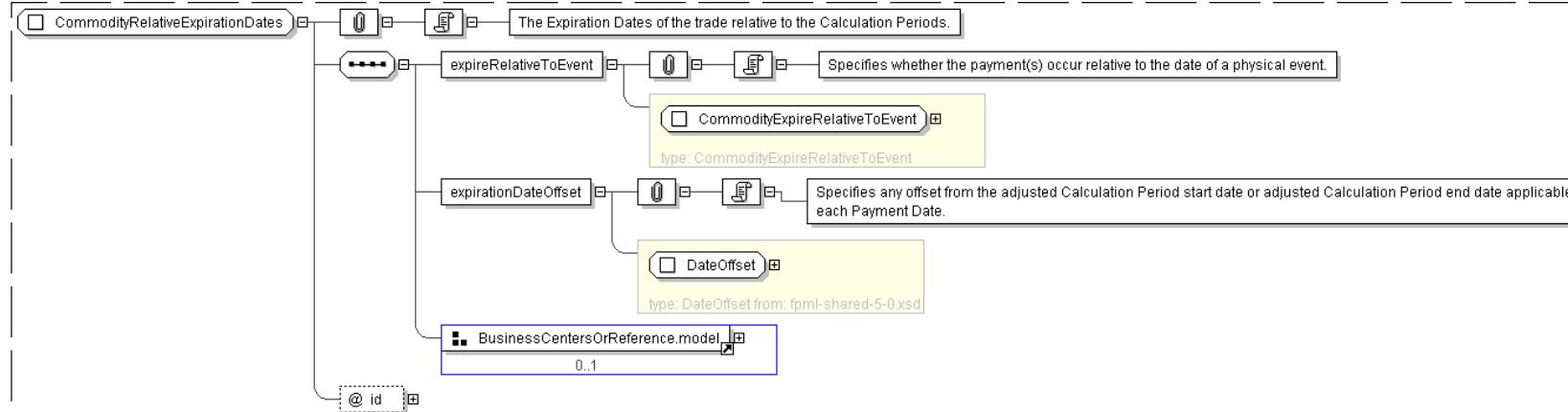
**XML Instance Representation**

```
<...>
  id="#xsd:ID [0..1]">
    <expireRelativeToEvent> CommodityExpireRelativeToEvent </expireRelativeToEvent> [1]
    'Specifies whether the payment(s) occur relative to the date of a physical event.'

    <expirationDateOffset> DateOffset </expirationDateOffset> [1]
    'Specifies any offset from the adjusted Calculation Period start date or adjusted Calculation Period end date applicable to each Payment Date.'

Start Group: BusinessCentersOrReference.model [0..1]
Start Choice [1]
  <businessCentersReference> BusinessCentersReference </businessCentersReference> [1]
    '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.'

    <businessCenters> BusinessCenters </businessCenters> [1]
End Choice
End Group: BusinessCentersOrReference.model
</...>
```

**Diagram****Schema Component Representation**

```
<xsd:complexType name="CommodityRelativeExpirationDates">
  <xsd:sequence>
    <xsd:element name="expireRelativeToEvent" type="CommodityExpireRelativeToEvent" />
    <xsd:element name="expirationDateOffset" type="DateOffset" />
    <xsd:group ref="BusinessCentersOrReference.model" * minOccurs="0" />
  </xsd:sequence>
  <xsd:attribute name="id" type="xsd:ID" />
</xsd:complexType>
```

**Complex Type: CommodityRelativePaymentDates**

<b>Super-types:</b>	None
---------------------	------

<b>Sub-types:</b>	None
<b>Name</b>	CommodityRelativePaymentDates
<b>Used by (from the same schema document)</b>	Model Group <a href="#">CommodityPaymentDates.model</a>
<b>Abstract</b>	no

**Documentation**  
The Payment Dates of the trade relative to the Calculation Periods.

**XML Instance Representation**

```
<...>
<id=" xsd:ID [0..1]">
Start Choice [1]
  <payRelativeTo> PayRelativeToEnum </payRelativeTo> [1]
    'Specifies whether the payment(s) occur relative to a date such as the end of each
    Calculation Period or the last Pricing Date in each Calculation Period.'
  <payRelativeToEvent> CommodityPayRelativeToEvent </payRelativeToEvent> [1]
    'Specifies whether the payment(s) occur relative to the date of a physical event.'

End Choice
Start Choice [1]
  <calculationPeriodsReference> CalculationPeriodsReference </calculationPeriodsReference> [1]
    'A pointer style reference to the Calculation Periods defined on another leg.'

  <calculationPeriodsScheduleReference> CalculationPeriodsScheduleReference
  </calculationPeriodsScheduleReference> [1]
    'A pointer style reference to the Calculation Periods Schedule defined on another leg.'

  <calculationPeriodsDatesReference> CalculationPeriodsDatesReference
  </calculationPeriodsDatesReference> [1]
    'A pointer style reference to single-day-duration Calculation Periods defined on another leg.'

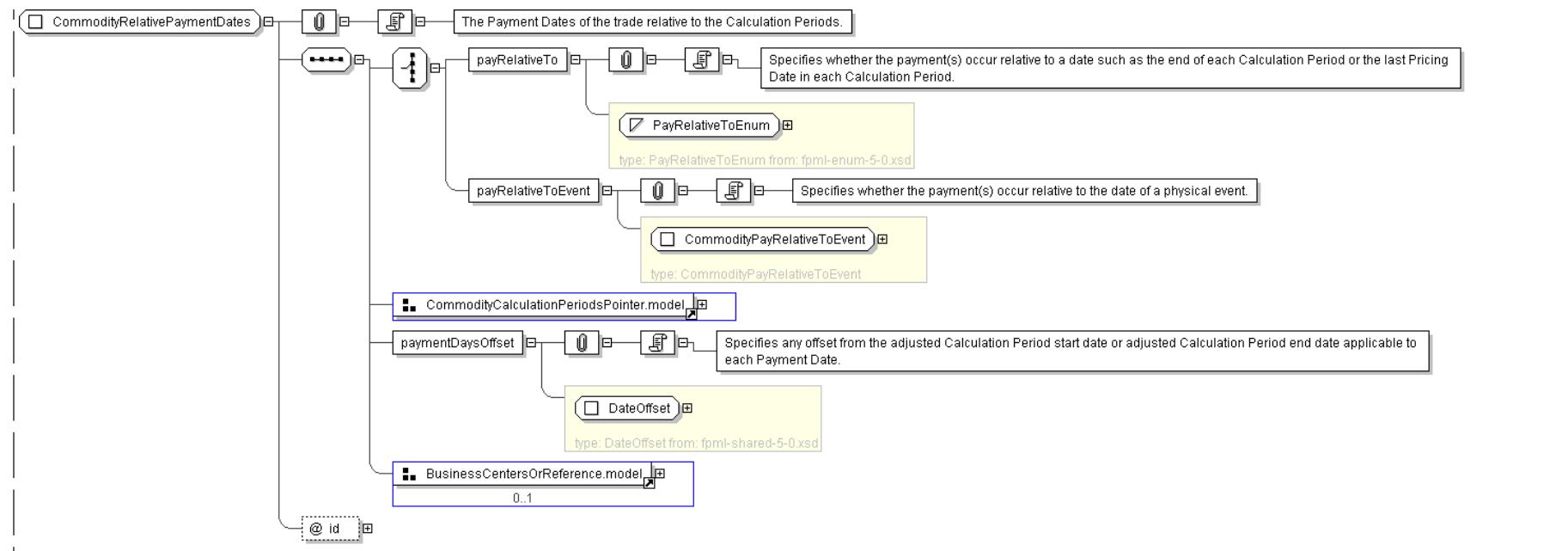
End Choice
<paymentDaysOffset> DateOffset </paymentDaysOffset> [1]
'Specifies any offset from the adjusted Calculation Period start date or adjusted
Calculation Period end date applicable to each Payment Date.'

Start Group: BusinessCentersOrReference.model [0..1]
Start Choice [1]
  <businessCentersReference> BusinessCentersReference </businessCentersReference> [1]
    '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.'

  <businessCenters> BusinessCenters </businessCenters> [1]

End Choice
End Group: BusinessCentersOrReference.model
</...>
```

**Diagram**

**Schema Component Representation**

```

<xsd:complexType name="CommodityRelativePaymentDates">
  <xsd:sequence>
    <xsd:choice>
      <xsd:element name="payRelativeTo" type=" PayRelativeToEnum " />
      <xsd:element name="payRelativeToEvent" type=" CommodityPayRelativeToEvent " />
    </xsd:choice>
    <xsd:group ref=" CommodityCalculationPeriodsPointer.model " />
    <xsd:element name="paymentDaysOffset" type=" DateOffset " />
    <xsd:group ref=" BusinessCentersOrReference.model " minOccurs="0" />
  </xsd:sequence>
  <xsd:attribute name="id" type=" xsd:ID " />
</xsd:complexType>

```

top

**Complex Type: CommoditySettlementPeriodsNotionalQuantity**

<b>Super-types:</b>	<a href="#">CommodityNotionalQuantity</a> < <b>CommoditySettlementPeriodsNotionalQuantity</b> (by extension)
<b>Sub-types:</b>	None

<b>Name</b>	CommoditySettlementPeriodsNotionalQuantity
<b>Used by (from the same schema document)</b>	Model Group <a href="#">CommodityNotionalQuantity.model</a>
<b>Abstract</b>	no
<b>Documentation</b>	The notional quantity of electricity that applies to one or more groups of Settlement Periods.

**XML Instance Representation**

```

<...
  id=" xsd:ID [0..1]">
  <quantityUnit> QuantityUnit </quantityUnit> [1]
  'Quantity Unit is the unit of measure applicable for the quantity on the Transaction.'

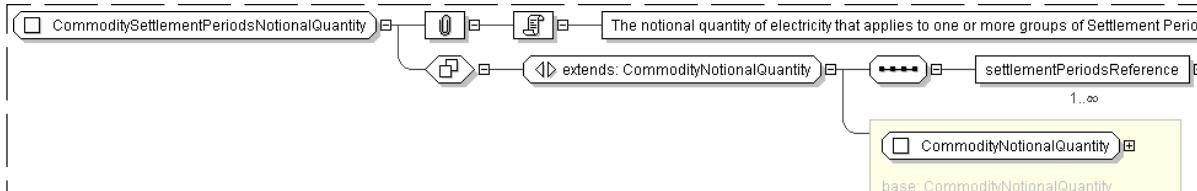
  <quantityFrequency> CommodityQuantityFrequency </quantityFrequency> [1]
  'The frequency at which the Notional Quantity is deemed to apply for purposes of
  calculating the Total Notional Quantity.'

  <quantity> xsd:decimal </quantity> [1]
  'Amount of commodity per quantity frequency.'

```

```
<settlementPeriodsReference> SettlementPeriodsReference </settlementPeriodsReference> [1..*]
'The range(s) of Settlement Periods to which the Notional Quantity applies.'
```

&lt;/...&gt;

**Diagram****Schema Component Representation**

```

<xsd:complexType name="CommoditySettlementPeriodsNotionalQuantity">
  <xsd:complexContent>
    <xsd:extension base="CommodityNotionalQuantity">
      <xsd:sequence>
        <xsd:element name="settlementPeriodsReference" type="SettlementPeriodsReference
          " maxOccurs="unbounded"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
  
```

[top](#)**Complex Type: [CommoditySettlementPeriodsNotionalQuantitySchedule](#)**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None

<b>Name</b>	CommoditySettlementPeriodsNotionalQuantitySchedule
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">CommodityNotionalQuantitySchedule</a>
<b>Abstract</b>	no
<b>Documentation</b>	The notional quantity schedule of electricity that applies to one or more groups of Settlement Periods.

**XML Instance Representation**

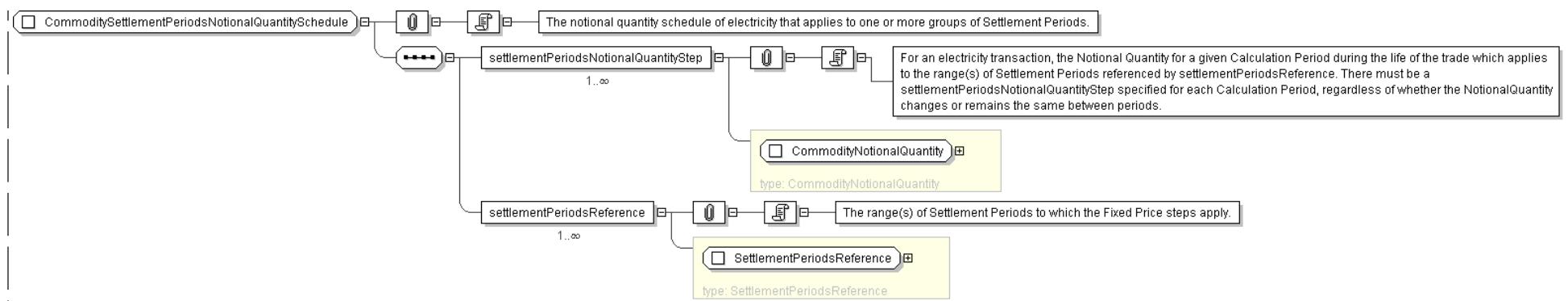
```

<...>
<settlementPeriodsNotionalQuantityStep> CommodityNotionalQuantity
</settlementPeriodsNotionalQuantityStep> [1..*]
'For an electricity transaction, the Notional Quantity for a given Calculation Period
during the life of the trade which applies to the range(s) of Settlement Periods referenced
by settlementPeriodsReference. There must be a settlementPeriodsNotionalQuantityStep
specified for each Calculation Period, regardless of whether the NotionalQuantity changes
or remains the same between periods.'

<settlementPeriodsReference> SettlementPeriodsReference </settlementPeriodsReference> [1..*]
'The range(s) of Settlement Periods to which the Fixed Price steps apply.'

</...>
  
```

**Diagram**

**Schema Component Representation**

```

<xsd:complexType name="CommoditySettlementPeriodsNotionalQuantitySchedule">
  <xsd:sequence>
    <xsd;element name="settlementPeriodsNotionalQuantityStep" type="CommodityNotionalQuantity"
      "maxOccurs="unbounded"/>
    <xsd;element name="settlementPeriodsReference" type="SettlementPeriodsReference"
      "maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>
  
```

[top](#)**Complex Type: CommoditySettlementPeriodsPriceSchedule**

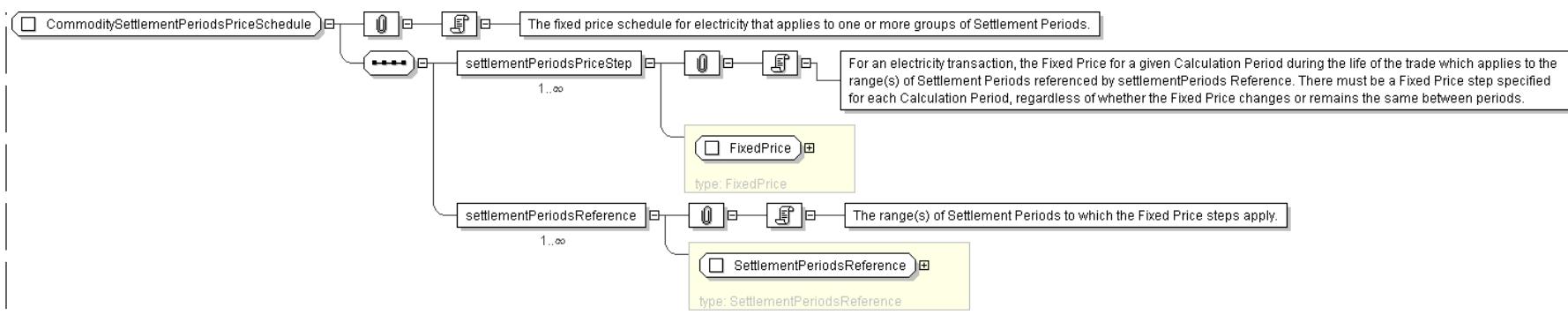
<b>Super-types:</b>	None
<b>Sub-types:</b>	None
<b>Name</b>	CommoditySettlementPeriodsPriceSchedule
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">CommodityFixedPriceSchedule</a>
<b>Abstract</b>	no
<b>Documentation</b>	The fixed price schedule for electricity that applies to one or more groups of Settlement Periods.

**XML Instance Representation**

```

<...>
<settlementPeriodsPriceStep> FixedPrice </settlementPeriodsPriceStep> [1..*]
'For an electricity transaction, the Fixed Price for a given Calculation Period during the life of the trade which applies to the range(s) of Settlement Periods referenced by settlementPeriods Reference. There must be a Fixed Price step specified for each Calculation Period, regardless of whether the Fixed Price changes or remains the same between periods.'
<settlementPeriodsReference> SettlementPeriodsReference </settlementPeriodsReference> [1..*]
'The range(s) of Settlement Periods to which the Fixed Price steps apply.'
</...>
  
```

**Diagram**

**Schema Component Representation**

```

<xsd:complexType name="CommoditySettlementPeriodsPriceSchedule">
  <xsd:sequence>
    <xsd:element name="settlementPeriodsPriceStep" type="FixedPrice" maxOccurs="unbounded"/>
    <xsd:element name="settlementPeriodsReference" type="SettlementPeriodsReference"
      " maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>
  
```

top

**Complex Type: CommoditySpreadSchedule**

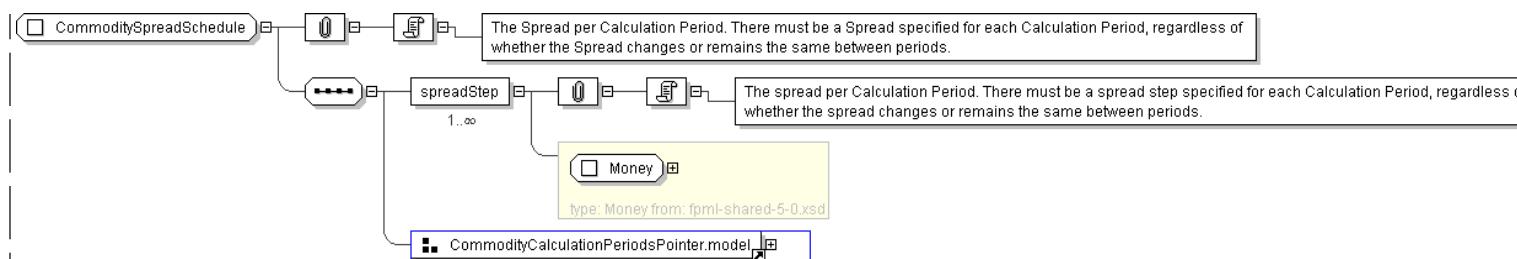
<b>Super-types:</b>	None
<b>Sub-types:</b>	None
<b>Name</b>	CommoditySpreadSchedule
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">FloatingLegCalculation</a>
<b>Abstract</b>	no
<b>Documentation</b>	The Spread per Calculation Period. There must be a Spread specified for each Calculation Period, regardless of whether the Spread changes or remains the same between periods.

**XML Instance Representation**

```

<...>
<spreadStep> Money </spreadStep> [1..*]
  'The spread per Calculation Period. There must be a spread step specified for each
  Calculation Period, regardless of whether the spread changes or remains the same
  between periods.'
Start Choice [1]
  <calculationPeriodsReference> CalculationPeriodsReference </calculationPeriodsReference> [1]
  'A pointer style reference to the Calculation Periods defined on another leg.'
  <calculationPeriodsScheduleReference> CalculationPeriodsScheduleReference
  </calculationPeriodsScheduleReference> [1]
  'A pointer style reference to the Calculation Periods Schedule defined on another leg.'
  <calculationPeriodsDatesReference> CalculationPeriodsDatesReference
  </calculationPeriodsDatesReference> [1]
  'A pointer style reference to single-day-duration Calculation Periods defined on another leg.'
End Choice
</...>
  
```

**Diagram**

**Schema Component Representation**

```

<xsd:complexType name="CommoditySpreadSchedule">
  <xsd:sequence>
    <xsd:element name="spreadStep" type="#Money" maxOccurs="unbounded"/>
    <xsd:group ref="CommodityCalculationPeriodsPointer.model" />
  </xsd:sequence>
</xsd:complexType>
  
```

top

**Complex Type: CommodityStrikeSchedule**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None

<b>Name</b>	CommodityStrikeSchedule
<b>Used by (from the same schema document)</b>	Model Group <b>CommodityStrikePrice.model</b>
<b>Abstract</b>	no
<b>Documentation</b>	The Strike Price per Unit per Calculation Period. There must be a Strike Price per Unit step specified for each Calculation Period, regardless of whether the Strike changes or remains the same between periods.

**XML Instance Representation**

```

<...>
  <strikePricePerUnitStep> NonNegativeMoney </strikePricePerUnitStep> [1..*]
  'The strike price per unit per Calculation Period. There must be a strike price per
  unit specified for each Calculation Period, regardless of whether the price changes or
  remains the same between periods.'

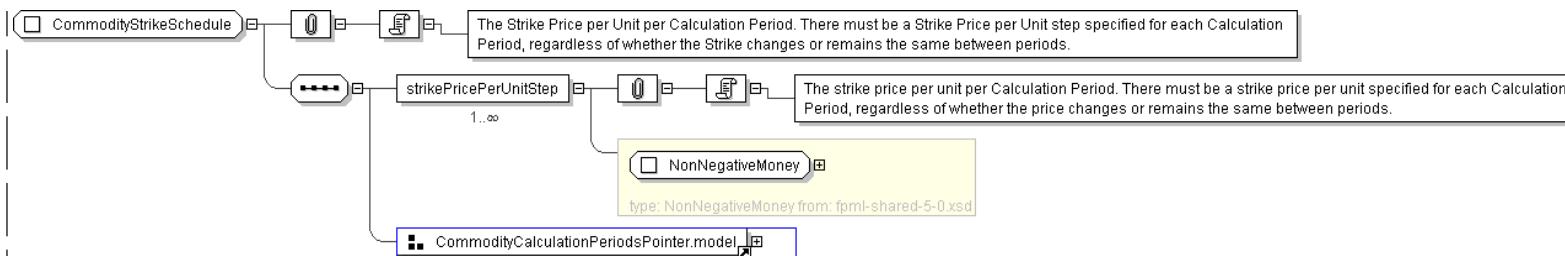
```

Start **Choice** [1]

- <calculationPeriodsReference> **CalculationPeriodsReference** </calculationPeriodsReference> [1]
 'A pointer style reference to the Calculation Periods defined on another leg.'
- <calculationPeriodsScheduleReference> **CalculationPeriodsScheduleReference** </calculationPeriodsScheduleReference> [1]
 'A pointer style reference to the Calculation Periods Schedule defined on another leg.'
- <calculationPeriodsDatesReference> **CalculationPeriodsDatesReference** </calculationPeriodsDatesReference> [1]
 'A pointer style reference to single-day-duration Calculation Periods defined on another leg.'

End **Choice**

**Diagram**

**Schema Component Representation**

```

<xsd:complexType name="CommodityStrikeSchedule">
  <xsd:sequence>
    <xsd:element name="strikePricePerUnitStep" type="NonNegativeMoney" maxOccurs="unbounded" />
    <xsd:group ref="CommodityCalculationPeriodsPointer.model" />
  </xsd:sequence>
</xsd:complexType>
  
```

top

**Complex Type: CommoditySwap**

<b>Super-types:</b>	<a href="#">Product</a> < <b>CommoditySwap</b> (by extension)
<b>Sub-types:</b>	None
<b>Name</b>	CommoditySwap
<b>Used by (from the same schema document)</b>	Element <a href="#">commoditySwap</a>
<b>Abstract</b>	no
<b>Documentation</b>	Commodity Swap.

**XML Instance Representation**

```

<...
  id="xsd:ID [0..1]">
  <productType> ProductType </productType> [0..*]
  'A classification of the type of product. FpML defines a simple product categorization using
  a coding scheme.'
  <productId> ProductId </productId> [0..*]
  'A product 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.'
  <effectiveDate> AdjustableOrRelativeDate </effectiveDate> [1]
  'Specifies the effective date of this leg of the swap. When defined in relation to a
  date specified somewhere else in the document (through the relativeDate component),
  this element will typically point to the effective date of the other leg of the swap.'
  <terminationDate> AdjustableOrRelativeDate </terminationDate> [1]
  'Specifies the termination date of this leg of the swap. When defined in relation to a
  date specified somewhere else in the document (through the relativeDate component),
  this element will typically point to the termination date of the other leg of the swap.'
  <settlementCurrency> IdentifiedCurrency </settlementCurrency> [1]
  'The currency into which the Commodity Swap Transaction will settle. If this is not the same
  as the currency in which the Commodity Reference Price is quoted on a given floating leg of
  the Commodity Swap Transaction, then an FX rate should also be specified for that leg.'
  Start Choice [1..*]
    <fixedLeg> FixedPriceLeg </fixedLeg> [1]
    'Fixed Price Leg.'
    <floatingLeg> FloatingPriceLeg </floatingLeg> [1]
    'Floating Price leg.'
    <coalPhysicalLeg> CoalPhysicalLeg </coalPhysicalLeg> [1]
    'Physically settled coal leg.'
  '

```

```

<electricityPhysicalLeg> ElectricityPhysicalLeg </electricityPhysicalLeg> [1]
'Physically settled electricity leg.'

<gasPhysicalLeg> GasPhysicalLeg </gasPhysicalLeg> [1]
'Physically settled natural gas leg.'

<oilPhysicalLeg> OilPhysicalLeg </oilPhysicalLeg> [1]
'Physically settled oil or refined products leg.'

<additionalCommoditySwapLeg> ... </additionalCommoditySwapLeg> [1]

End Choice
Start Group: CommodityContent.model [0..1]
<commonPricing> xsd:boolean </commonPricing> [0..1]
'Common pricing may be relevant for a Transaction that references more than one
Commodity Reference Price. If Common Pricing is not specified as applicable, it will be
deemed not to apply.'

<marketDisruption> CommodityMarketDisruption </marketDisruption> [0..1]
'Market disruption events as defined in the ISDA 1993 Commodity Definitions or in ISDA
2005 Commodity Definitions, as applicable.'

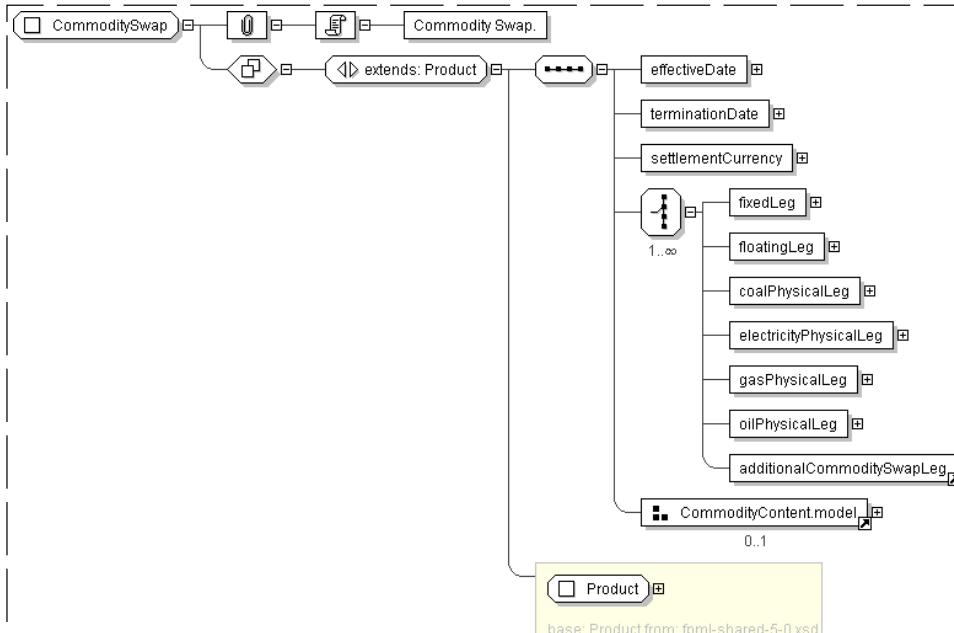
<settlementDisruption> CommodityBullionSettlementDisruptionEnum </settlementDisruption> [0..1]
'The consequences of Bullion Settlement Disruption Events.'

<rounding> Rounding </rounding> [0..1]
'Rounding direction and precision for amounts.'

```

End Group: CommodityContent.model

&lt;...&gt;

**Diagram****Schema Component Representation**

```

<xsd:complexType name="CommoditySwap">
  <xsd:complexContent>
    <xsd:extension base=" Product ">
      <xsd:sequence>
        <xsd:element name="effectiveDate" type=" AdjustableOrRelativeDate "/>
        <xsd:element name="terminationDate" type=" AdjustableOrRelativeDate "/>
        <xsd:element name="settlementCurrency" type=" IdentifiedCurrency "/>
        <xsd:element name="fixedLeg" type=" sequence ">
          <xsd:sequence>
            <xsd:element name="floatingLeg" type=" sequence ">
              <xsd:sequence>
                <xsd:element name="coalPhysicalLeg" type=" string "/>
                <xsd:element name="electricityPhysicalLeg" type=" string "/>
                <xsd:element name="gasPhysicalLeg" type=" string "/>
                <xsd:element name="oilPhysicalLeg" type=" string "/>
                <xsd:element name="additionalCommoditySwapLeg" type=" sequence ">
                  <xsd:sequence>
                    <xsd:element name="CommodityContent.model" type=" string "/>

```

```

<xsd:choice maxOccurs="unbounded">
  <xsd:element name="fixedLeg" type=" FixedPriceLeg " />
  <xsd:element name="floatingLeg" type=" FloatingPriceLeg " />
  <xsd:element name="coalPhysicalLeg" type=" CoalPhysicalLeg " />
  <xsd:element name="electricityPhysicalLeg" type=" ElectricityPhysicalLeg " />
  <xsd:element name="gasPhysicalLeg" type=" GasPhysicalLeg " />
  <xsd:element name="oilPhysicalLeg" type=" OilPhysicalLeg " />
  <xsd:element ref=" additionalCommoditySwapLeg " />
</xsd:choice>
<xsd:group ref=" CommodityContent.model " minOccurs="0 " />
</xsd:sequence>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>

```

[top](#)

## Complex Type: DisruptionFallback

Super-types:	<a href="#">Scheme</a> < DisruptionFallback (by extension)
Sub-types:	None

Name	DisruptionFallback
Used by (from the same schema document)	Complex Type <a href="#">SequencedDisruptionFallback</a>
Abstract	no
Documentation	A Disruption Fallback.

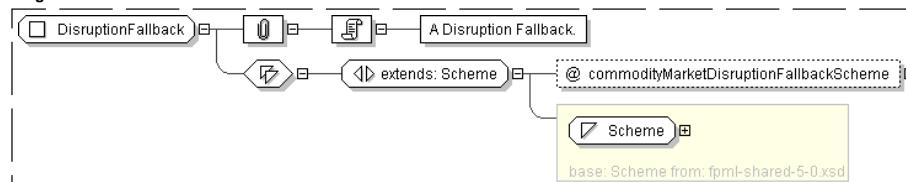
### XML Instance Representation

```

<...
  commodityMarketDisruptionFallbackScheme=" xsd:anyURI [0..1]">
  Scheme
</...>

```

### Diagram



### Schema Component Representation

```

<xsd:complexType name="DisruptionFallback">
  <xsd:simpleContent>
    <xsd:extension base=" Scheme ">
      <xsd:attribute name="commodityMarketDisruptionFallbackScheme" type=" xsd:anyURI
        " default="http://www.fpml.org/coding-scheme/commodity-market-disruption-fallback"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>

```

[top](#)

## Complex Type: ElectricityDelivery

Super-types:	None
Sub-types:	None

Name	ElectricityDelivery
Used by (from the same schema document)	Complex Type <a href="#">ElectricityPhysicalLeg</a>
Abstract	no
Documentation	The physical delivery conditions for electricity.

### XML Instance Representation

```

<...>
| 

```

```

Start Choice [1]
<deliveryPoint> ElectricityDeliveryPoint </deliveryPoint> [1]
'The point at which delivery of the electricity will occur.'

<deliveryType> [0..1]
'Indicates the under what conditions the Parties' delivery obligations apply.'

Start Choice [1]
<firm> ElectricityDeliveryFirm </firm> [1]
'Indicates under what conditions the Parties' delivery obligations apply.'

<nonFirm> xsd:boolean </nonFirm> [1]
'If present and set to true, indicates that delivery or receipt of the electricity may
be interrupted for any reason or for no reason, without liability on the part of either
Party. This element should never have a value of false.'

<systemFirm> ElectricityDeliverySystemFirm </systemFirm> [1]
'Indicates that the electricity is intended to be supplied from the owned or
controlled generation or pre-existing purchased power assets of the system specified.'

<unitFirm> ElectricityDeliveryUnitFirm </unitFirm> [1]
'Indicates that the electricity is intended to be supplied from a generation asset which
can optionally be specified.'

End Choice
</deliveryType>
<transmissionContingency> ElectricityTransmissionContingency </transmissionContingency> [0..1]
'Indicates that the performance of the buyer or seller shall be excused (under the
conditions specified) if transmission of the electricity is unavailable or interrupted.'

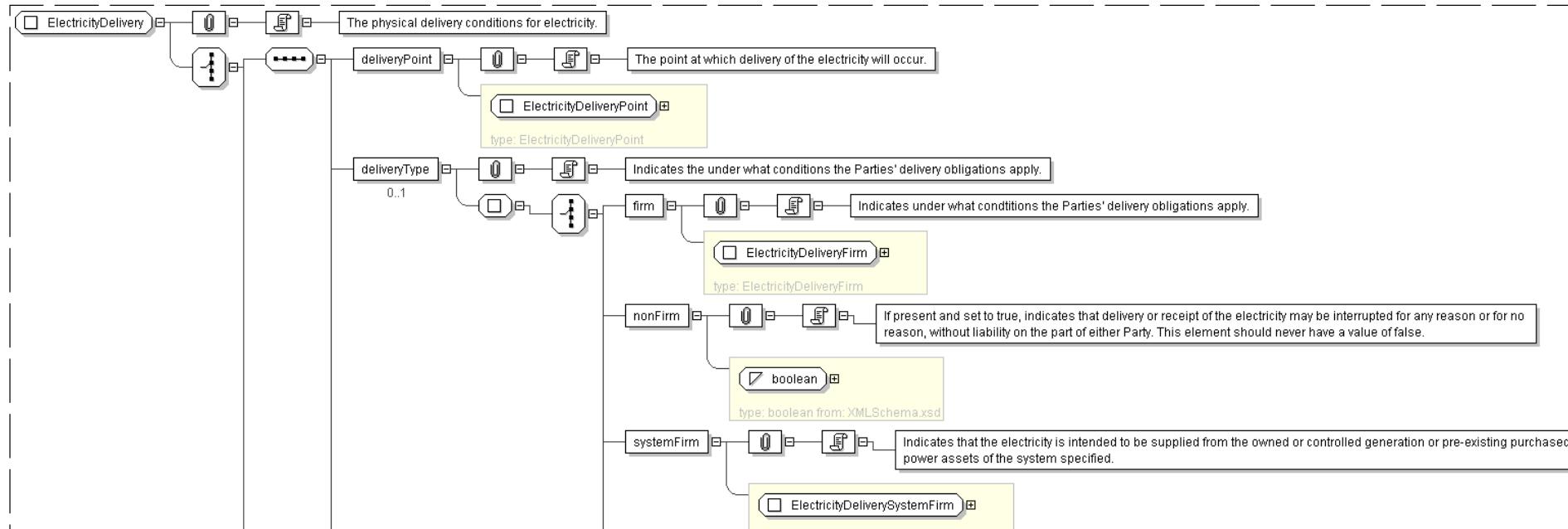
<deliveryZone> CommodityDeliveryPoint </deliveryZone> [1]
'The zone covering potential delivery points for the electricity.'

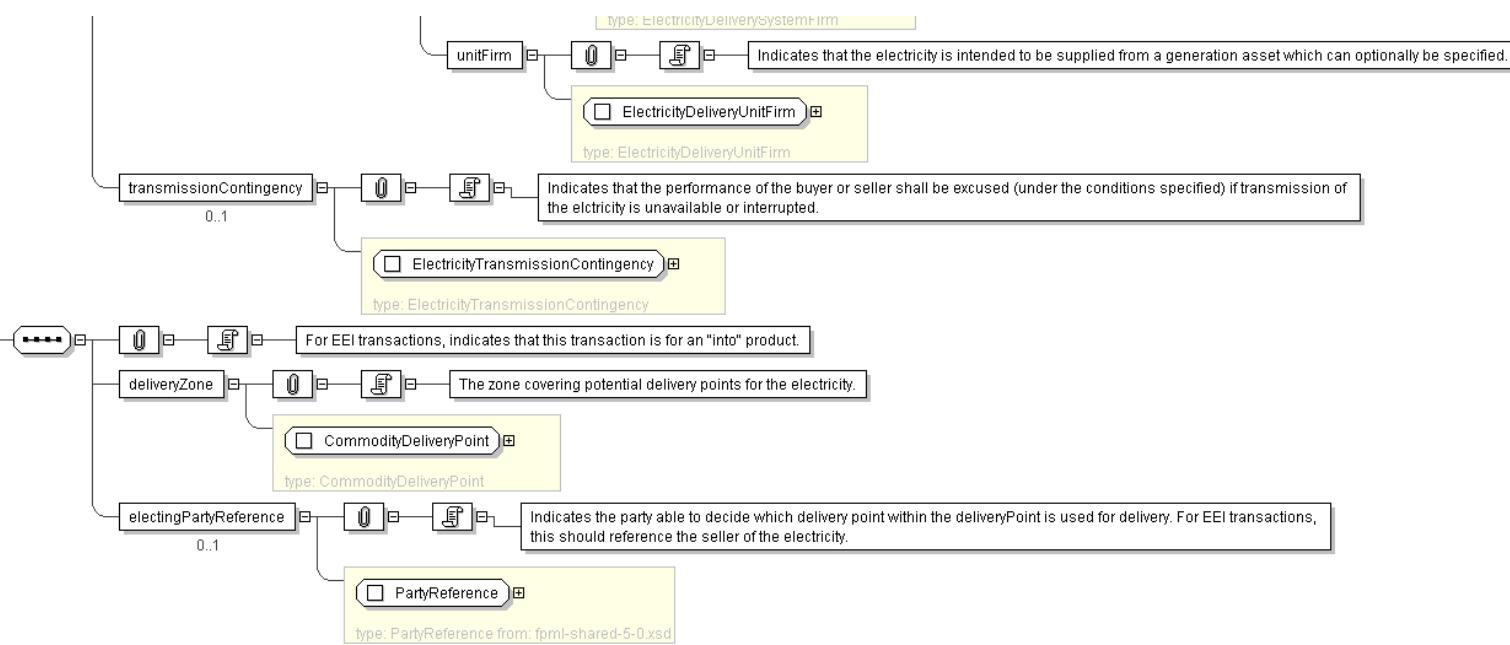
<electingPartyReference> PartyReference </electingPartyReference> [0..1]
'Indicates the party able to decide which delivery point within the deliveryPoint is used
for delivery. For EBI transactions, this should reference the seller of the electricity.'

End Choice
</...>

```

## Diagram



**Schema Component Representation**

```

<xsd:complexType name="ElectricityDelivery">
  <xsd:choice>
    <xsd:sequence>
      <xsd:element name="deliveryPoint" type=" ElectricityDeliveryPoint " />
      <xsd:element name="deliveryType" minOccurs="0">
        <xsd:complexType>
          <xsd:choice>
            <xsd:element name="firm" type=" ElectricityDeliveryFirm " />
            <xsd:element name="nonFirm" type=" xsd:boolean " />
            <xsd:element name="systemFirm" type=" ElectricityDeliverySystemFirm " />
            <xsd:element name="unitFirm" type=" ElectricityDeliveryUnitFirm " />
          </xsd:choice>
        </xsd:complexType>
      </xsd:element>
      <xsd:element name="transmissionContingency" type=" ElectricityTransmissionContingency "
        " minOccurs="0" />
    </xsd:sequence>
    <xsd:sequence>
      <xsd:element name="deliveryZone" type=" CommodityDeliveryPoint " />
      <xsd:element name="electingPartyReference" type=" PartyReference " minOccurs="0"/>
    </xsd:sequence>
  </xsd:choice>
</xsd:complexType>

```

top

**Complex Type: ElectricityDeliveryFirm**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None

<b>Name</b>	<code>ElectricityDeliveryFirm</code>
<b>Used by (from the same schema document)</b>	Complex Type <code>ElectricityDelivery</code>
<b>Abstract</b>	no
<b>Documentation</b>	The physical delivery obligation options specific to a firm transaction.

**XML Instance Representation**

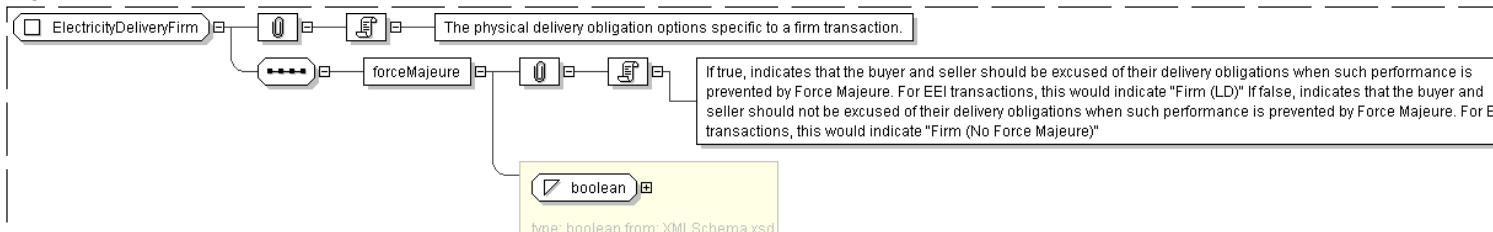
```

<...>
  <forceMajeure> xsd:boolean </forceMajeure> [1]

```

'If true, indicates that the buyer and seller should be excused of their delivery obligations when such performance is prevented by Force Majeure. For EEI transactions, this would indicate "Firm (LD)". If false, indicates that the buyer and seller should not be excused of their delivery obligations when such performance is prevented by Force Majeure. For EEI transactions, this would indicate "Firm (No Force Majeure)"'

&lt;/...&gt;

**Diagram****Schema Component Representation**

```

<xsd:complexType name="ElectricityDeliveryFirm">
  <xsd:sequence>
    <xsd:element name="forceMajeure" type="xsd:boolean" />
  </xsd:sequence>
</xsd:complexType>

```

[top](#)**Complex Type: ElectricityDeliveryPeriods**

<b>Super-types:</b>	CommodityDeliveryPeriods < ElectricityDeliveryPeriods (by extension)
---------------------	--

<b>Sub-types:</b>	None
-------------------	------

<b>Name</b>	ElectricityDeliveryPeriods
-------------	----------------------------

<b>Abstract</b>	no
-----------------	----

<b>Documentation</b>	The different options for specifying the Delivery Periods for a physically settled electricity trade.
----------------------	---

**XML Instance Representation**

```

<...
  id="xsd:ID [0..1]">
  Start Choice [1]
  <periods> AdjustableDates </periods> [1]
  <!-- The Delivery Periods for this leg of the swap. This type is only intended to be used if
      the Delivery Periods differ from the Calculation Periods on the fixed or floating leg.
      If DeliveryPeriods mirror another leg, then the calculationPeriodsReference element should
      be used to point to the Calculation Periods on that leg - or
      the calculationPeriodsScheduleReference can be used to point to the Calculation
      Periods Schedule for that leg. -->
  <periodsSchedule> CommodityCalculationPeriodsSchedule </periodsSchedule> [1]
  <!-- The Delivery Periods for this leg of the swap. This type is only intended to be used if
      the Delivery Periods differ from the Calculation Periods on the fixed or floating leg.
      If DeliveryPeriods mirror another leg, then the calculationPeriodsReference element should
      be used to point to the Calculation Periods on that leg - or
      the calculationPeriodsScheduleReference can be used to point to the Calculation
      Periods Schedule for that leg. -->

```

```

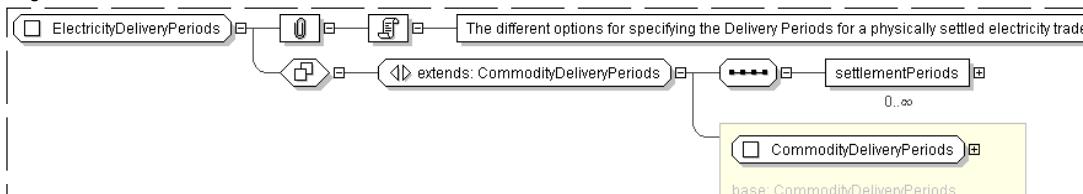
  Start Choice [1]
  <calculationPeriodsReference> CalculationPeriodsReference </calculationPeriodsReference> [1]
  <!-- A pointer style reference to the Calculation Periods defined on another leg. -->
  <calculationPeriodsScheduleReference> CalculationPeriodsScheduleReference
  </calculationPeriodsScheduleReference> [1]
  <!-- A pointer style reference to the Calculation Periods Schedule defined on another leg. -->
  <calculationPeriodsDatesReference> CalculationPeriodsDatesReference
  </calculationPeriodsDatesReference> [1]
  <!-- A pointer style reference to single-day-duration Calculation Periods defined on another leg. -->

```

```

End Choice
End Choice
<settlementPeriods> SettlementPeriods </settlementPeriods> [0..*]
  'The periods within the Delivery Periods during which the electricity will be delivered.'
</...>

```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="ElectricityDeliveryPeriods">
  <xsd:complexContent>
    <xsd:extension base="#CommodityDeliveryPeriods">
      <xsd:sequence>
        <xsd:element name="settlementPeriods" type="#SettlementPeriods" 
          minOccurs="0" maxOccurs="unbounded"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

top

**Complex Type: ElectricityDeliveryPoint**

<b>Super-types:</b>	Scheme < ElectricityDeliveryPoint (by extension)
<b>Sub-types:</b>	None

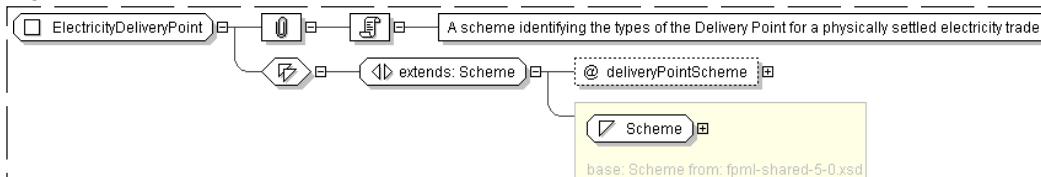
<b>Name</b>	ElectricityDeliveryPoint
<b>Used by (from the same schema document)</b>	Complex Type ElectricityDelivery
<b>Abstract</b>	no
<b>Documentation</b>	A scheme identifying the types of the Delivery Point for a physically settled electricity trade.

**XML Instance Representation**

```

<...
  deliveryPointScheme=" xsd:anyURI [0..1]">
  Scheme
<...>

```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="ElectricityDeliveryPoint">
  <xsd:simpleContent>
    <xsd:extension base="#Scheme">
      <xsd:attribute name="deliveryPointScheme" type="xsd:anyURI" />
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>

```

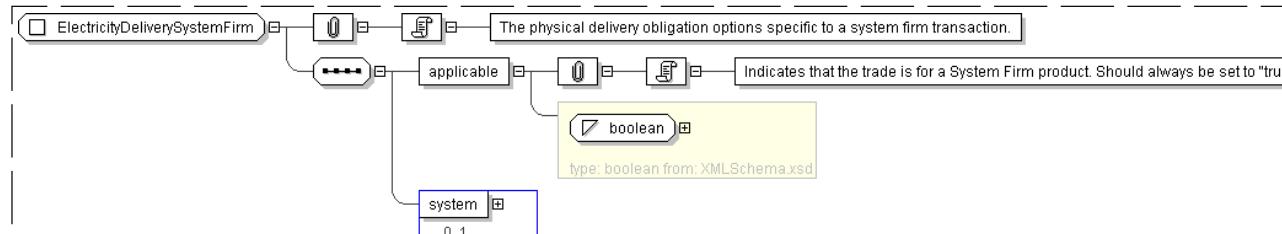
**Complex Type: ElectricityDeliverySystemFirm**

Super-types:	None
Sub-types:	None

Name	ElectricityDeliverySystemFirm
Used by (from the same schema document)	Complex Type <a href="#">ElectricityDelivery</a>
Abstract	no
Documentation	The physical delivery obligation options specific to a system firm transaction.

**XML Instance Representation**

```
<...>
  <applicable> xsd:boolean </applicable> [1]
    'Indicates that the trade is for a System Firm product. Should always be set to "true".'
  <system> CommodityDeliveryPoint </system> [0..1]
</...>
```

**Diagram****Schema Component Representation**

```
<xsd:complexType name="ElectricityDeliverySystemFirm">
  <xsd:sequence>
    <xsd:element name="applicable" type=" xsd:boolean " />
    <xsd:element name="system" type=" CommodityDeliveryPoint " minOccurs="0" />
  </xsd:sequence>
</xsd:complexType>
```

top

**Complex Type: ElectricityDeliveryUnitFirm**

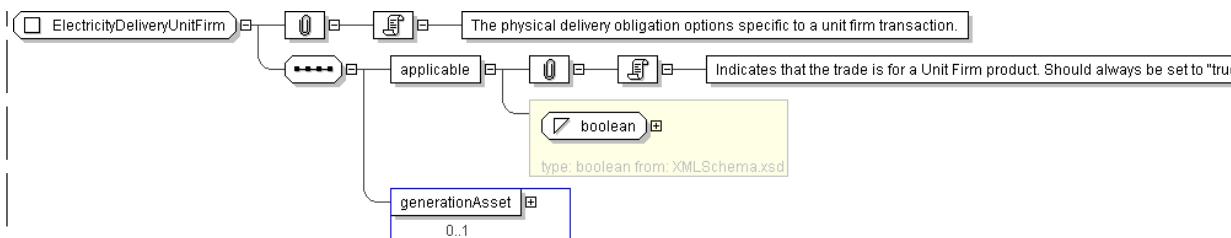
Super-types:	None
Sub-types:	None

Name	ElectricityDeliveryUnitFirm
Used by (from the same schema document)	Complex Type <a href="#">ElectricityDelivery</a>
Abstract	no
Documentation	The physical delivery obligation options specific to a unit firm transaction.

**XML Instance Representation**

```
<...>
  <applicable> xsd:boolean </applicable> [1]
    'Indicates that the trade is for a Unit Firm product. Should always be set to "true".'
  <generationAsset> CommodityDeliveryPoint </generationAsset> [0..1]
</...>
```

**Diagram**

**Schema Component Representation**

```
<xsd:complexType name="ElectricityDeliveryUnitFirm">
  <xsd:sequence>
    <xsd:element name="applicable" type="xsd:boolean" />
    <xsd:element name="generationAsset" type="#CommodityDeliveryPoint" minOccurs="0" />
  </xsd:sequence>
</xsd:complexType>
```

top

**Complex Type: ElectricityPhysicalDeliveryQuantity**

<b>Super-types:</b>	<a href="#">CommodityNotionalQuantity</a> < <b>ElectricityPhysicalDeliveryQuantity</b> (by extension)
<b>Sub-types:</b>	None
<b>Name</b>	ElectricityPhysicalDeliveryQuantity
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">ElectricityPhysicalQuantity</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type defining the physical quantity of the electricity to be delivered.

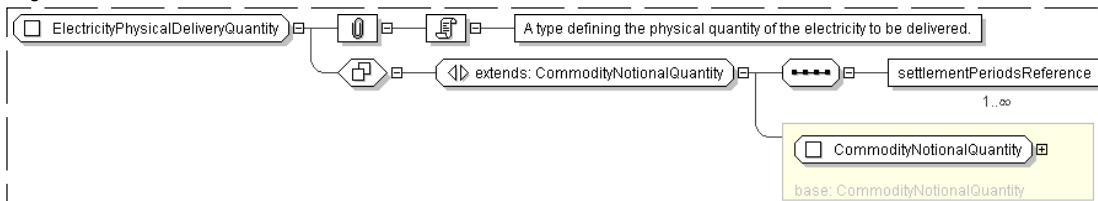
**XML Instance Representation**

```
<...
  id="xsd:ID [0..1]">
  <quantityUnit> QuantityUnit </quantityUnit> [1]
  'Quantity Unit is the unit of measure applicable for the quantity on the Transaction.'

  <quantityFrequency> CommodityQuantityFrequency </quantityFrequency> [1]
  'The frequency at which the Notional Quantity is deemed to apply for purposes of
  calculating the Total Notional Quantity.'

  <quantity> xsd:decimal </quantity> [1]
  'Amount of commodity per quantity frequency.'

  <settlementPeriodsReference> SettlementPeriodsReference </settlementPeriodsReference> [1..*]
  'A pointer style reference to the range(s) of Settlement Periods to which this
  quantity applies.'
</...>
```

**Diagram****Schema Component Representation**

```
<xsd:complexType name="ElectricityPhysicalDeliveryQuantity">
  <xsd:complexContent>
    <xsd:extension base="#CommodityNotionalQuantity">
      <xsd:sequence>
        <xsd:element name="settlementPeriodsReference" type="#SettlementPeriodsReference">
          <xsd:annotation>
            <description>A pointer style reference to the range(s) of Settlement Periods to which this quantity applies.</description>
          </xsd:annotation>
        </xsd:element>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

```

    " maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>

```

## Complex Type: ElectricityPhysicalDeliveryQuantitySchedule

**Super-types:** [CommodityPhysicalQuantitySchedule](#) < **ElectricityPhysicalDeliveryQuantitySchedule** (by extension)

**Sub-types:** None

**Name** ElectricityPhysicalDeliveryQuantitySchedule

**Used by (from the same schema document)** Complex Type [ElectricityPhysicalQuantity](#)

**Abstract** no

**Documentation** Allows the documentation of a shaped quantity trade where the quantity changes over the life of the transaction.

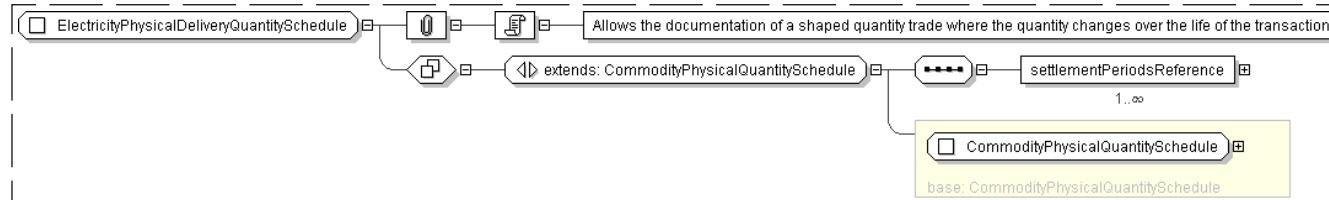
### XML Instance Representation

```

<...
  id=" xsd:ID [0..1]">
  <quantityStep> CommodityNotionalQuantity </quantityStep> [1...*]
    'The quantity per Calculation Period. There must be a quantity specified for each
    Calculation Period, regardless of whether the quantity changes or remains the same
    between periods.'
  Start Choice [1]
    <deliveryPeriodsReference> CalculationPeriodsReference </deliveryPeriodsReference> [1]
      'A pointer style reference to the Delivery Periods defined elsewhere.'
    <deliveryPeriodsScheduleReference> CalculationPeriodsScheduleReference
    </deliveryPeriodsScheduleReference> [1]
      'A pointer style reference to the Calculation Periods Schedule defined elsewhere.'
  End Choice
  <settlementPeriodsReference> SettlementPeriodsReference </settlementPeriodsReference> [1...*]
    'A pointer style reference to the range(s) of Settlement Periods to which this
    quantity applies.'
</...>

```

### Diagram



### Schema Component Representation

```

<xsd:complexType name="ElectricityPhysicalDeliveryQuantitySchedule">
  <xsd:complexContent>
    <xsd:extension base=" CommodityPhysicalQuantitySchedule ">
      <xsd:sequence>
        <xsd:element name="settlementPeriodsReference" type=" SettlementPeriodsReference
          " maxOccurs="unbounded"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

## Complex Type: ElectricityPhysicalLeg

<b>Super-types:</b>	<a href="#">Leg</a> < <a href="#">PhysicalLeg</a> (by extension) < <b>ElectricityPhysicalLeg</b> (by extension)
<b>Sub-types:</b>	None
<b>Name</b>	<b>ElectricityPhysicalLeg</b>
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">CommoditySwap</a>
<b>Abstract</b>	no
<b>Documentation</b>	Physically settled leg of a physically settled electricity transaction.

**XML Instance Representation**

```
<...>
<id=" xsd:ID [0..1]">
<payerPartyReference> PartyReference </payerPartyReference> [1]
'A reference to the party responsible for making the payments defined by this structure.'

<payerAccountReference> AccountReference </payerAccountReference> [0..1]
'A reference to the account responsible for making the payments defined by this structure.'

<receiverPartyReference> PartyReference </receiverPartyReference> [1]
'A reference to the party that receives the payments corresponding to this structure.'

<receiverAccountReference> AccountReference </receiverAccountReference> [0..1]
'A reference to the account that receives the payments corresponding to this structure.'

<deliveryPeriods> CommodityDeliveryPeriods </deliveryPeriods> [1]
'The different options for specifying the Delivery or Supply Periods. Unless the quantity
or price is to vary periodically during the trade or physical delivery occurs on a
periodic basis, periodsSchedule should be used and set to IT.'

<settlementPeriods> SettlementPeriods </settlementPeriods> [1..*]
'The specification of the Settlement Periods in which the electricity will be delivered.
The Settlement Periods will apply from and including the Effective Date up to and including
the Termination Date. If more than one settlementPeriods element is present this
indicates multiple ranges of Settlement Periods apply to the entire trade - for example
off-peak weekdays and all day weekends. Settlement Period ranges should not overlap.'

<settlementPeriodsSchedule> SettlementPeriodsSchedule </settlementPeriodsSchedule> [0..1]
'The specification of the Settlement Periods in which the electricity will be delivered for
a \"shaped\" trade i.e. where different Settlement Period ranges will apply to
different periods of the trade.'

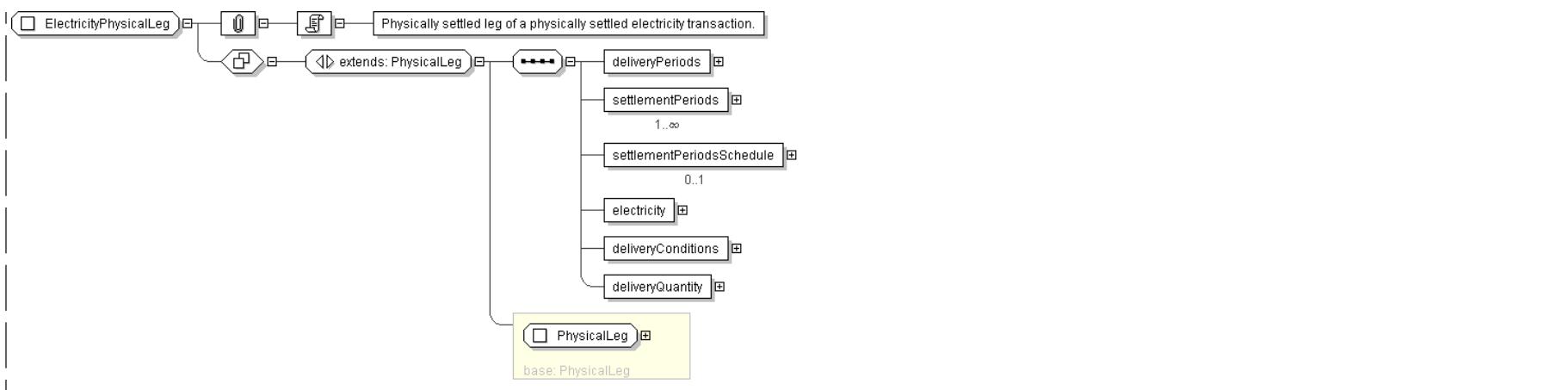
<electricity> ElectricityProduct </electricity> [1]
'The specification of the electricity to be delivered.'

<deliveryConditions> ElectricityDelivery </deliveryConditions> [1]
'The physical delivery conditions for the transaction.'

<deliveryQuantity> ElectricityPhysicalQuantity </deliveryQuantity> [1]
'The different options for specifying the quantity.'

</...>
```

**Diagram**

**Schema Component Representation**

```

<xsd:complexType name="ElectricityPhysicalLeg">
  <xsd:complexContent>
    <xsd:extension base=" PhysicalLeg ">
      <xsd:sequence>
        <xsd:element name="deliveryPeriods" type=" CommodityDeliveryPeriods "/>
        <xsd:element name="settlementPeriods" type=" SettlementPeriods " maxOccurs="unbounded"/>
        <xsd:element name="settlementPeriodsSchedule" type=" SettlementPeriodsSchedule " minOccurs="0"/>
        <xsd:element name="electricity" type=" ElectricityProduct "/>
        <xsd:element name="deliveryConditions" type=" ElectricityDelivery "/>
        <xsd:element name="deliveryQuantity" type=" ElectricityPhysicalQuantity "/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
  
```

top

**Complex Type: ElectricityPhysicalQuantity**

<b>Super-types:</b>	<a href="#">CommodityPhysicalQuantityBase</a> < ElectricityPhysicalQuantity (by extension)
<b>Sub-types:</b>	None
<b>Name</b>	ElectricityPhysicalQuantity
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">ElectricityPhysicalLeg</a>
<b>Abstract</b>	no
<b>Documentation</b>	The quantity of gas to be delivered.

**XML Instance Representation**

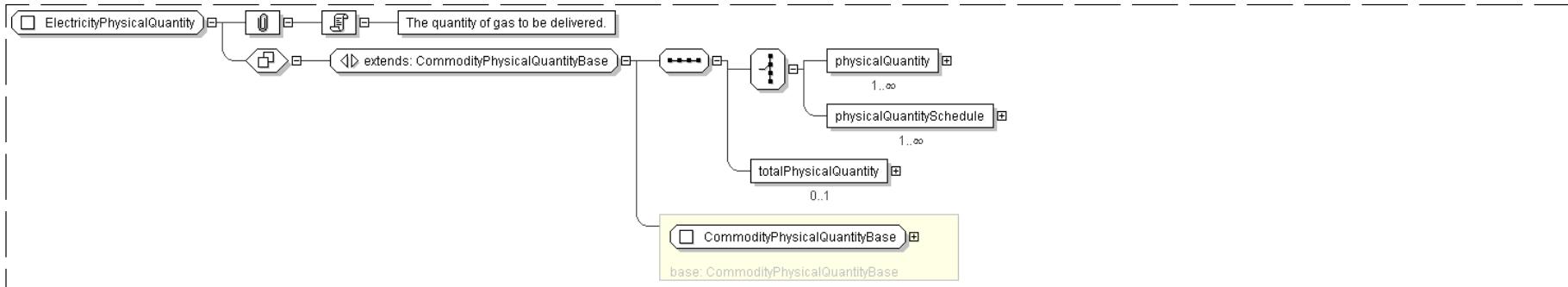
```

<...
id=" xsd:ID [0..1]">
Start Choice [1]
  <physicalQuantity> ElectricityPhysicalDeliveryQuantity </physicalQuantity> [1..*]
  'The Quantity per Delivery Period.'

  <physicalQuantitySchedule> ElectricityPhysicalDeliveryQuantitySchedule
  </physicalQuantitySchedule> [1..*]
  'Allows the documentation of a shaped quantity trade where the quantity changes over the
  life of the transaction. Note that if the range of Settlement Periods also varies over the
  life of the transaction this element should not be used. Instead, physicalquantity should
  be repeated for each range of Settlement Periods that apply at any point during the trade.'

End Choice
  <totalPhysicalQuantity> UnitQuantity </totalPhysicalQuantity> [0..1]
  'The Total Quantity of the commodity to be delivered.'

</...>
  
```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="ElectricityPhysicalQuantity">
  <xsd:complexContent>
    <xsd:extension base="CommodityPhysicalQuantityBase">
      <xsd:sequence>
        <xsd:choice>
          <xsd:element name="physicalQuantity" type="ElectricityPhysicalDeliveryQuantity"
            " maxOccurs="unbounded"/>
          <xsd:element name="physicalQuantitySchedule" type="ElectricityPhysicalDeliveryQuantitySchedule"
            " maxOccurs="unbounded"/>
        </xsd:choice>
        <xsd:element name="totalPhysicalQuantity" type="UnitQuantity" minOccurs="0"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
  
```

top

**Complex Type: ElectricityProduct**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None

<b>Name</b>	ElectricityProduct
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">ElectricityPhysicalLeg</a>
<b>Abstract</b>	no
<b>Documentation</b>	The specification of the electricity to be delivered.

**XML Instance Representation**

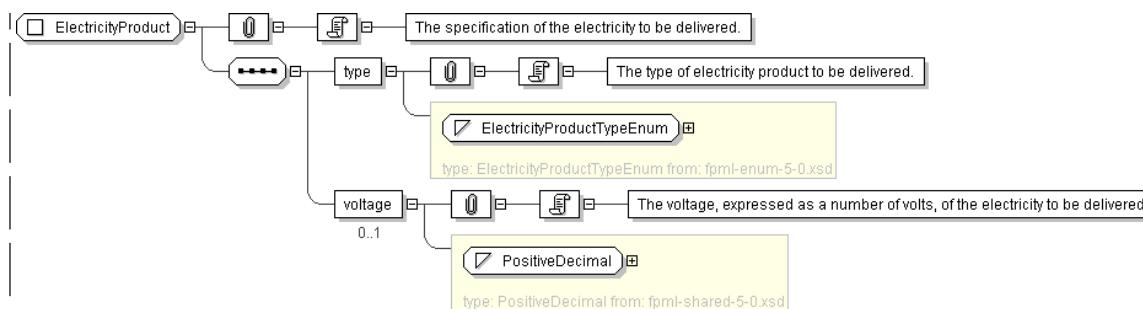
```

<...>
<type> ElectricityProductTypeEnum </type> [1]
'The type of electricity product to be delivered.'

<voltage> PositiveDecimal </voltage> [0..1]
'The voltage, expressed as a number of volts, of the electricity to be delivered.'

</...>
  
```

**Diagram**

**Schema Component Representation**

```
<xsd:complexType name="ElectricityProduct">
  <xsd:sequence>
    <xsd:element name="type" type="#ElectricityProductTypeEnum" />
    <xsd:element name="voltage" type="#PositiveDecimal" minOccurs="0" />
  </xsd:sequence>
</xsd:complexType>
```

top

**Complex Type: ElectricityTransmissionContingency**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None

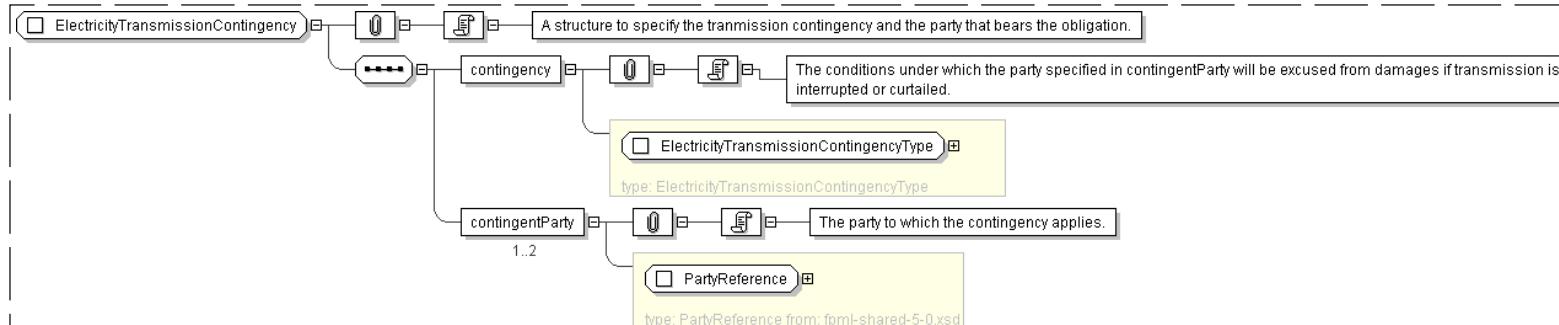
<b>Name</b>	ElectricityTransmissionContingency
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">ElectricityDelivery</a>
<b>Abstract</b>	no
<b>Documentation</b>	A structure to specify the transmission contingency and the party that bears the obligation.

**XML Instance Representation**

```
<...>
<contingency> ElectricityTransmissionContingencyType </contingency> [1]
'The conditions under which the party specified in contingentParty will be excused from damages if transmission is interrupted or curtailed.'

<contingentParty> PartyReference </contingentParty> [1..2]
'The party to which the contingency applies.'

</...>
```

**Diagram****Schema Component Representation**

```
<xsd:complexType name="ElectricityTransmissionContingency">
  <xsd:sequence>
    <xsd:element name="contingency" type="#ElectricityTransmissionContingencyType" />
    <xsd:element name="contingentParty" type="#PartyReference" minOccurs="1" maxOccurs="2" />
  </xsd:sequence>
</xsd:complexType>
```

```

<xsd:element name="contingentParty" type="#PartyReference" maxOccurs="2"/>
</xsd:sequence>
</xsd:complexType>

```

## Complex Type: ElectricityTransmissionContingencyType

Super-types:	<a href="#">Scheme</a> < ElectricityTransmissionContingencyType (by extension)
Sub-types:	None

Name	ElectricityTransmissionContingencyType
Used by (from the same schema document)	Complex Type <a href="#">ElectricityTransmissionContingency</a>
Abstract	no
Documentation	The type of transmission contingency, i.e. what portion of the transmission the delivery obligations are applicable.

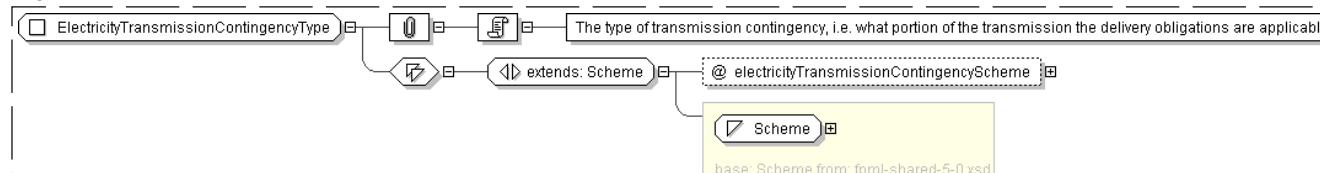
### XML Instance Representation

```

<...
  electricityTransmissionContingencyScheme=" xsd:anyURI [0..1]">
  Scheme
</...>

```

### Diagram



### Schema Component Representation

```

<xsd:complexType name="ElectricityTransmissionContingencyType">
  <xsd:simpleContent>
    <xsd:extension base="#Scheme">
      <xsd:attribute name="electricityTransmissionContingencyScheme" type="xsd:anyURI"
        " default="http://www.fpml.org/coding-scheme/electricity-transmission-contingency" />
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>

```

## Complex Type: FixedPrice

Super-types:	None
Sub-types:	• <a href="#">SettlementPeriodsFixedPrice</a> (by extension)

Name	FixedPrice
Used by (from the same schema document)	Complex Type <a href="#">CommodityFixedPriceSchedule</a> , Complex Type <a href="#">CommoditySettlementPeriodsPriceSchedule</a> , Complex Type <a href="#">NonPeriodicFixedPriceLeg</a> , Model Group <a href="#">CommodityFixedPrice.model</a>
Abstract	no
Documentation	A type defining the Fixed Price.

### XML Instance Representation

```

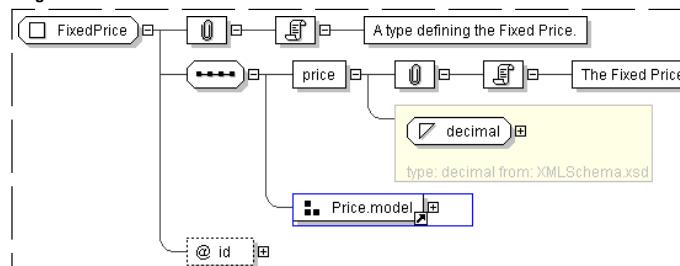
<...
  id="#xsd:ID [0..1]">
  <price> xsd:decimal </price> [1]
  'The Fixed Price.'

  <priceCurrency> Currency </priceCurrency> [1]
  'Currency of the fixed price.'

  <priceUnit> QuantityUnit </priceUnit> [1]
  'The unit of measure used to calculate the Fixed Price.'

```

&lt;...&gt;

**Diagram****Schema Component Representation**

```

<xsd:complexType name="FixedPrice">
  <xsd:sequence>
    <xsd:element name="price" type="xsd:decimal" />
    <xsd:group ref=" Price_model " />
  </xsd:sequence>
  <xsd:attribute name="id" type="xsd:ID" />
</xsd:complexType>
  
```

top

**Complex Type: FixedPriceLeg**

<b>Super-types:</b>	<a href="#">Leg</a> < <b>FixedPriceLeg</b> (by extension)
<b>Sub-types:</b>	None
<b>Name</b>	FixedPriceLeg
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">CommoditySwap</a>
<b>Abstract</b>	no
<b>Documentation</b>	Fixed Price Leg of a Commodity Swap.

**XML Instance Representation**

```

<...
  id=" xsd:ID [0..1]">
  <payerPartyReference> PartyReference </payerPartyReference> [1]
  'A reference to the party responsible for making the payments defined by this structure.'

  <payerAccountReference> AccountReference </payerAccountReference> [0..1]
  'A reference to the account responsible for making the payments defined by this structure.'

  <receiverPartyReference> PartyReference </receiverPartyReference> [1]
  'A reference to the party that receives the payments corresponding to this structure.'

  <receiverAccountReference> AccountReference </receiverAccountReference> [0..1]
  'A reference to the account that receives the payments corresponding to this structure.'

Start Choice [1]
  <calculationDates> AdjustableDates </calculationDates> [1]
  'The Calculation Period dates for this leg of the trade where the Calculation Periods are all one day long, typically a physically-settled emissions or metals trade. Only dates explicitly included determine the Calculation Periods and there is a Calculation Period for each date specified.'

  <calculationPeriods> AdjustableDates </calculationPeriods> [1]
  'The Calculation Period start dates for this leg of the swap. This type is only intended to be used if the Calculation Periods differ on each leg. If Calculation Periods mirror another leg, then the calculationPeriodsReference element should be used to point to the Calculation Periods on that leg - or the calculationPeriodsScheduleReference can be used to point to the Calculation Periods Schedule for that leg.'

  <calculationPeriodsSchedule> CommodityCalculationPeriodsSchedule </
  
```

`calculationPeriodsSchedule> [1]`

'The Calculation Periods for this leg of the swap. This type is only intended to be used if the Calculation Periods differ on each leg. If Calculation Periods mirror another leg, then the calculationPeriodsReference element should be used to point to the Calculation Periods on the other leg - or the calculationPeriodsScheduleReference can be used to point to the Calculation Periods Schedule for that leg.'

Start Choice [1]

`<calculationPeriodsReference> CalculationPeriodsReference </calculationPeriodsReference> [1]`

'A pointer style reference to the Calculation Periods defined on another leg.'

`<calculationPeriodsScheduleReference> CalculationPeriodsScheduleReference`

`</calculationPeriodsScheduleReference> [1]`

'A pointer style reference to the Calculation Periods Schedule defined on another leg.'

`<calculationPeriodsDatesReference> CalculationPeriodsDatesReference`

`</calculationPeriodsDatesReference> [1]`

'A pointer style reference to single-day-duration Calculation Periods defined on another leg.'

End Choice

End Choice

Start Choice [1]

`<fixedPriceSchedule> CommodityFixedPriceSchedule </fixedPriceSchedule> [1]`

'Allows the specification of a Fixed Price that varies over the life of the trade.'

Start Choice [1]

`<fixedPrice> FixedPrice </fixedPrice> [1]`

'Fixed price on which fixed payments are based.'

`<worldscaleRate> xsd:decimal </worldscaleRate> [1]`

'For a NET Voyager Charter Commodity Swap, the number of Worldscale Points for purposes of the calculation of a Fixed Amount.'

`<contractRate> NonNegativeMoney </contractRate> [1]`

'For a DRY Voyage Charter or Time Charter Commodity Swap, the price per relevant unit for purposes of the calculation of a Fixed Amount.'

`<settlementPeriodsPrice> SettlementPeriodsFixedPrice </settlementPeriodsPrice> [1..*]`

'For an electricity transaction, the fixed price for one or more groups of Settlement Periods on which fixed payments are based. If the fixed price differs for different groups of Settlement Periods, this element should be repeated.'

End Choice

End Choice

`<totalPrice> NonNegativeMoney </totalPrice> [0..1]`

'The total amount of all fixed payments due during the term of the trade.'

Start Choice [1]

Start Choice [1]

`<notionalQuantitySchedule> CommodityNotionalQuantitySchedule </notionalQuantitySchedule> [1]`

'Allows the documentation of a shaped notional trade where the notional changes over the life of the transaction.'

`<notionalQuantity> CommodityNotionalQuantity </notionalQuantity> [1]`

'The Notional Quantity.'

`<settlementPeriodsNotionalQuantity> CommoditySettlementPeriodsNotionalQuantity`

`</settlementPeriodsNotionalQuantity> [1..*]`

'For an electricity transaction, the Notional Quantity for a one or more groups of Settlement Periods to which the Notional Quantity is based. If the schedule differs for different groups of Settlement Periods, this element should be repeated.'

End Choice

`<totalNotionalQuantity> xsd:decimal </totalNotionalQuantity> [0..1]`

'The Total Notional Quantity.'

`<quantityReference> QuantityReference </quantityReference> [1]`

'A pointer style reference to a quantity defined on another leg.'

End Choice

Start Choice [1]

`<relativePaymentDates> CommodityRelativePaymentDates </relativePaymentDates> [1]`

'The Payment Dates of the trade relative to the Calculation Periods.'

Start Choice [1]  
 <paymentDates> AdjustableDatesOrRelativeDateOffset </paymentDates> [1]

'Dates on which payments will be made.'

<masterAgreementPaymentDates> xsd:boolean </masterAgreementPaymentDates> [1]

'If present and true indicates that the Payment Date(s) are specified in the relevant master agreement.'

End Choice

End Choice

Start Group: CommodityFreightFlatRate.model [0..1]

<flatRate> FlatRateEnum </flatRate> [1]

'Whether the Flat Rate is the New Worldwide Tanker Nominal Freight Scale for the Freight Index Route taken at the Trade Date of the transaction or taken on each Pricing Date.'

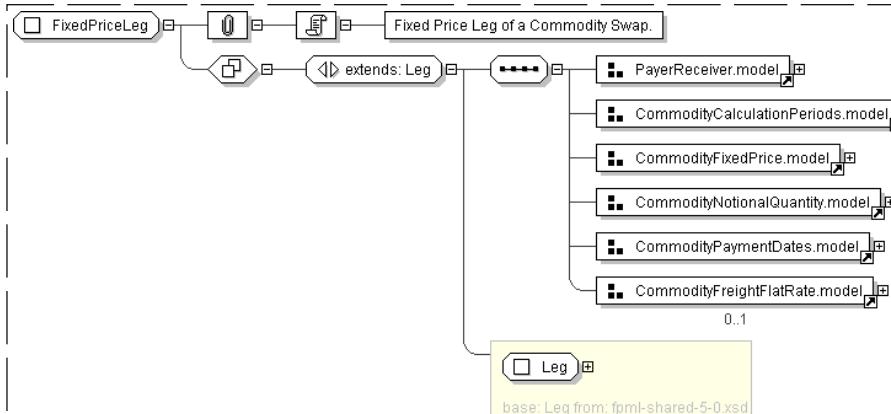
<flatRateAmount> NonNegativeMoney </flatRateAmount> [0..1]

'If flatRate is set to \"Fixed\", the actual value of the Flat Rate.'

End Group: CommodityFreightFlatRate.model

</...>

#### Diagram



#### Schema Component Representation

```

<xsd:complexType name="FixedPriceLeg">
  <xsd:complexContent>
    <xsd:extension base=" Leg ">
      <xsd:sequence>
        <xsd:group ref=" PayerReceiver.model "/>
        <xsd:group ref=" CommodityCalculationPeriods.model "/>
        <xsd:group ref=" CommodityFixedPrice.model "/>
        <xsd:group ref=" CommodityNotionalQuantity.model "/>
        <xsd:group ref=" CommodityPaymentDates.model "/>
        <xsd:group ref=" CommodityFreightFlatRate.model " minOccurs="0"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
  
```

top

#### Complex Type: FloatingLegCalculation

Super-types:

None

Sub-types:

None

Name

FloatingLegCalculation

Used by (from the same schema document)

Complex Type [FloatingPriceLeg](#)

**Abstract**

no

**Documentation**

A type to capture details relevant to the calculation of the floating price.

**XML Instance Representation**

```
<...>
<pricingDates> CommodityPricingDates </pricingDates> [1]
'Commodity Pricing Dates.'

<averagingMethod> AveragingMethodEnum </averagingMethod> [0..1]
'The parties may specify a Method of Averaging where more than one pricing Dates is
being specified as being applicable.'

<conversionFactor> xsd:decimal </conversionFactor> [0..1]
'If the Notional Quantity is specified in a unit that does not match the unit in which
the Commodity Reference Price is quoted, the scaling or conversion factor used to convert
the Commodity Reference Price unit into the Notional Quantity unit should be stated here.
If there is no conversion, this element is not intended to be used.'

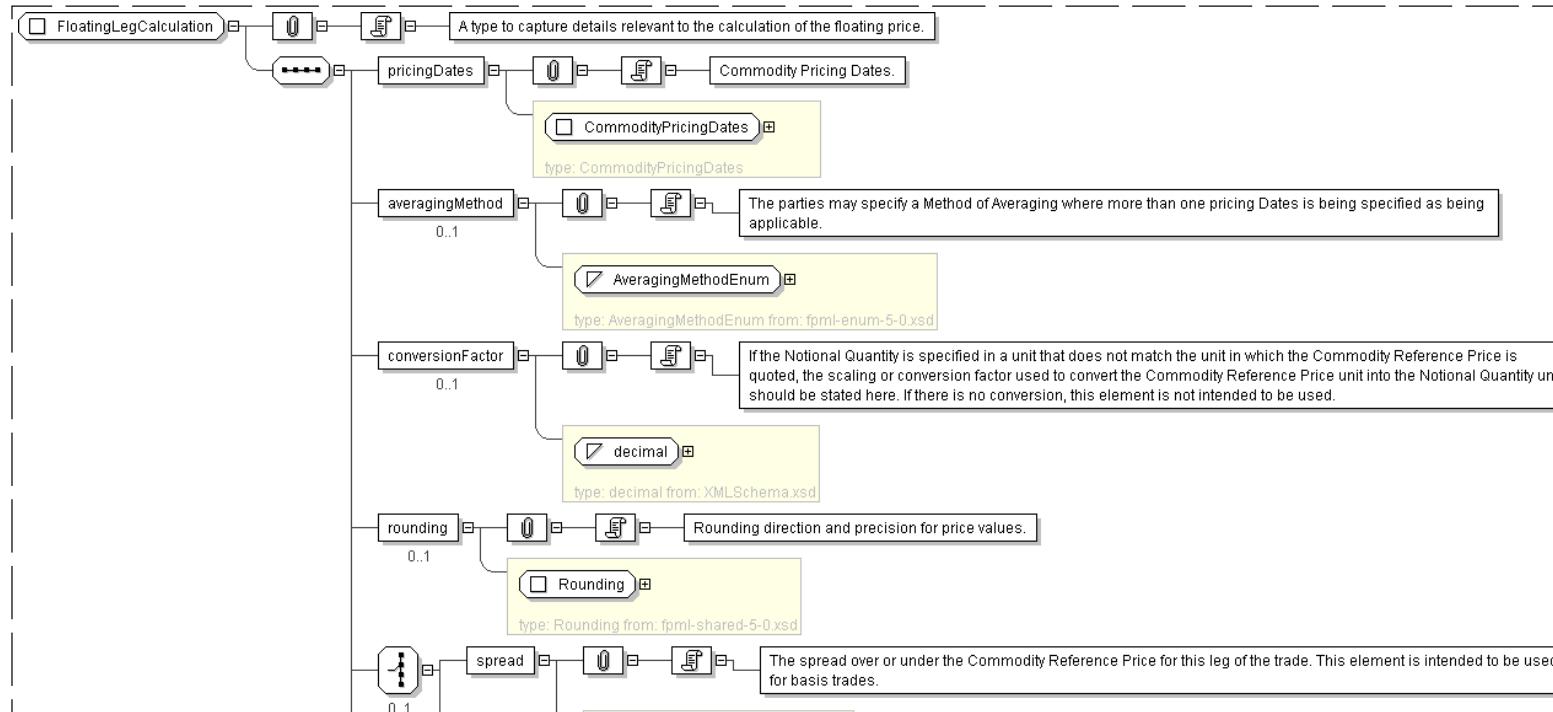
<rounding> Rounding </rounding> [0..1]
'Rounding direction and precision for price values.'

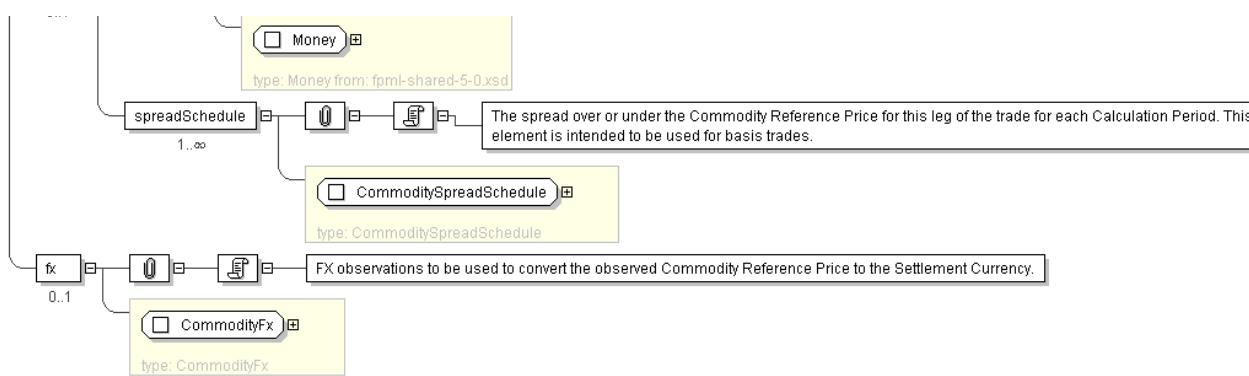
Start Choice [0..1]
<spread> Money </spread> [1]
'The spread over or under the Commodity Reference Price for this leg of the trade. This
element is intended to be used for basis trades.'

<spreadSchedule> CommoditySpreadSchedule </spreadSchedule> [1..*]
'The spread over or under the Commodity Reference Price for this leg of the trade for
each Calculation Period. This element is intended to be used for basis trades.'

End Choice
<fx> CommodityFx </fx> [0..1]
'FX observations to be used to convert the observed Commodity Reference Price to the
Settlement Currency.'

</...>
```

**Diagram**

**Schema Component Representation**

```

<xsd:complexType name="FloatingLegCalculation">
    <xsd:sequence>
        <xsd:element name="pricingDates" type="CommodityPricingDates" />
        <xsd:element name="averagingMethod" type="AveragingMethodEnum" minOccurs="0" />
        <xsd:element name="conversionFactor" type="xsd:decimal" minOccurs="0" />
        <xsd:element name="rounding" type="Rounding" minOccurs="0" />
        <xsd:choice minOccurs="0">
            <xsd:element name="spread" type="Money" />
            <xsd:element name="spreadSchedule" type="CommoditySpreadschedule" maxOccurs="unbounded" />
        </xs:choice>
        <xsd:element name="fx" type="CommodityFx" minOccurs="0" />
    </xs:sequence>
</xsd:complexType>

```

top

**Complex Type: FloatingPriceLeg**

<b>Super-types:</b>	<a href="#">Leg</a> < <b>FloatingPriceLeg</b> (by extension)
<b>Sub-types:</b>	None

<b>Name</b>	FloatingPriceLeg
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">CommoditySwap</a>
<b>Abstract</b>	no
<b>Documentation</b>	Floating Price Leg of a Commodity Swap.

**XML Instance Representation**

```

<...
  id="xsd:ID [0..1]">
  <payerPartyReference> PartyReference </payerPartyReference> [1]
  'A reference to the party responsible for making the payments defined by this structure.'

  <payerAccountReference> AccountReference </payerAccountReference> [0..1]
  'A reference to the account responsible for making the payments defined by this structure.'

  <receiverPartyReference> PartyReference </receiverPartyReference> [1]
  'A reference to the party that receives the payments corresponding to this structure.'

  <receiverAccountReference> AccountReference </receiverAccountReference> [0..1]
  'A reference to the account that receives the payments corresponding to this structure.'

Start Choice [1]
  <calculationDates> AdjustableDates </calculationDates> [1]
  'The Calculation Period dates for this leg of the trade where the Calculation Periods are all one day long, typically a physically-settled emissions or metals trade. Only dates explicitly included determine the Calculation Periods and there is a Calculation Period for each date specified.'

  <calculationPeriods> AdjustableDates </calculationPeriods> [1]
  'The Calculation Period start dates for this leg of the swap. This type is only intended to
  ...

```

be used if the Calculation Periods differ on each leg. If Calculation Periods mirror another leg, then the calculationPeriodsReference element should be used to point to the Calculation Periods on that leg - or the calculationPeriodsScheduleReference can be used to point to the Calculation Periods Schedule for that leg.'

```
<calculationPeriodsSchedule> CommodityCalculationPeriodsSchedule </calculationPeriodsSchedule> [1]
```

'The Calculation Periods for this leg of the swap. This type is only intended to be used if the Calculation Periods differ on each leg. If Calculation Periods mirror another leg, then the calculationPeriodsReference element should be used to point to the Calculation Periods on the other leg - or the calculationPeriodsScheduleReference can be used to point to the Calculation Periods Schedule for that leg.'

Start Choice [1]

```
<calculationPeriodsReference> CalculationPeriodsReference </calculationPeriodsReference> [1]
```

'A pointer style reference to the Calculation Periods defined on another leg.'

```
<calculationPeriodsScheduleReference> CalculationPeriodsScheduleReference </calculationPeriodsScheduleReference> [1]
```

'A pointer style reference to the Calculation Periods Schedule defined on another leg.'

```
<calculationPeriodsDatesReference> CalculationPeriodsDatesReference </calculationPeriodsDatesReference> [1]
```

'A pointer style reference to single-day-duration Calculation Periods defined on another leg.'

End Choice

End Choice

```
<commodity> Commodity </commodity> [1]
```

'Specifies the underlying instrument. At this time, only underlyers of type Commodity are supported; the choice group in the future could offer the possibility of adding other types later.'

Start Choice [1]

Start Choice [1]

```
<notionalQuantitySchedule> CommodityNotionalQuantitySchedule </notionalQuantitySchedule> [1]
```

'Allows the documentation of a shaped notional trade where the notional changes over the life of the transaction.'

```
<notionalQuantity> CommodityNotionalQuantity </notionalQuantity> [1]
```

'The Notional Quantity.'

```
<settlementPeriodsNotionalQuantity> CommoditySettlementPeriodsNotionalQuantity </settlementPeriodsNotionalQuantity> [1..*]
```

'For an electricity transaction, the Notional Quantity for a one or more groups of Settlement Periods to which the Notional Quantity is based. If the schedule differs for different groups of Settlement Periods, this element should be repeated.'

End Choice

```
<totalNotionalQuantity> xsd:decimal </totalNotionalQuantity> [0..1]
```

'The Total Notional Quantity.'

```
<quantityReference> QuantityReference </quantityReference> [1]
```

'A pointer style reference to a quantity defined on another leg.'

End Choice

```
<calculation> FloatingLegCalculation </calculation> [1]
```

'Defines details relevant to the calculation of the floating price.'

Start Choice [1]

```
<relativePaymentDates> CommodityRelativePaymentDates </relativePaymentDates> [1]
```

'The Payment Dates of the trade relative to the Calculation Periods.'

Start Choice [1]

```
<paymentDates> AdjustableDatesOrRelativeDateOffset </paymentDates> [1]
```

'Dates on which payments will be made.'

```
<masterAgreementPaymentDates> xsd:boolean </masterAgreementPaymentDates> [1]
```

'If present and true indicates that the Payment Date(s) are specified in the relevant master agreement.'

End Choice

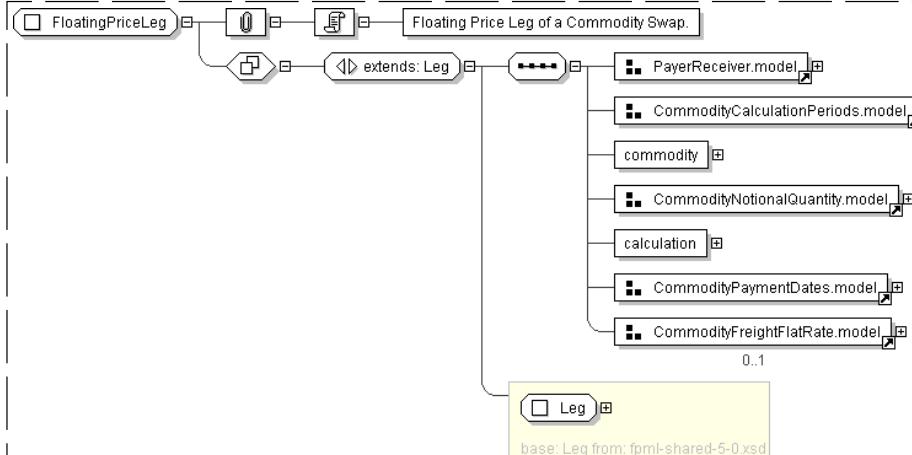
End Choice

Start Group: CommodityFreightFlatRate.model [0..1]

```
<flatRate> FlatRateEnum </flatRate> [1]
'Whether the Flat Rate is the New Worldwide Tanker Nominal Freight Scale for the Freight
Index Route taken at the Trade Date of the transaction or taken on each Pricing Date.'

<flatRateAmount> NonNegativeMoney </flatRateAmount> [0..1]
'If flatRate is set to \"Fixed\", the actual value of the Flat Rate.'
```

End Group: [CommodityFreightFlatRate.model](#)

**Diagram****Schema Component Representation**

```

<xsd:complexType name="FloatingPriceLeg">
  <xsd:complexContent>
    <xsd:extension base=" Leg ">
      <xsd:sequence>
        <xsd:group ref=" PayerReceiver.model ">
        <xsd:group ref=" CommodityCalculationPeriods.model ">
        <xsd:element name="commodity" type=" Commodity ">
        <xsd:group ref=" CommodityNotionalQuantity.model ">
        <xsd:element name="calculation" type=" FloatingLegCalculation ">
        <xsd:group ref=" CommodityPaymentDates.model ">
        <xsd:group ref=" CommodityFreightFlatRate.model " minOccurs="0"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

top

**Complex Type: GasDelivery**

**Super-types:** None  
**Sub-types:** None

<b>Name</b>	GasDelivery
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">GasPhysicalLeg</a>
<b>Abstract</b>	no
<b>Documentation</b>	The specification of the gas to be delivered.

**XML Instance Representation**

```
<...>
Start Choice [1]
<deliveryPoint> GasDeliveryPoint </deliveryPoint> [1]
'The physical or virtual point at which the commodity will be delivered.'
```

```

<entryPoint> CommodityDeliveryPoint </entryPoint> [1]
'The physical or virtual point at which the commodity enters a transportation system.'

<withdrawalPoint> CommodityDeliveryPoint </withdrawalPoint> [1]
'The physical or virtual point at which the commodity is withdrawn from a
transportation system.'

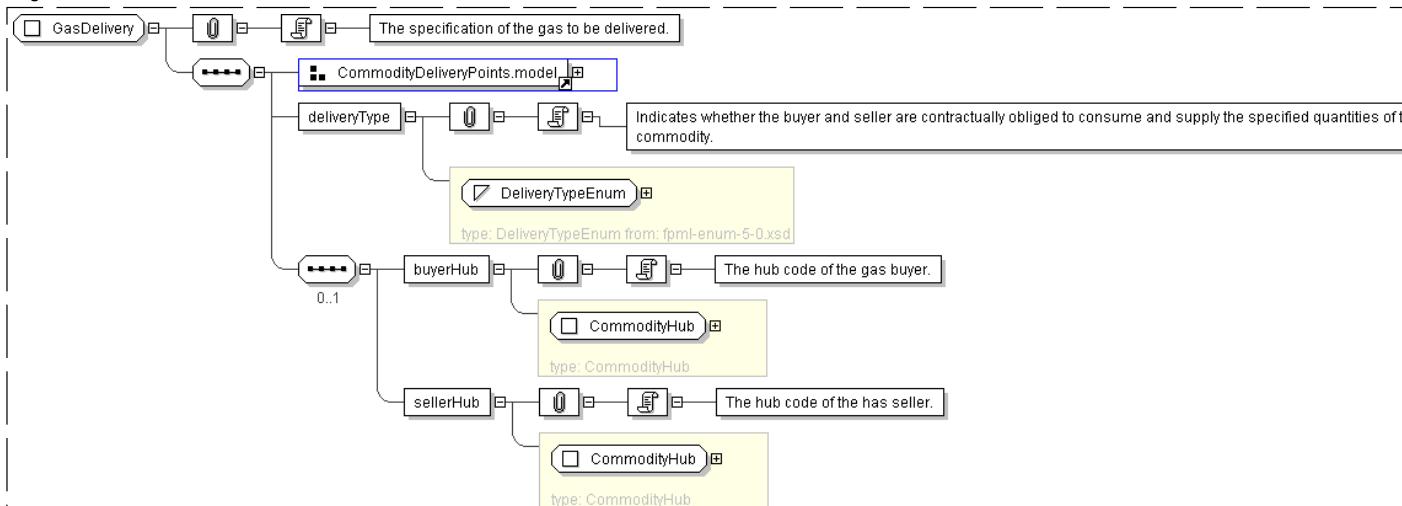
End Choice
<deliveryType> DeliveryTypeEnum </deliveryType> [1]
'Indicates whether the buyer and seller are contractually obliged to consume and supply
the specified quantities of the commodity.'

Start Sequence [0..1]
<buyerHub> CommodityHub </buyerHub> [1]
'The hub code of the gas buyer.'

<sellerHub> CommodityHub </sellerHub> [1]
'The hub code of the has seller.'

End Sequence
</...>

```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="GasDelivery">
  <xsd:sequence>
    <xsd:group ref="#CommodityDeliveryPoints.model" />
    <xsd:element name="deliveryType" type="#DeliveryTypeEnum" />
    <xsd:sequence minOccurs="0">
      <xsd:element name="buyerHub" type="#CommodityHub" />
      <xsd:element name="sellerHub" type="#CommodityHub" />
    </xsd:sequence>
  </xsd:sequence>
</xsd:complexType>

```

top

**Complex Type: GasDeliveryPeriods**

Super-types:	CommodityDeliveryPeriods < GasDeliveryPeriods (by extension)
--------------	--

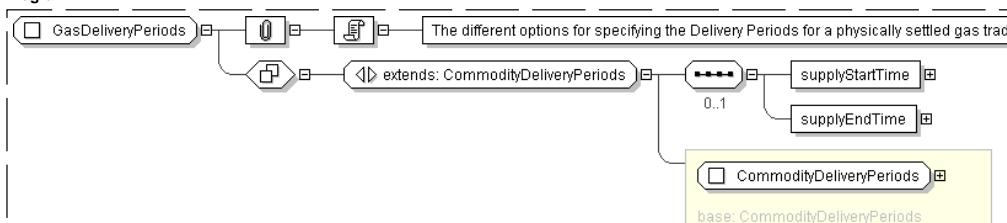
Sub-types:	None
------------	------

Name	GasDeliveryPeriods
Used by (from the same schema document)	Complex Type GasPhysicalLeg

<b>Abstract</b>	no
<b>Documentation</b>	The different options for specifying the Delivery Periods for a physically settled gas trade.

**XML Instance Representation**

```
<...>
  id=" xsd:ID [0..1]">
  Start Choice [1]
    <periods> AdjustableDates </periods> [1]
      'The Delivery Periods for this leg of the swap. This type is only intended to be used if the Delivery Periods differ from the Calculation Periods on the fixed or floating leg. If DeliveryPeriods mirror another leg, then the calculationPeriodsReference element should be used to point to the Calculation Periods on that leg - or the calculationPeriodsScheduleReference can be used to point to the Calculation Periods Schedule for that leg.'
    <periodsSchedule> CommodityCalculationPeriodsSchedule </periodsSchedule> [1]
      'The Delivery Periods for this leg of the swap. This type is only intended to be used if the Delivery Periods differ from the Calculation Periods on the fixed or floating leg. If DeliveryPeriods mirror another leg, then the calculationPeriodsReference element should be used to point to the Calculation Periods on that leg - or the calculationPeriodsScheduleReference can be used to point to the Calculation Periods Schedule for that leg.'
    Start Choice [1]
      <calculationPeriodsReference> CalculationPeriodsReference </calculationPeriodsReference> [1]
        'A pointer style reference to the Calculation Periods defined on another leg.'
      <calculationPeriodsScheduleReference> CalculationPeriodsScheduleReference </calculationPeriodsScheduleReference> [1]
        'A pointer style reference to the Calculation Periods Schedule defined on another leg.'
      <calculationPeriodsDatesReference> CalculationPeriodsDatesReference </calculationPeriodsDatesReference> [1]
        'A pointer style reference to single-day-duration Calculation Periods defined on another leg.'
    End Choice
    End Choice
  Start Sequence [0..1]
    <supplyStartTime> PrevailingTime </supplyStartTime> [1]
    'The time at which gas delivery should start on each day of the Delivery Period(s).'
    <supplyEndTime> PrevailingTime </supplyEndTime> [1]
    'The time at which gas delivery should end on each day of the Delivery Period(s).'
  End Sequence
<...>
```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="GasDeliveryPeriods">
  <xsd:complexContent>
    <xsd:extension base=" CommodityDeliveryPeriods ">
      <xsd:sequence minOccurs="0">
        <xsd:element name="supplyStartTime" type=" PrevailingTime "/>
        <xsd:element name="supplyEndTime" type=" PrevailingTime "/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
  
```

## Complex Type: GasDeliveryPoint

Super-types: [Scheme](#) < **GasDeliveryPoint** (by extension)

Sub-types: None

Name **GasDeliveryPoint**

Used by (from the same schema document) Model Group [CommodityDeliveryPoints.model](#)

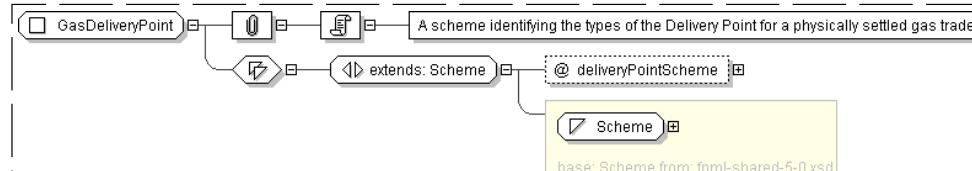
Abstract no

Documentation A scheme identifying the types of the Delivery Point for a physically settled gas trade.

### XML Instance Representation

```
<...  
deliveryPointScheme=" xsd:anyURI [0..1]">  
Scheme  
</...>
```

### Diagram



### Schema Component Representation

```
<xsd:complexType name="GasDeliveryPoint">  
  <xsd:simpleContent>  
    <xsd:extension base=" Scheme ">  
      <xsd:attribute name="deliveryPointScheme" type=" xsd:anyURI " />  
    </xsd:extension>  
  </xsd:simpleContent>  
</xsd:complexType>
```

## Complex Type: GasPhysicalLeg

Super-types: [Leg](#) < [PhysicalLeg](#) (by extension) < **GasPhysicalLeg** (by extension)

Sub-types: None

Name **GasPhysicalLeg**

Used by (from the same schema document) Complex Type [CommoditySwap](#)

Abstract no

Documentation Physically settled leg of a physically settled gas transaction.

### XML Instance Representation

```
<...  
id=" xsd:ID [0..1]">  
  <payerPartyReference> PartyReference </payerPartyReference> [1]  
  'A reference to the party responsible for making the payments defined by this structure.'  
  
  <payerAccountReference> AccountReference </payerAccountReference> [0..1]  
  'A reference to the account responsible for making the payments defined by this structure.'  
  
  <receiverPartyReference> PartyReference </receiverPartyReference> [1]  
  'A reference to the party that receives the payments corresponding to this structure.'  
  
  <receiverAccountReference> AccountReference </receiverAccountReference> [0..1]  
  'A reference to the account that receives the payments corresponding to this structure.'  
  
  <deliveryPeriods> GasDeliveryPeriods </deliveryPeriods> [1]  
  'The different options for specifying the Delivery or Supply Periods. Unless the quantity
```

*or price is to vary periodically during the trade or physical delivery occurs on a periodic basis, periodsSchedule should be used and set to IT.'*

<gas> `GasProduct` </gas> [1]

'The specification of the gas to be delivered.'

<deliveryConditions> `GasDelivery` </deliveryConditions> [1]

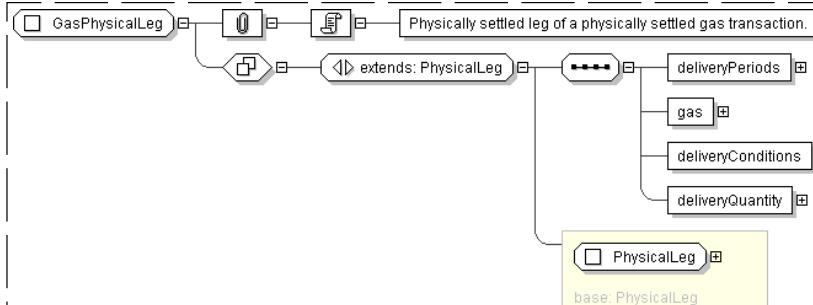
'The physical delivery conditions for the transaction.'

<deliveryQuantity> `GasPhysicalQuantity` </deliveryQuantity> [1]

'The different options for specifying the quantity. For Fixed trades where the quantity is known at the time of confirmation, a single quantity or a quantity per Delivery Period may be specified. For Variable trades minimum and maximum trades may be specified.'

<...>

#### Diagram



#### Schema Component Representation

```

<xsd:complexType name="GasPhysicalLeg">
  <xsd:complexContent>
    <xsd:extension base=" PhysicalLeg ">
      <xsd:sequence>
        <xsd:element name="deliveryPeriods" type=" GasDeliveryPeriods "/>
        <xsd:element name="gas" type=" GasProduct "/>
        <xsd:element name="deliveryConditions" type=" GasDelivery "/>
        <xsd:element name="deliveryQuantity" type=" GasPhysicalQuantity "/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

[top](#)

#### Complex Type: `GasPhysicalQuantity`

Super-types: `CommodityPhysicalQuantityBase` < `GasPhysicalQuantity` (by extension)

Sub-types: None

Name `GasPhysicalQuantity`

Used by (from the same schema document) Complex Type `GasPhysicalLeg`

Abstract no

Documentation The quantity of gas to be delivered.

#### XML Instance Representation

```

<...>
<id=" xsd:ID [0..1]">
Start Choice [1]
Start Choice [1]
  <physicalQuantity> CommodityNotionalQuantity </physicalQuantity> [1]
  'The Quantity per Delivery Period.'

  <physicalQuantitySchedule> CommodityPhysicalQuantitySchedule </physicalQuantitySchedule> [1]
  'Allows the documentation of a shaped quantity trade where the quantity changes over the

```

*life of the transaction.'*

End Choice  
 <totalPhysicalQuantity> UnitQuantity </totalPhysicalQuantity> [0..1]

'The Total Quantity of the commodity to be delivered.'

<minPhysicalQuantity> CommodityNotionalQuantity </minPhysicalQuantity> [1..\*]

'The minimum quantity to be delivered. If separate minimums need to be specified for different periods (e.g. a minimum per day and a minimum per month) this element should be repeated.'

<maxPhysicalQuantity> CommodityNotionalQuantity </maxPhysicalQuantity> [1..\*]

'The maximum quantity to be delivered. If separate maximums need to be specified for different periods (e.g. a minimum per day and a minimum per month) this element should be repeated.'

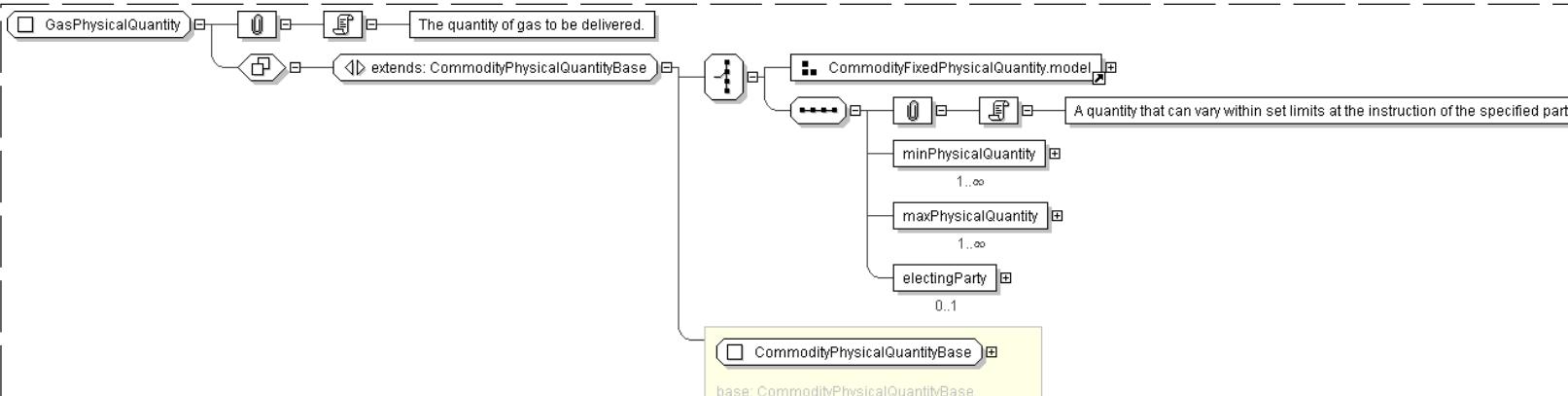
<electingParty> PartyReference </electingParty> [0..1]

'Indicates the party able to choose whether the gas is delivered for a particular period e.g. a swing or interruptible contract.'

End Choice

</...>

#### Diagram



#### Schema Component Representation

```

<xsd:complexType name="GasPhysicalQuantity">
  <xsd:complexContent>
    <xsd:extension base=" CommodityPhysicalQuantityBase ">
      <xsd:choice>
        <xsd:group ref=" CommodityFixedPhysicalQuantity.model ">
          <xsd:sequence>
            <xsd:element name="minPhysicalQuantity" type=" CommodityNotionalQuantity "
              " maxOccurs="unbounded"/>
            <xsd:element name="maxPhysicalQuantity" type=" CommodityNotionalQuantity "
              " maxOccurs="unbounded"/>
            <xsd:element name="electingParty" type=" PartyReference " minOccurs="0"/>
          </xsd:sequence>
        </xsd:choice>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>

```

top

#### Complex Type: GasProduct

Super-types:	None
Sub-types:	None

Name	GasProduct
------	------------

<b>Used by (from the same schema document)</b>	Complex Type <a href="#">GasPhysicalLeg</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type defining the characteristics of the gas being traded in a physically settled gas transaction.

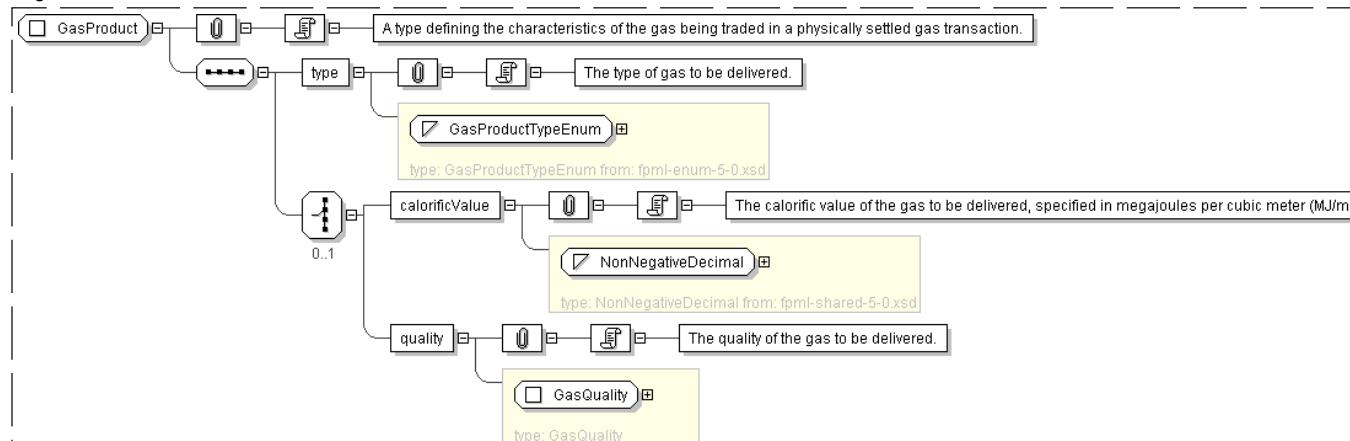
**XML Instance Representation**

```
<...>
<type> GasProductTypeEnum </type> [1]
'The type of gas to be delivered.'

Start Choice [0..1]
<calorificValue> NonNegativeDecimal </calorificValue> [1]
'The calorific value of the gas to be delivered, specified in megajoules per cubic meter
(MJ/m³).'

<quality> GasQuality </quality> [1]
'The quality of the gas to be delivered.'

End Choice
</...>
```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="GasProduct">
  <xsd:sequence>
    <xsd:element name="type" type="GasProductTypeEnum" />
    <xsd:choice minOccurs="0">
      <xsd:element name="calorificValue" type="NonNegativeDecimal" />
      <xsd:element name="quality" type="GasQuality" />
    </xsd:choice>
  </xsd:sequence>
</xsd:complexType>
  
```

[top](#)**Complex Type: [GasQuality](#)**

<b>Super-types:</b>	<a href="#">Scheme</a> < <a href="#">GasQuality</a> (by extension)
<b>Sub-types:</b>	None

<b>Name</b>	GasQuality
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">GasProduct</a>
<b>Abstract</b>	no
<b>Documentation</b>	The quantity of gas to be delivered.

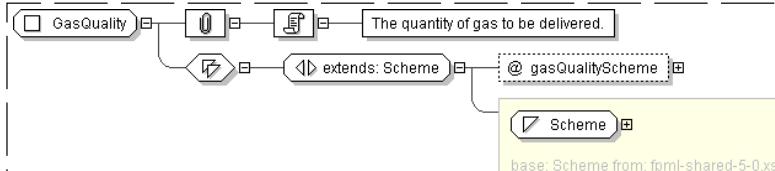
**XML Instance Representation**

```
<...>
```

```
| gasQualityScheme=" xsd:anyURI [0..1]">
```

```
| Scheme
```

```
|</...>
```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="GasQuality">
  <xsd:simpleContent>
    <xsd:extension base=" Scheme ">
      <xsd:attribute name="gasQualityScheme" type=" xsd:anyURI " default="http://www.fpml.org/
        coding-scheme/commodity-gas-quality"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
  
```

top

**Complex Type: Lag**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None

<b>Name</b>	Lag
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">CommodityPricingDates</a> , Model Group <a href="#">LagOrReference.model</a>
<b>Abstract</b>	no
<b>Documentation</b>	An observation period that is offset from a Calculation Period.

**XML Instance Representation**

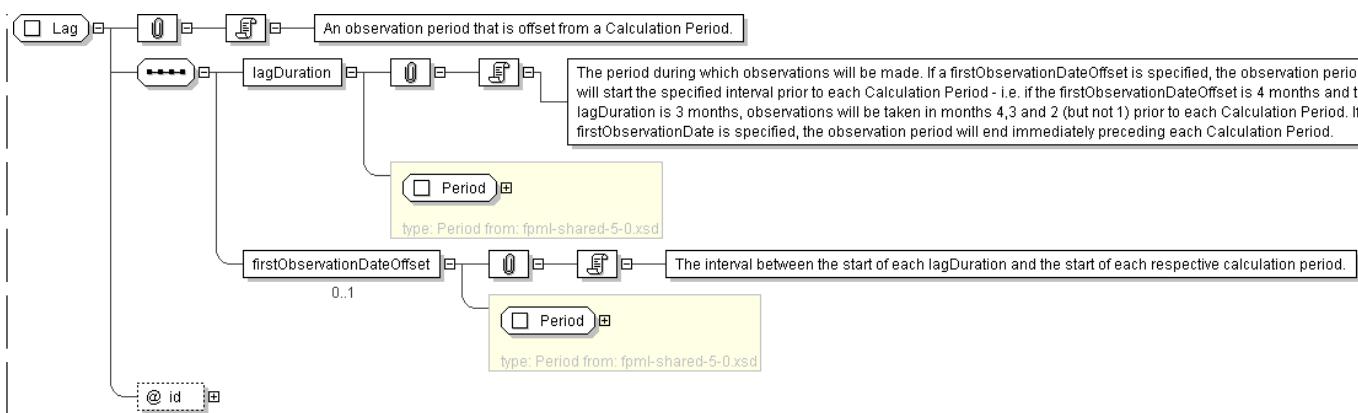
```

<...
  id=" xsd:ID [0..1]">
    <lagDuration> Period </lagDuration> [1]

    'The period during which observations will be made. If a firstObservationDateOffset
    is specified, the observation period will start the specified interval prior to
    each Calculation Period - i.e. if the firstObservationDateOffset is 4 months and
    the lagDuration is 3 months, observations will be taken in months 4,3 and 2 (but not 1)
    prior to each Calculation Period. If no firstObservationDate is specified, the
    observation period will end immediately preceding each Calculation Period.'

    <firstObservationDateOffset> Period </firstObservationDateOffset> [0..1]
    'The interval between the start of each lagDuration and the start of each
    respective calculation period.'
  </...>
  
```

**Diagram**

**Schema Component Representation**

```
<xsd:complexType name="Lag">
  <xsd:sequence>
    <xsd:element name="lagDuration" type=" Period " />
    <xsd:element name="firstObservationDateOffset" type=" Period " minOccurs="0"/>
  </xsd:sequence>
  <xsd:attribute name="id" type=" xsd:ID ">
</xsd:complexType>
```

top

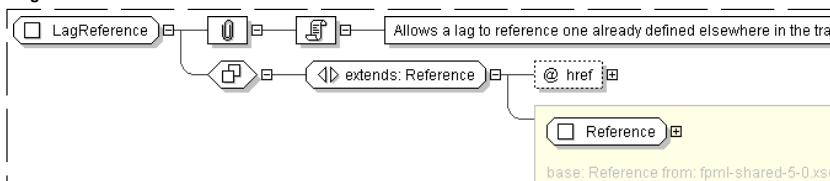
**Complex Type: LagReference**

Super-types:	<a href="#">Reference</a> < LagReference (by extension)
Sub-types:	None

Name	LagReference
Used by (from the same schema document)	Model Group <a href="#">LagOrReference.model</a>
Abstract	no
Documentation	Allows a lag to reference one already defined elsewhere in the trade.

**XML Instance Representation**

```
<...>
  <@ href=" xsd:IDREF [1]" />
```

**Diagram****Schema Component Representation**

```
<xsd:complexType name="LagReference">
  <xsd:complexContent>
    <xsd:extension base=" Reference ">
      <xsd:attribute name="href" type=" xsd:IDREF " use="required" reference="Lag"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

top

**Complex Type: MarketDisruptionEvent**

<b>Super-types:</b>	<a href="#">Scheme</a> < <b>MarketDisruptionEvent</b> (by extension)
<b>Sub-types:</b>	None
<b>Name</b>	MarketDisruptionEvent
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">CommodityMarketDisruption</a> , Complex Type <a href="#">CommodityMarketDisruption</a>
<b>Abstract</b>	no
<b>Documentation</b>	A Market Disruption Event.

**XML Instance Representation**

```
<...  
commodityMarketDisruptionScheme=" xsd:anyURI [0..1]">  
Scheme  
<...>
```

**Diagram**

```

classDiagram
    class MarketDisruptionEvent
    class Scheme
    MarketDisruptionEvent "2" -- "1" Scheme : @ commodityMarketDisruptionScheme
    
```

**Schema Component Representation**

```
<xsd:complexType name="MarketDisruptionEvent">  
  <xsd:simpleContent>  
    <xsd:extension base=" Scheme ">  
      <xsd:attribute name="commodityMarketDisruptionScheme" type=" xsd:anyURI " default="http://  
        www.fpml.org/coding-scheme/commodity-market-disruption"/>  
    </xsd:extension>  
  </xsd:simpleContent>  
</xsd:complexType>
```

[top](#)

## Complex Type: NonPeriodicFixedPriceLeg

<b>Super-types:</b>	<a href="#">Leg</a> < <b>NonPeriodicFixedPriceLeg</b> (by extension)
<b>Sub-types:</b>	None
<b>Name</b>	NonPeriodicFixedPriceLeg
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">CommodityForward</a>
<b>Abstract</b>	no
<b>Documentation</b>	The details of a fixed payment. Can be used for a forward transaction or as the base for a more complex fixed leg component such as the fixed leg of a swap.

**XML Instance Representation**

```
<...  
id=" xsd:ID [0..1]">  
  <payerPartyReference> PartyReference </payerPartyReference> [1]  
  'A reference to the party responsible for making the payments defined by this structure.'  
  
  <payerAccountReference> AccountReference </payerAccountReference> [0..1]  
  'A reference to the account responsible for making the payments defined by this structure.'  
  
  <receiverPartyReference> PartyReference </receiverPartyReference> [1]  
  'A reference to the party that receives the payments corresponding to this structure.'  
  
  <receiverAccountReference> AccountReference </receiverAccountReference> [0..1]  
  'A reference to the account that receives the payments corresponding to this structure.'  
  
  <fixedPrice> FixedPrice </fixedPrice> [1]  
  'Fixed price on which fixed payments are based.'  
  
  <totalPrice> NonNegativeMoney </totalPrice> [0..1]  
  'The total amount of the fixed payment for all units of the underlying commodity.'
```

```

<quantityReference> QuantityReference </quantityReference> [1]
'A pointer style reference to a quantity defined on another leg.'

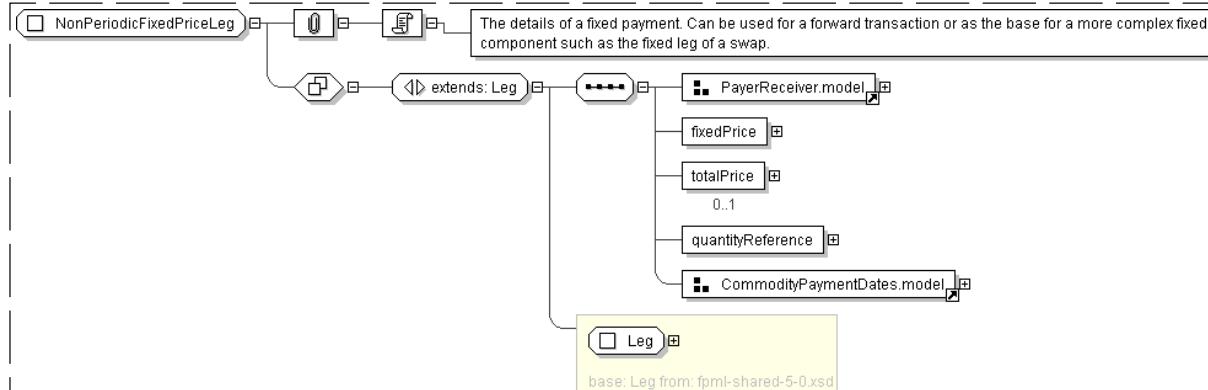
Start Choice [1]
<relativePaymentDates> CommodityRelativePaymentDates </relativePaymentDates> [1]
'The Payment Dates of the trade relative to the Calculation Periods.'

Start Choice [1]
<paymentDates> AdjustableDatesOrRelativeDateOffset </paymentDates> [1]
'Dates on which payments will be made.'

<masterAgreementPaymentDates> xsd:boolean </masterAgreementPaymentDates> [1]
'If present and true indicates that the Payment Date(s) are specified in the relevant
master agreement.'

End Choice
End Choice
</...>

```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="NonPeriodicFixedPriceLeg">
  <xsd:complexContent>
    <xsd:extension base=" Leg ">
      <xsd:sequence>
        <xsd:group ref=" PayerReceiver.model ">
          <xsd:element name="fixedPrice" type=" FixedPrice ">/>
          <xsd:element name="totalPrice" type=" NonNegativeMoney " minOccurs="0">/>
          <xsd:element name="quantityReference" type=" QuantityReference ">/>
        </xsd:group>
        <xsd:group ref=" CommodityPaymentDates.model ">/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

top

**Complex Type: OilDelivery**

Super-types:	None
Sub-types:	None

Name	OilDelivery
Used by (from the same schema document)	Complex Type <a href="#">OilPhysicalLeg</a>
Abstract	no
Documentation	The physical delivery conditions for an oil product.

**XML Instance Representation**

```
<...>
Start Choice [1]
  <pipeline> OilPipelineDelivery </pipeline> [1]
    'Specified the delivery conditions where the oil product is to be delivered by pipeline.'
  <transfer> OilTransferDelivery </transfer> [1]
    'Specified the delivery conditions where the oil product is to be delivered by title transfer.'
```

End Choice

```
<importerOfRecord> PartyReference </importerOfRecord> [0..1]
  'Specifies which party is the Importer of Record for the purposes of paying customs duties and applicable taxes or costs related to the import of the oil product.'
```

Start Choice [0..1]

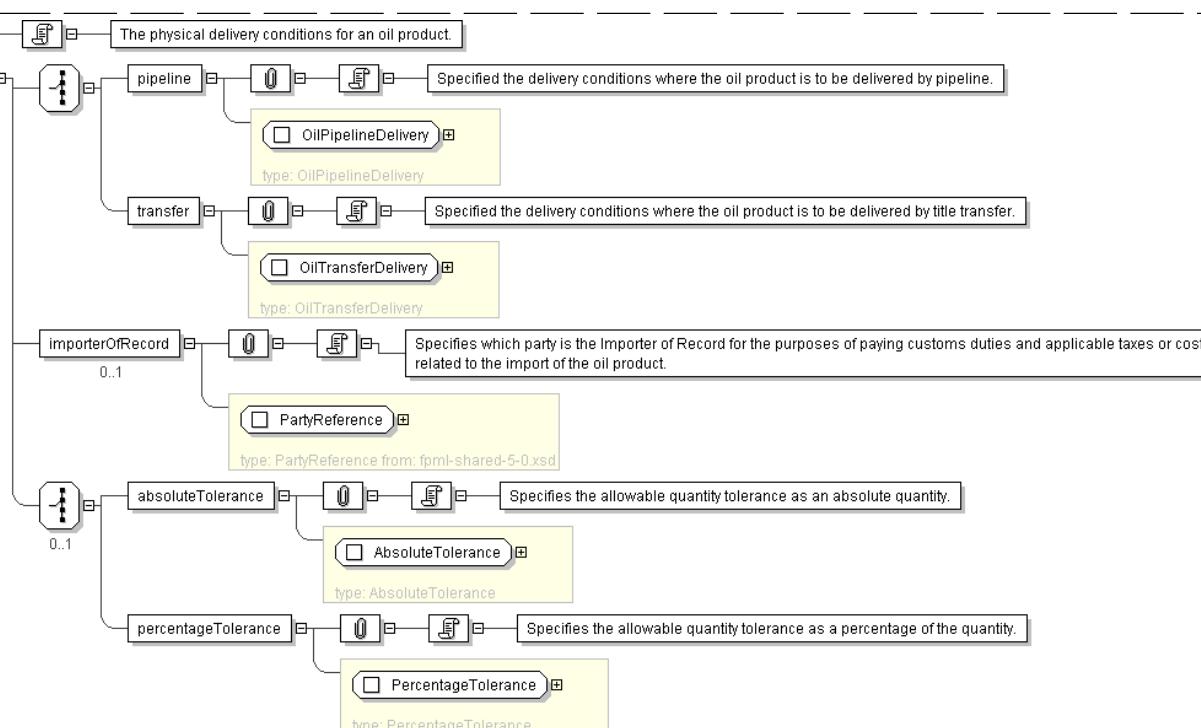
```
<absoluteTolerance> AbsoluteTolerance </absoluteTolerance> [1]
  'Specifies the allowable quantity tolerance as an absolute quantity.'
```

```
<percentageTolerance> PercentageTolerance </percentageTolerance> [1]
  'Specifies the allowable quantity tolerance as a percentage of the quantity.'
```

End Choice

</...>

#### Diagram



#### Schema Component Representation

```

<xsd:complexType name="OilDelivery">
  <xsd:sequence>
    <xsd:choice>
      <xsd:element name="pipeline" type=" OilPipelineDelivery " />
      <xsd:element name="transfer" type=" OilTransferDelivery " />
    </xsd:choice>
    <xsd:element name="importerOfRecord" type=" PartyReference " minOccurs="0" />
    <xsd:choice minOccurs="0">
      <xsd:element name="absoluteTolerance" type=" AbsoluteTolerance " />
      <xsd:element name="percentageTolerance" type=" PercentageTolerance " />
    </xsd:choice>
  </xsd:sequence>
</xsd:complexType>
  
```

```
</xsd:sequence>
</xsd:complexType>
```

**Complex Type: OilPhysicalLeg**

<b>Super-types:</b>	<a href="#">Leg</a> < <a href="#">PhysicalLeg</a> (by extension) < <b>OilPhysicalLeg</b> (by extension)
<b>Sub-types:</b>	None

<b>Name</b>	OilPhysicalLeg
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">CommoditySwap</a>
<b>Abstract</b>	no
<b>Documentation</b>	Physically settled leg of a physically settled oil product transaction.

**XML Instance Representation**

```
<...
  id="xsd:ID [0..1]">
  <payerPartyReference> PartyReference </payerPartyReference> [1]
  'A reference to the party responsible for making the payments defined by this structure.'

  <payerAccountReference> AccountReference </payerAccountReference> [0..1]
  'A reference to the account responsible for making the payments defined by this structure.'

  <receiverPartyReference> PartyReference </receiverPartyReference> [1]
  'A reference to the party that receives the payments corresponding to this structure.'

  <receiverAccountReference> AccountReference </receiverAccountReference> [0..1]
  'A reference to the account that receives the payments corresponding to this structure.'

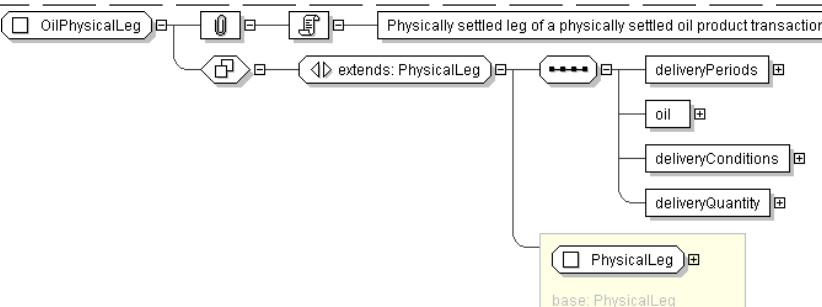
  <deliveryPeriods> CommodityDeliveryPeriods </deliveryPeriods> [1]
  'The different options for specifying the Delivery or Supply Periods. Unless the quantity or price is to vary periodically during the trade or physical delivery occurs on a periodic basis, periodsSchedule should be used and set to IT.'

  <oil> OilProduct </oil> [1]
  'The specification of the oil product to be delivered.'

  <deliveryConditions> OilDelivery </deliveryConditions> [1]
  'The physical delivery conditions for the transaction.'

  <deliveryQuantity> CommodityPhysicalQuantity </deliveryQuantity> [1]
  'The different options for specifying the quantity.'

</...>
```

**Diagram****Schema Component Representation**

```
<xsd:complexType name="OilPhysicalLeg">
  <xsd:complexContent>
    <xsd:extension base=" PhysicalLeg ">
      <xsd:sequence>
```

```

<xsd:element name="deliveryPeriods" type="#CommodityDeliveryPeriods" />
<xsd:element name="oil" type="#OilProduct" />
<xsd:element name="deliveryConditions" type="#OilDelivery" />
<xsd:element name="deliveryQuantity" type="#CommodityPhysicalQuantity" />
</xsd:sequence>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>

```

## Complex Type: OilPipelineDelivery

Super-types:	None
Sub-types:	None

Name	OilPipelineDelivery
Used by (from the same schema document)	Complex Type <a href="#">OilDelivery</a>
Abstract	no
Documentation	The physical delivery conditions specific to an oil product delivered by pipeline.

### XML Instance Representation

```

<...>
<pipelineName> CommodityPipeline </pipelineName> [1]
'The name of pipeline by which the oil product will be delivered.'

<withdrawalPoint> CommodityDeliveryPoint </withdrawalPoint> [0..1]
'The location at which the transfer of the title to the commodity takes place.'

<entryPoint> CommodityDeliveryPoint </entryPoint> [0..1]
'The point at which the oil product will enter the pipeline.'

<deliverableByBarge> xsd:boolean </deliverableByBarge> [1]
'Whether or not the delivery can go to barge. For trades documented under the ISDA
Master Agreement and Oil Annex, this should always be set to \'false\'.'

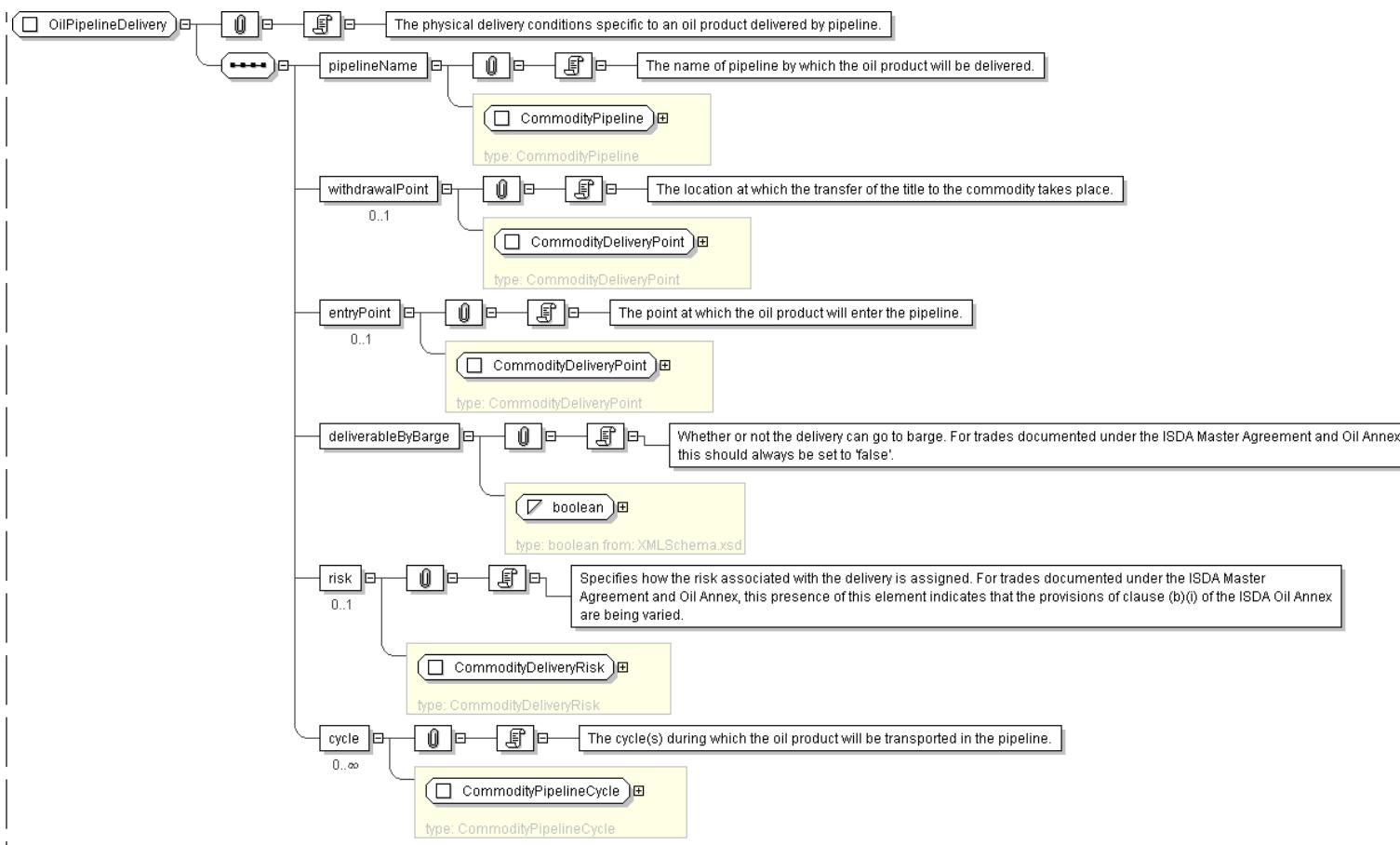
<risk> CommodityDeliveryRisk </risk> [0..1]
'Specifies how the risk associated with the delivery is assigned. For trades documented
under the ISDA Master Agreement and Oil Annex, this presence of this element indicates that
the provisions of clause (b)(i) of the ISDA Oil Annex are being varied.'

<cycle> CommodityPipelineCycle </cycle> [0..*]
'The cycle(s) during which the oil product will be transported in the pipeline.'

</...>

```

### Diagram

**Schema Component Representation**

```

<xsd:complexType name="OilPipelineDelivery">
  <xsd:sequence>
    <xsd;element name="pipelineName" type="CommodityPipeline" />
    <xsd;element name="withdrawalPoint" type="CommodityDeliveryPoint" minOccurs="0" />
    <xsd;element name="entryPoint" type="CommodityDeliveryPoint" minOccurs="0" />
    <xsd;element name="deliverableByBarge" type="xsd:boolean" />
    <xsd;element name="risk" type="CommodityDeliveryRisk" minOccurs="0" />
    <xsd;element name="cycle" type="CommodityPipelineCycle" minOccurs="0" maxOccurs="unbounded" />
  </xsd:sequence>
</xsd:complexType>
  
```

top

**Complex Type: OilProduct**

Super-types:	None
Sub-types:	None

Name	OilProduct
Used by (from the same schema document)	Complex Type <a href="#">OilPhysicalLeg</a>
Abstract	no
Documentation	The specification of the oil product to be delivered.

**XML Instance Representation**

&lt;...&gt;

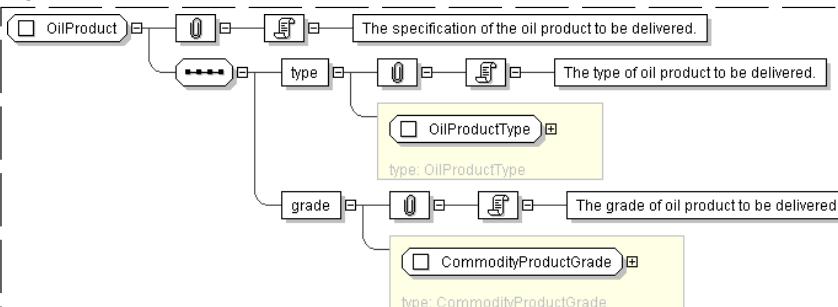
```

<type> OilProductType </type> [1]
'The type of oil product to be delivered.'

<grade> CommodityProductGrade </grade> [1]
'The grade of oil product to be delivered.'

</...>

```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="OilProduct">
  <xsd:sequence>
    <xsd:element name="type" type=" OilProductType " />
    <xsd:element name="grade" type=" CommodityProductGrade " />
  </xsd:sequence>
</xsd:complexType>

```

[top](#)**Complex Type: OilProductType**

<b>Super-types:</b>	<a href="#">Scheme</a> < <b>OilProductType</b> (by extension)
<b>Sub-types:</b>	None

<b>Name</b>	OilProductType
-------------	----------------

<b>Used by (from the same schema document)</b>	Complex Type <a href="#">OilProduct</a>
--	---

<b>Abstract</b>	no
-----------------	----

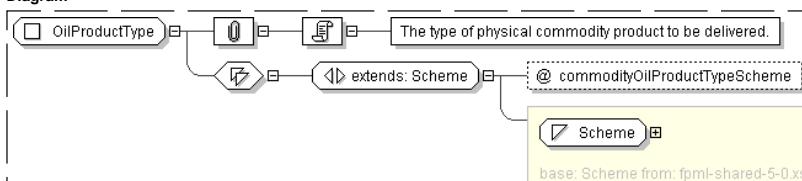
<b>Documentation</b>	The type of physical commodity product to be delivered.
----------------------	---

**XML Instance Representation**

```

<...
  commodityOilProductTypeScheme=" xsd:anyURI [0..1]">
  <a href="#">Scheme</a>
</...>

```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="OilProductType">
  <xsd:simpleContent>
    <xsd:extension base=" Scheme ">
      <xsd:attribute name="commodityOilProductTypeScheme" type=" xsd:anyURI " default="http://
        www.fpml.org/coding-scheme/commodity-oil-product-type"/>
    </xsd:extension>

```

```
</xsd:simpleContent>
</xsd:complexType>
```

## Complex Type: OilTransferDelivery

Super-types:	None
Sub-types:	None

Name	OilTransferDelivery
Used by (from the same schema document)	Complex Type <a href="#">OilDelivery</a>
Abstract	no
Documentation	The physical delivery conditions specific to an oil product delivered by title transfer.

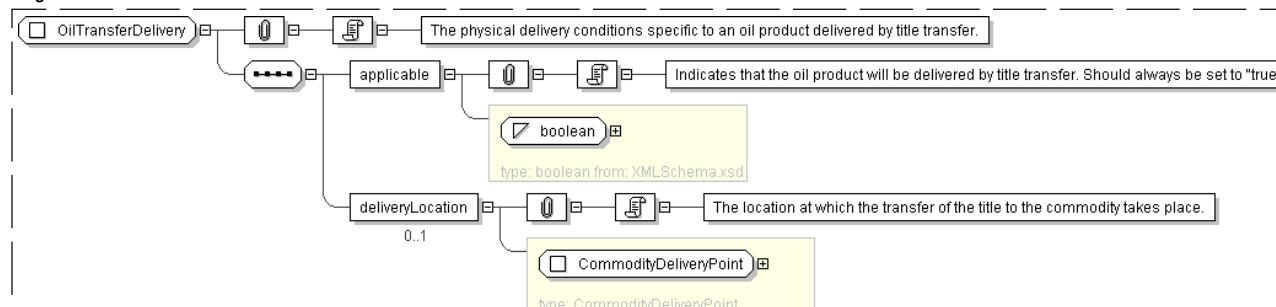
### XML Instance Representation

```
<...>
<applicable> xsd:boolean </applicable> [1]
'Indicates that the oil product will be delivered by title transfer. Should always be set
to "true".'

<deliveryLocation> CommodityDeliveryPoint </deliveryLocation> [0..1]
'The location at which the transfer of the title to the commodity takes place.'
```

&lt;/...&gt;

### Diagram



### Schema Component Representation

```
<xsd:complexType name="OilTransferDelivery">
  <xsd:sequence>
    <xsd:element name="applicable" type=" xsd:boolean " />
    <xsd:element name="deliveryLocation" type=" CommodityDeliveryPoint " minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>
```

## Complex Type: PercentageTolerance

Super-types:	None
Sub-types:	None

Name	PercentageTolerance
Used by (from the same schema document)	Complex Type <a href="#">OilDelivery</a>
Abstract	no
Documentation	The acceptable tolerance in the delivered quantity of a physical commodity product in terms of a percentage of the agreed delivery quantity.

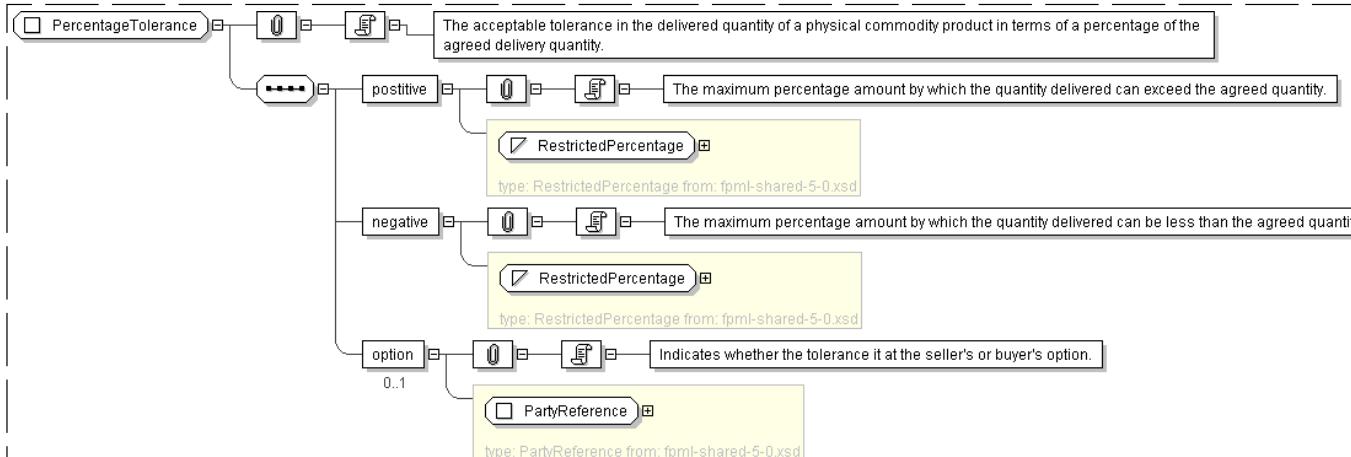
### XML Instance Representation

```
<...>
<positive> RestrictedPercentage </positive> [1]
'The maximum percentage amount by which the quantity delivered can exceed the agreed quantity.'
```

```
<negative> RestrictedPercentage </negative> [1]
'The maximum percentage amount by which the quantity delivered can be less than the
agreed quantity.'
```

```
<option> PartyReference </option> [0..1]
'Indicates whether the tolerance it at the seller\'s or buyer\'s option.'
```

```
<...>
```

**Diagram****Schema Component Representation**

```
<xsd:complexType name="PercentageTolerance">
  <xsd:sequence>
    <xsd;element name="positive" type=" RestrictedPercentage "/>
    <xsd;element name="negative" type=" RestrictedPercentage "/>
    <xsd;element name="option" type=" PartyReference " minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>
```

top

**Complex Type: PhysicalLeg****Super-types:**[Leg](#) < **PhysicalLeg** (by extension)**Sub-types:**

- [BullionPhysicalLeg](#) (by extension)
- [CoalPhysicalLeg](#) (by extension)
- [ElectricityPhysicalLeg](#) (by extension)
- [GasPhysicalLeg](#) (by extension)
- [OilPhysicalLeg](#) (by extension)

**Name**

PhysicalLeg

**Abstract**

yes

**Documentation**

The common components of a physically settled leg of a Commodity Swap. This is an abstract type and should be extended by commodity-specific types.

**XML Instance Representation**

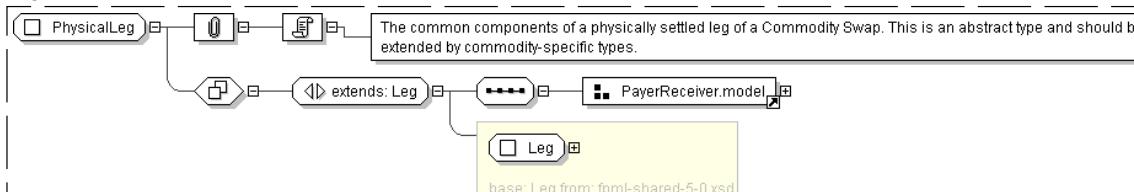
```
<...
id=" xsd:ID [0..1]">
<payerPartyReference> PartyReference </payerPartyReference> [1]
'A reference to the party responsible for making the payments defined by this structure.'
```

```
<payerAccountReference> AccountReference </payerAccountReference> [0..1]
'A reference to the account responsible for making the payments defined by this structure.'
```

```
<receiverPartyReference> PartyReference </receiverPartyReference> [1]
'A reference to the party that receives the payments corresponding to this structure.'
```

```
<receiverAccountReference> AccountReference </receiverAccountReference> [0..1]
'A reference to the account that receives the payments corresponding to this structure.'
```

&lt;/...&gt;

**Diagram****Schema Component Representation**

```
<xsd:complexType name="PhysicalLeg" abstract="true">
  <xsd:complexContent>
    <xsd:extension base=" Leg ">
      <xsd:sequence>
        <xsd:group ref=" PayerReceiver.model " />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

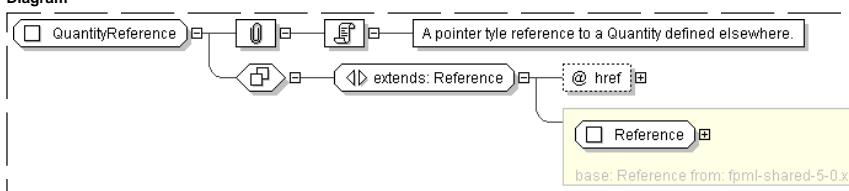
[top](#)**Complex Type: QuantityReference**

<b>Super-types:</b>	<a href="#">Reference</a> < <b>QuantityReference</b> (by extension)
<b>Sub-types:</b>	None

<b>Name</b>	QuantityReference
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">NonPeriodicFixedPriceLeg</a> , Model Group <a href="#">CommodityNotionalQuantity.model</a>
<b>Abstract</b>	no
<b>Documentation</b>	A pointer type reference to a Quantity defined elsewhere.

**XML Instance Representation**

```
<...
  href="#" type="IDREF" />
```

**Diagram****Schema Component Representation**

```
<xsd:complexType name="QuantityReference">
  <xsd:complexContent>
    <xsd:extension base=" Reference ">
      <xsd:attribute name="href" type=" xsd:IDREF "
        use="required" reference="CommodityPhysicalQuantityBase" />
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

[top](#)

Complex Type: **QuantityScheduleReference**

Super-types:

[Reference](#) < **QuantityScheduleReference** (by extension)

Sub-types:

None

Name **QuantityScheduleReference**

Abstract

no

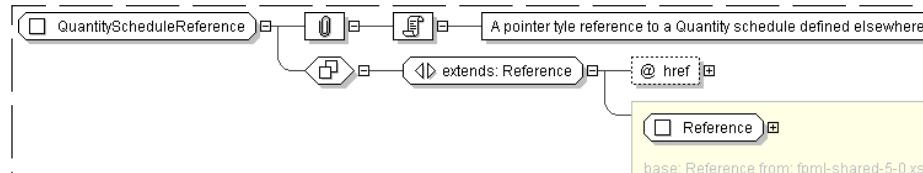
Documentation

A pointer type reference to a Quantity schedule defined elsewhere.

## XML Instance Representation

```
<...>
  href="# IDREF [1]" />
```

## Diagram



## Schema Component Representation

```

<xsd:complexType name="QuantityScheduleReference">
  <xsd:complexContent>
    <xsd:extension base=" Reference ">
      <xsd:attribute name="href" type=" xsd:IDREF "
        use="required" reference="CommodityNotionalQuantitySchedule"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
  
```

top

Complex Type: **SequencedDisruptionFallback**

Super-types:

None

Sub-types:

None

Name **SequencedDisruptionFallback**Used by (from the same schema document) Complex Type [CommodityMarketDisruption](#)

Abstract

no

Documentation

A Disruption Fallback with the sequence in which it should be applied relative to other Disruption Fallbacks.

## XML Instance Representation

```

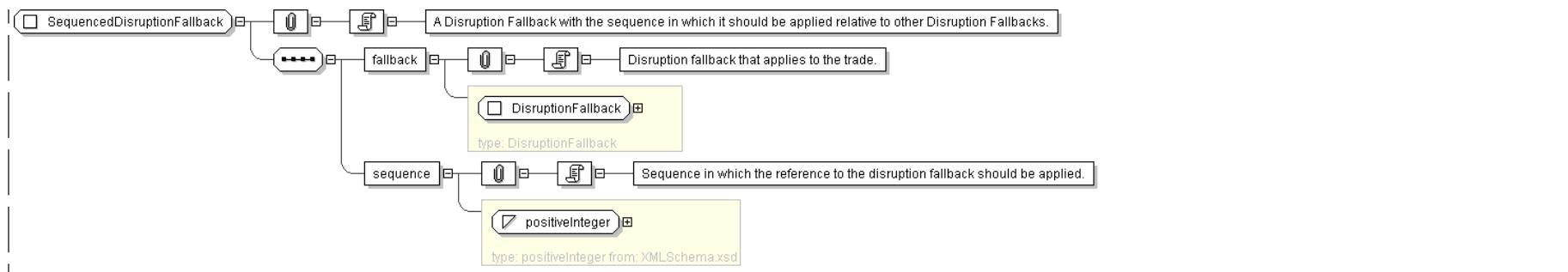
<...>
  <fallback> DisruptionFallback </fallback> [1]
  'Disruption fallback that applies to the trade.'

  <sequence> xsd:positiveInteger </sequence> [1]
  'Sequence in which the reference to the disruption fallback should be applied.'

</...>
  
```

## Diagram



**Schema Component Representation**

```

<xsd:complexType name="SequencedDisruptionFallback">
  <xsd:sequence>
    <xsd:element name="fallback" type="DisruptionFallback" />
    <xsd:element name="sequence" type="xsd:positiveInteger" />
  </xsd:sequence>
</xsd:complexType>
  
```

top

**Complex Type: SettlementPeriods**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None
<b>Name</b>	SettlementPeriods
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">CommodityPricingDates</a> , Complex Type <a href="#">ElectricityDeliveryPeriods</a> , Complex Type <a href="#">ElectricityPhysicalLeg</a>
<b>Abstract</b>	no
<b>Documentation</b>	Specifies a set of Settlement Periods associated with an Electricity Transaction for delivery on an Applicable Day or for a series of Applicable Days.

**XML Instance Representation**

```

<...
  id="xsd:ID [0..1]"
  <duration> SettlementPeriodDurationEnum </duration> [1]
  'The length of each Settlement Period.'

  <applicableDay> DayOfWeekEnum </applicableDay> [0..7]
  'Specifies the Applicable Day with respect to a range of Settlement Periods. This element
  can only be omitted if includesHolidays is present, in which case this range of
  Settlement Periods will apply to days that are holidays only.'

  <startTime> OffsetPrevailingTime </startTime> [1]
  'Specifies the hour-ending Start Time with respect to a range of Settlement Periods.'

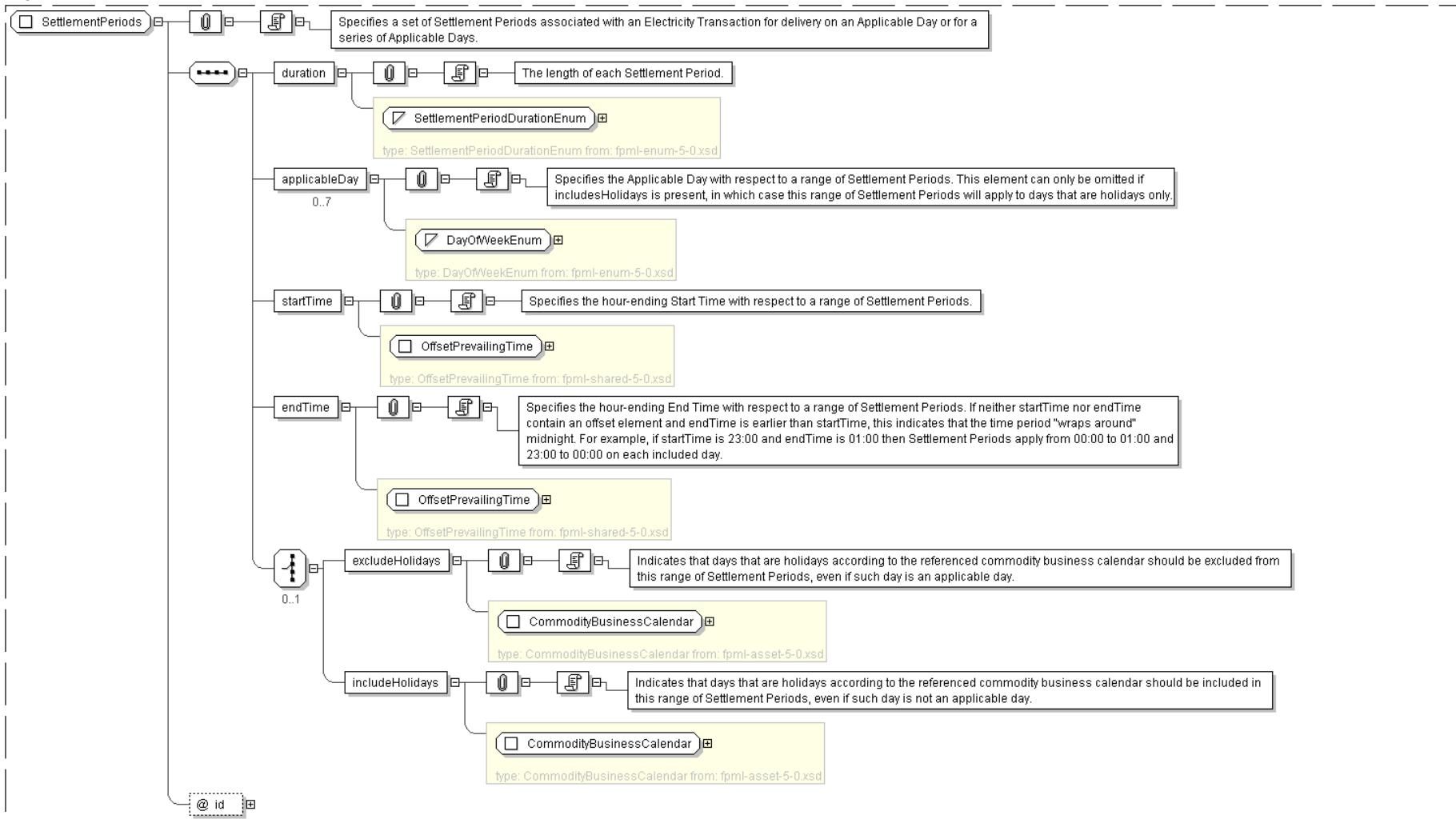
  <endTime> OffsetPrevailingTime </endTime> [1]
  'Specifies the hour-ending End Time with respect to a range of Settlement Periods. If
  neither startTime nor endTime contain an offset element and endTime is earlier than
  startTime, this indicates that the time period "wraps around" midnight. For example,
  if startTime is 23:00 and endTime is 01:00 then Settlement Periods apply from 00:00 to
  01:00 and 23:00 to 00:00 on each included day.'

  Start Choice [0..1]
  <excludeHolidays> CommodityBusinessCalendar </excludeHolidays> [1]
  'Indicates that days that are holidays according to the referenced commodity business
  calendar should be excluded from this range of Settlement Periods, even if such day is
  an applicable day.'

  <includeHolidays> CommodityBusinessCalendar </includeHolidays> [1]
  'Indicates that days that are holidays according to the referenced commodity business
  calendar should be included in this range of Settlement Periods, even if such day is not
  an applicable day.'

  End Choice
<...>
  
```

## Diagram



## Schema Component Representation

```

<xsd:complexType name="SettlementPeriods">
  <xsd:sequence>
    <xsd:element name="duration" type=" SettlementPeriodDurationEnum " />
    <xsd:element name="applicableDay" type=" DayOfWeekEnum " minOccurs="0" maxOccurs="7" />
    <xsd:element name="startTime" type=" OffsetPrevailingTime " />
    <xsd:element name="endTime" type=" OffsetPrevailingTime " />
    <xsd:choice minOccurs="0">
      <xsd:element name="excludeHolidays" type=" CommodityBusinessCalendar " />
      <xsd:element name="includeHolidays" type=" CommodityBusinessCalendar " />
    </xsd:choice>
  </xsd:sequence>
  <xsd:attribute name="id" type=" xsd:ID " />
</xsd:complexType>

```

## Complex Type: SettlementPeriodsFixedPrice

**Super-types:** [FixedPrice](#) < **SettlementPeriodsFixedPrice** (by extension)

None

**Name** SettlementPeriodsFixedPrice

**Used by (from the same schema document)** Model Group [CommodityFixedPrice.model](#)

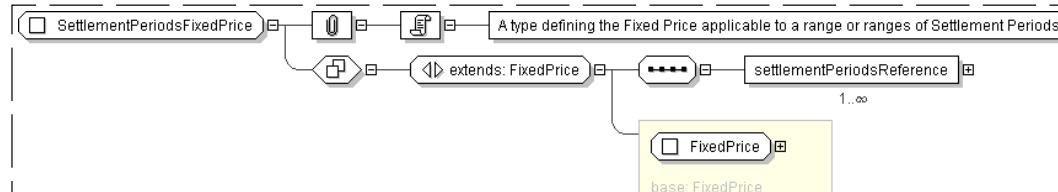
**Abstract** no

**Documentation** A type defining the Fixed Price applicable to a range or ranges of Settlement Periods.

#### XML Instance Representation

```
<...>
  id="xsd:ID [0..1]">
    <price> xsd:decimal </price> [1]
    'The Fixed Price.'
    <priceCurrency> Currency </priceCurrency> [1]
    'Currency of the fixed price.'
    <priceUnit> QuantityUnit </priceUnit> [1]
    'The unit of measure used to calculate the Fixed Price.'
    <settlementPeriodsReference> SettlementPeriodsReference </settlementPeriodsReference> [1..*]
</...>
```

#### Diagram



#### Schema Component Representation

```

<xsd:complexType name="SettlementPeriodsFixedPrice">
  <xsd:complexContent>
    <xsd:extension base=" FixedPrice ">
      <xsd:sequence>
        <xsd:element name="settlementPeriodsReference" type=" SettlementPeriodsReference "
          " maxOccurs="unbounded"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
  
```

top

## Complex Type: **SettlementPeriodsReference**

**Super-types:** [Reference](#) < **SettlementPeriodsReference** (by extension)

None

**Name** SettlementPeriodsReference

**Used by (from the same schema document)** Complex Type [CommodityPricingDates](#) , Complex Type [CommoditySettlementPeriodsNotionalQuantity](#) , Complex Type [CommoditySettlementPeriodsNotionalQuantitySchedule](#) , Complex Type [CommoditySettlementPeriodsPriceSchedule](#) , Complex Type [ElectricityPhysicalDeliveryQuantity](#) , Complex Type [ElectricityPhysicalDeliveryQuantitySchedule](#) , Complex Type [SettlementPeriodsFixedPrice](#) , Complex Type [SettlementPeriodsStep](#)

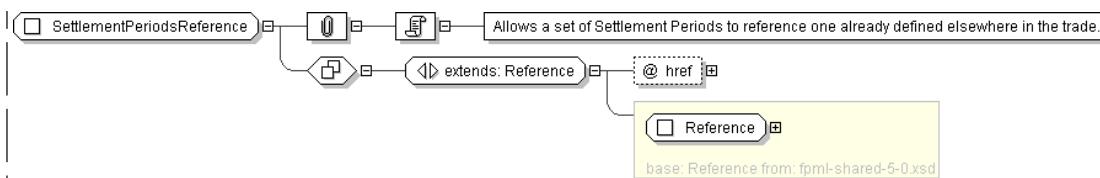
**Abstract** no

**Documentation** Allows a set of Settlement Periods to reference one already defined elsewhere in the trade.

#### XML Instance Representation

```
<...>
  href=" xsd:IDREF [1]" />
```

#### Diagram

**Schema Component Representation**

```
<xsd:complexType name="SettlementPeriodsReference">
  <xsd:complexContent>
    <xsd:extension base=" Reference ">
      <xsd:attribute name="href" type=" xsd:IDREF " use="required" reference="SettlementPeriods" />
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

[top](#)**Complex Type: SettlementPeriodsSchedule**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None

<b>Name</b>	SettlementPeriodsSchedule
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">ElectricityPhysicalLeg</a>
<b>Abstract</b>	no
<b>Documentation</b>	The specification of the Settlement Periods in which the electricity will be delivered for a "shaped" trade i.e. where different Settlement Period ranges will apply to different periods of the trade.

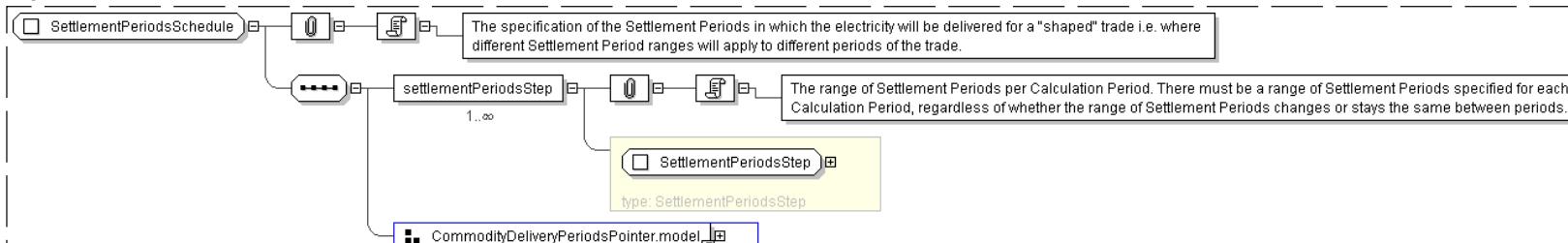
**XML Instance Representation**

```
<...>
<settlementPeriodsStep> SettlementPeriodsStep </settlementPeriodsStep> [1...*]
  'The range of Settlement Periods per Calculation Period. There must be a range of
  Settlement Periods specified for each Calculation Period, regardless of whether the range
  of Settlement Periods changes or stays the same between periods.'

Start Choice [1]
  <deliveryPeriodsReference> CalculationPeriodsReference </deliveryPeriodsReference> [1]
    'A pointer style reference to the Delivery Periods defined elsewhere.'

  <deliveryPeriodsScheduleReference> CalculationPeriodsScheduleReference
  </deliveryPeriodsScheduleReference> [1]
    'A pointer style reference to the Calculation Periods Schedule defined elsewhere.'

End Choice
</...>
```

**Diagram****Schema Component Representation**

```
<xsd:complexType name="SettlementPeriodsSchedule">
  <xsd:sequence>
    <xsd;element name="settlementPeriodsStep" type=" SettlementPeriodsStep " maxOccurs="unbounded" />
    <xsd:group ref=" CommodityDeliveryPeriodsPointer.model " />
```

```
</xsd:sequence>
</xsd:complexType>
```

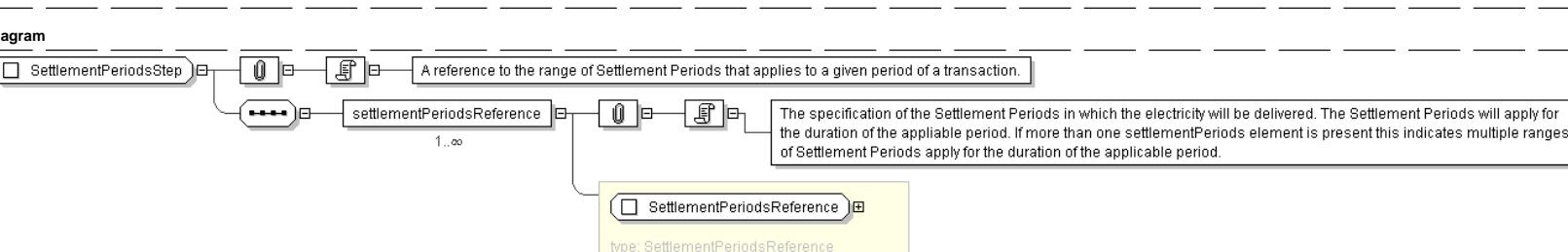
**Complex Type: SettlementPeriodsStep**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None

<b>Name</b>	SettlementPeriodsStep
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">SettlementPeriodsSchedule</a>
<b>Abstract</b>	no
<b>Documentation</b>	A reference to the range of Settlement Periods that applies to a given period of a transaction.

**XML Instance Representation**

```
<...>
<settlementPeriodsReference> SettlementPeriodsReference </settlementPeriodsReference> [1..*]
'The specification of the Settlement Periods in which the electricity will be delivered.
The Settlement Periods will apply for the duration of the applicable period. If more than
one settlementPeriods element is present this indicates multiple ranges of Settlement
Periods apply for the duration of the applicable period.'
```

**Diagram****Schema Component Representation**

```
<xsd:complexType name="SettlementPeriodsStep">
  <xsd:sequence>
    <xsd:element name="settlementPeriodsReference" type="#SettlementPeriodsReference
      " maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>
```

**Complex Type: UnitQuantity**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None

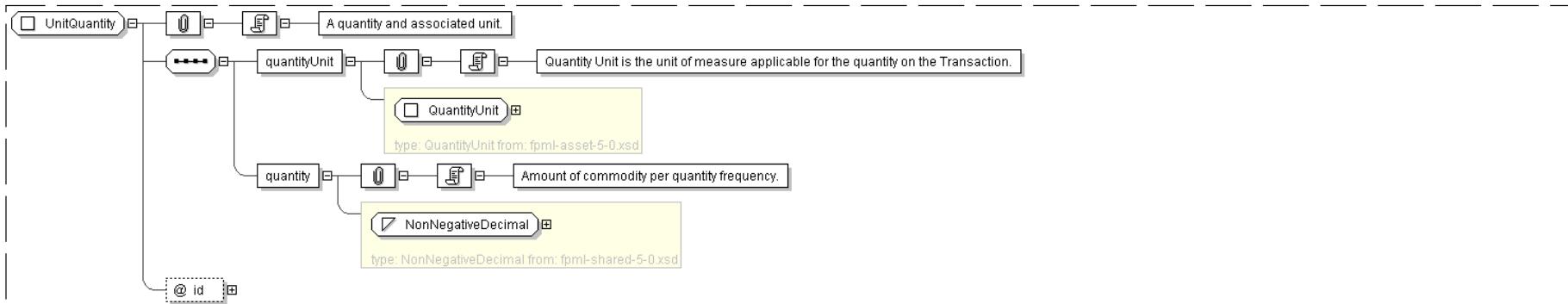
<b>Name</b>	UnitQuantity
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">ElectricityPhysicalQuantity</a> , Model Group <a href="#">CommodityFixedPhysicalQuantity.model</a>
<b>Abstract</b>	no
<b>Documentation</b>	A quantity and associated unit.

**XML Instance Representation**

```
<...
id="#ID [0..1]">
<quantityUnit> QuantityUnit </quantityUnit> [1]
'Quantity Unit is the unit of measure applicable for the quantity on the Transaction.'

<quantity> NonNegativeDecimal </quantity> [1]
'Amount of commodity per quantity frequency.'
```

## Diagram



## Schema Component Representation

```

<xsd:complexType name="UnitQuantity">
  <xsd:sequence>
    <xsd:element name="quantityUnit" type="#QuantityUnit" />
    <xsd:element name="quantity" type="NonNegativeDecimal" />
  </xsd:sequence>
  <xsd:attribute name="id" type="xsd:ID" />
</xsd:complexType>
  
```

top

## Model Group: CommodityAsian.model

Name	CommodityAsian.model
Used by (from the same schema document)	Model Group <a href="#">CommodityFinancialOption.model</a>
Documentation	Model group containing features specific to asian/averaging commodity options.

## XML Instance Representation

```

<effectiveDate> AdjustableOrRelativeDate </effectiveDate> [1]
  
```

'The effective date of the Commodity Option Transaction. Note that the Termination/Expiration Date should be specified in expirationDate within the CommodityAmericanExercise type or the CommodityEuropeanExercise type, as applicable.'

```

Start Choice [1]
  <calculationPeriodsSchedule> CommodityCalculationPeriodsSchedule </
  calculationPeriodsSchedule> [1]
  
```

'A parametric representation of the Calculation Periods of the Commodity Option Transaction.'

```

<calculationPeriods> AdjustableDates </calculationPeriods> [1]
  
```

'An absolute representation of the Calculation Period start dates of the Commodity Option Transaction.'

```

End Choice
  <pricingDates> CommodityPricingDates </pricingDates> [1]
  
```

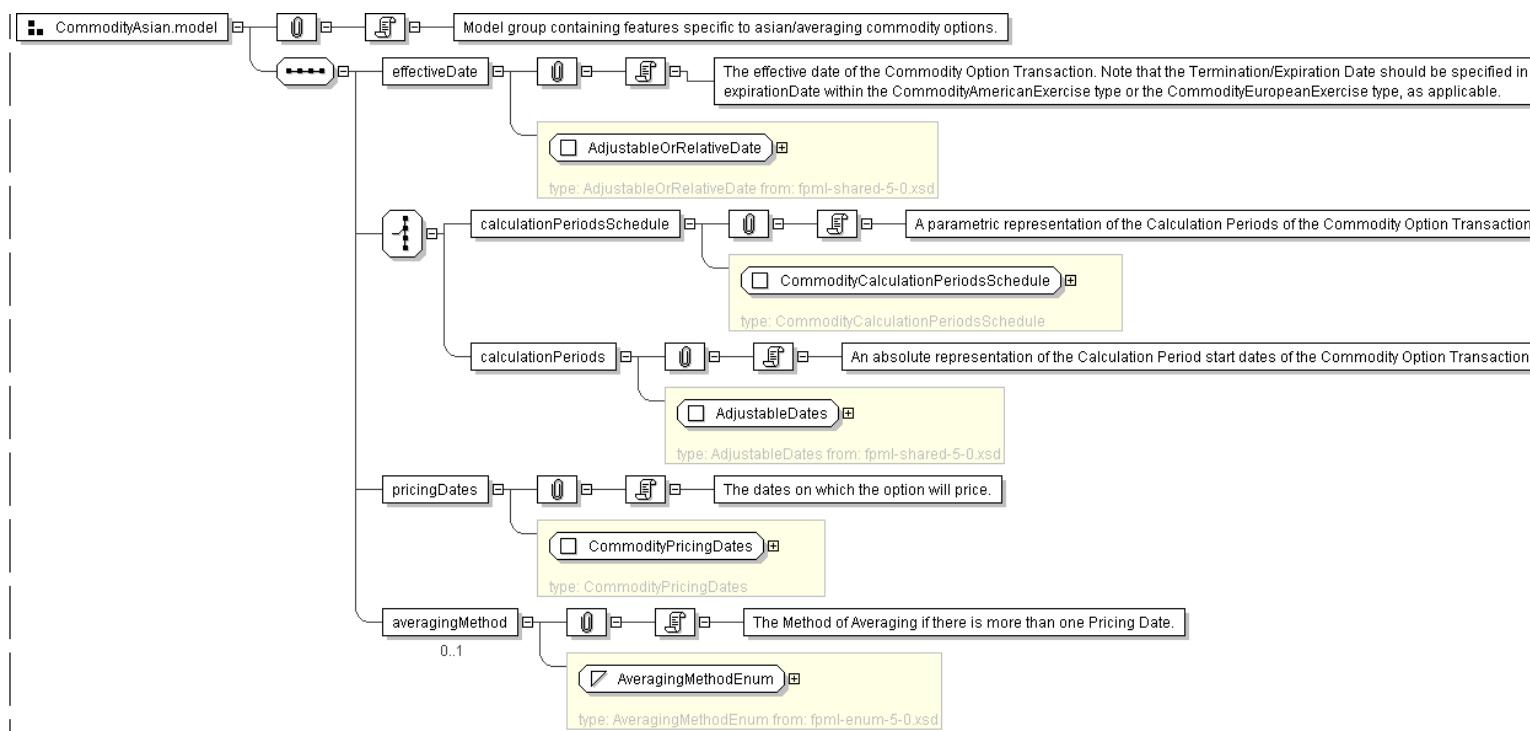
'The dates on which the option will price.'

```

<averagingMethod> AveragingMethodEnum </averagingMethod> [0..1]
  
```

'The Method of Averaging if there is more than one Pricing Date.'

## Diagram



#### Schema Component Representation

```

<xsd:group name="CommodityAsian.model">
  <xsd:sequence>
    <xsd:element name="effectiveDate" type=" AdjustableOrRelativeDate " />
    <xsd:choice>
      <xsd:element name="calculationPeriodsSchedule" type=" CommodityCalculationPeriodsSchedule " />
      <xsd:element name="calculationPeriods" type=" AdjustableDates " />
    </xsd:choice>
    <xsd:element name="pricingDates" type=" CommodityPricingDates " />
    <xsd:element name="averagingMethod" type=" AveragingMethodEnum " minOccurs="0 " />
  </xsd:sequence>
</xsd:group>
  
```

top

#### Model Group: CommodityCalculationPeriods.model

Name	CommodityCalculationPeriods.model
Used by (from the same schema document)	Complex Type <a href="#">FixedPriceLeg</a> , Complex Type <a href="#">FloatingPriceLeg</a>
Documentation	The different options for specifying the Calculation Periods.

#### XML Instance Representation

```

Start Choice [1]
<calculationDates> AdjustableDates </calculationDates> [1]
'The Calculation Period dates for this leg of the trade where the Calculation Periods are
all one day long, typically a physically-settled emissions or metals trade. Only
dates explicitly included determine the Calculation Periods and there is a Calculation
Period for each date specified.'
<calculationPeriods> AdjustableDates </calculationPeriods> [1]
'The Calculation Period start dates for this leg of the swap. This type is only intended to
be used if the Calculation Periods differ on each leg. If Calculation Periods mirror
another leg, then the calculationPeriodsReference element should be used to point to
the Calculation Periods on that leg - or the calculationPeriodsScheduleReference can be used
to point to the Calculation Periods Schedule for that leg.'
  
```

```
<calculationPeriodsSchedule> CommodityCalculationPeriodsSchedule </calculationPeriodsSchedule> [1]
```

'The Calculation Periods for this leg of the swap. This type is only intended to be used if the Calculation Periods differ on each leg. If Calculation Periods mirror another leg, then the calculationPeriodsReference element should be used to point to the Calculation Periods on the other leg - or the calculationPeriodsScheduleReference can be used to point to the Calculation Periods Schedule for that leg.'

Start Choice [1]

```
<calculationPeriodsReference> CalculationPeriodsReference </calculationPeriodsReference> [1]
```

'A pointer style reference to the Calculation Periods defined on another leg.'

```
<calculationPeriodsScheduleReference> CalculationPeriodsScheduleReference </calculationPeriodsScheduleReference> [1]
```

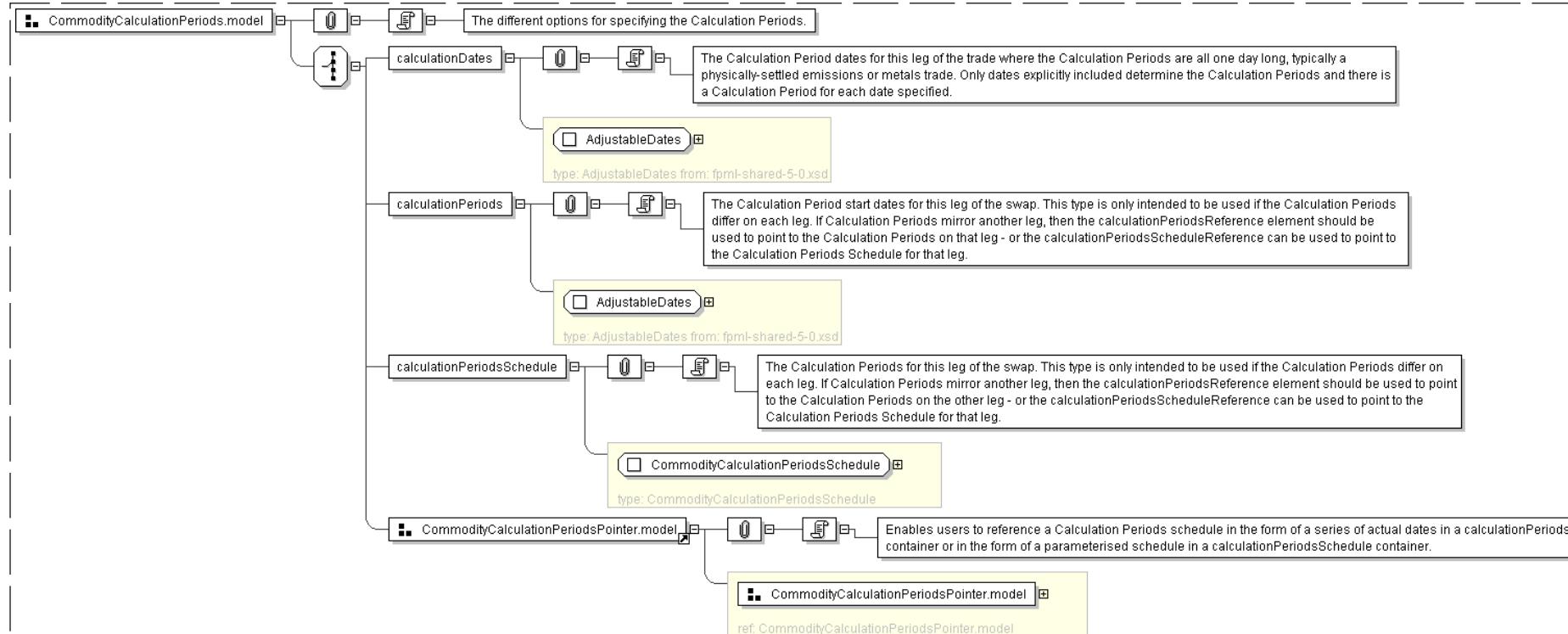
'A pointer style reference to the Calculation Periods Schedule defined on another leg.'

```
<calculationPeriodsDatesReference> CalculationPeriodsDatesReference </calculationPeriodsDatesReference> [1]
```

'A pointer style reference to single-day-duration Calculation Periods defined on another leg.'

End Choice  
End Choice

#### Diagram



#### Schema Component Representation

```
<xsd:group name="CommodityCalculationPeriods.model">
  <xsd:choice>
    <xsd:element name="calculationDates" type=" AdjustableDates " />
    <xsd:element name="calculationPeriods" type=" AdjustableDates " />
    <xsd:element name="calculationPeriodsSchedule" type=" CommodityCalculationPeriodsSchedule " />
  <xsd:group ref=" CommodityCalculationPeriodsPointer.model ">
  </xsd:choice>
</xsd:group>
```

**Model Group: CommodityCalculationPeriodsPointer.model**

<b>Name</b>	CommodityCalculationPeriodsPointer.model
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">CommodityDeliveryPeriods</a> , Complex Type <a href="#">CommodityFixedPriceSchedule</a> , Complex Type <a href="#">CommodityFx</a> , Complex Type <a href="#">CommodityNotionalQuantitySchedule</a> , Complex Type <a href="#">CommodityPricingDates</a> , Complex Type <a href="#">CommodityRelativePaymentDates</a> , Complex Type <a href="#">CommoditySpreadSchedule</a> , Complex Type <a href="#">CommodityStrikeSchedule</a> , Model Group <a href="#">CommodityCalculationPeriods.model</a>
<b>Documentation</b>	Model group to enable users to reference a Calculation Periods schedule in the form of a series of actual dates in a calculationPeriods container or in the form of a parameterised schedule in a calculationPeriodsSchedule container.

**XML Instance Representation**

```

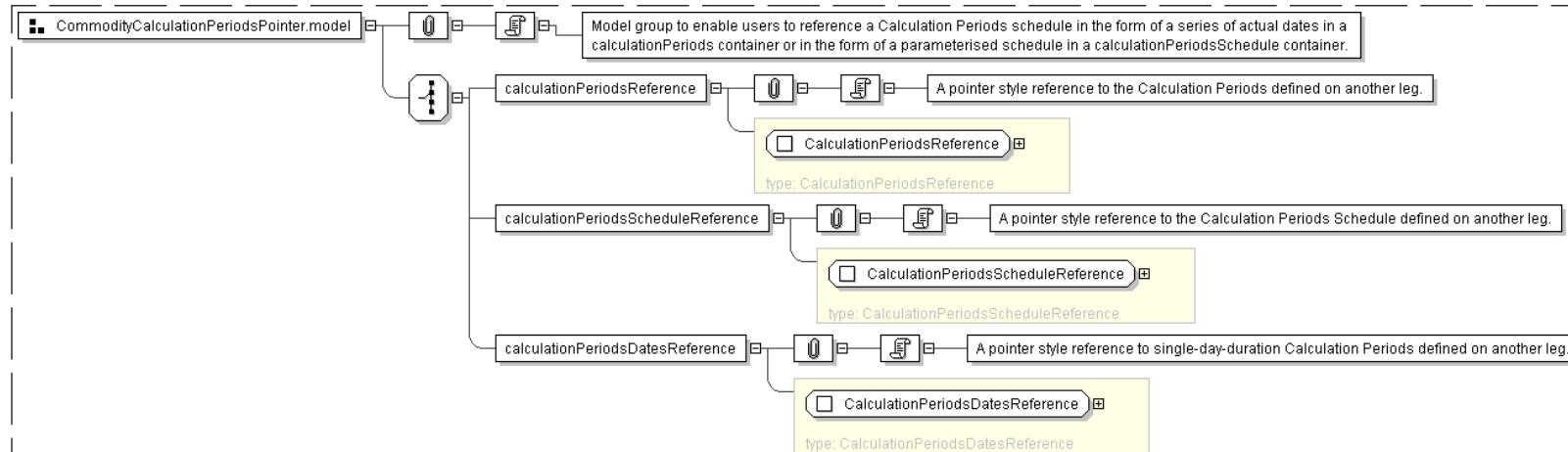
Start Choice [1]
<calculationPeriodsReference> CalculationPeriodsReference </calculationPeriodsReference> [1]
'A pointer style reference to the Calculation Periods defined on another leg.'

<calculationPeriodsScheduleReference> CalculationPeriodsScheduleReference
</calculationPeriodsScheduleReference> [1]
'A pointer style reference to the Calculation Periods Schedule defined on another leg.'

<calculationPeriodsDatesReference> CalculationPeriodsDatesReference
</calculationPeriodsDatesReference> [1]
'A pointer style reference to single-day-duration Calculation Periods defined on another leg.'

```

End Choice

**Diagram****Schema Component Representation**

```

<xsd:group name="CommodityCalculationPeriodsPointer.model">
  <xsd:choice>
    <xsd:element name="calculationPeriodsReference" type="#CalculationPeriodsReference" />
    <xsd:element name="calculationPeriodsScheduleReference"
      type="#CalculationPeriodsScheduleReference" />
    <xsd:element name="calculationPeriodsDatesReference" type="#CalculationPeriodsDatesReference" />
  </xsd:choice>
</xsd:group>

```

**Model Group: CommodityCoalComposition.model**

<b>Name</b>	CommodityCoalComposition.model
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">CoalStandardQuality</a>
<b>Documentation</b>	Items defining the chemical composition of the coal product.

**XML Instance Representation**

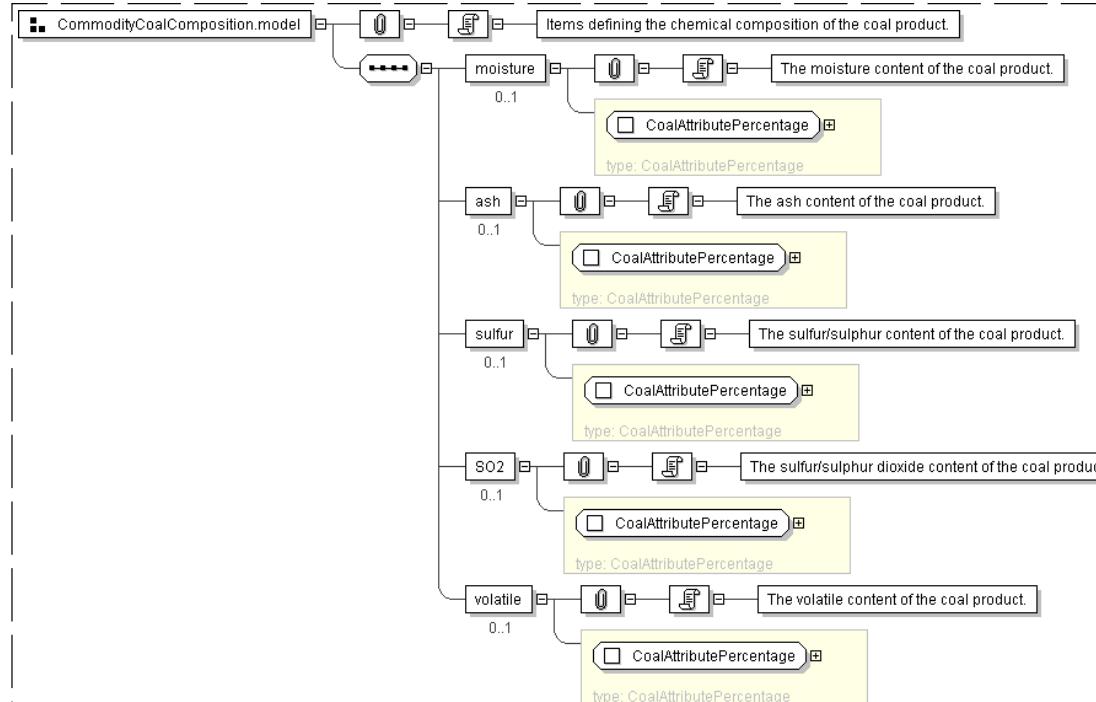
```
<moisture> CoalAttributePercentage </moisture> [0..1]
'The moisture content of the coal product.'

<ash> CoalAttributePercentage </ash> [0..1]
'The ash content of the coal product.'

<sulfur> CoalAttributePercentage </sulfur> [0..1]
'The sulfur/sulphur content of the coal product.'

<SO2> CoalAttributePercentage </SO2> [0..1]
'The sulfur/sulphur dioxide content of the coal product.'

<volatile> CoalAttributePercentage </volatile> [0..1]
'The volatile content of the coal product.'
```

**Diagram****Schema Component Representation**

```

<xsd:group name="CommodityCoalComposition.model">
  <xsd:sequence>
    <xsd:element name="moisture" type="CoalAttributePercentage" minOccurs="0"/>
    <xsd:element name="ash" type="CoalAttributePercentage" minOccurs="0"/>
    <xsd:element name="sulfur" type="CoalAttributePercentage" minOccurs="0"/>
    <xsd:element name="SO2" type="CoalAttributePercentage" minOccurs="0"/>
    <xsd:element name="volatile" type="CoalAttributePercentage" minOccurs="0"/>
  </xsd:sequence>
</xsd:group>
```

**Model Group: CommodityCoalProperties.model**

Name	CommodityCoalProperties.model
Used by (from the same schema document)	Complex Type <b>CoalStandardQuality</b>

**Documentation**

Items defining the physical attributes of the coal product.

**XML Instance Representation**

&lt;BTUpperLB&gt; CoalAttributeDecimal &lt;/BTUpperLB&gt; [0..1]

'The number of British Thermal Units per Pound of the coal product.'

&lt;topSize&gt; CoalAttributeDecimal &lt;/topSize&gt; [0..1]

'The smallest sieve opening that will result in less than 5% of a sample of the coal product remaining.'

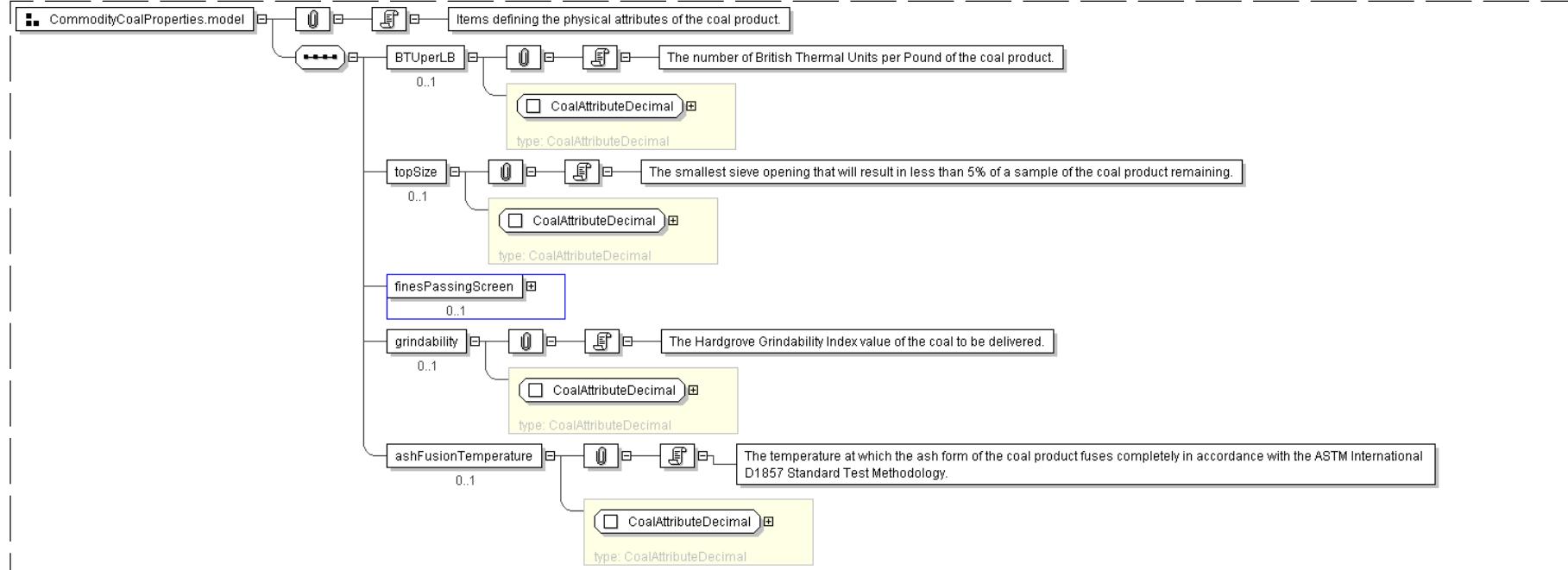
&lt;finesPassingScreen&gt; CoalAttributeDecimal &lt;/finesPassingScreen&gt; [0..1]

&lt;grindability&gt; CoalAttributeDecimal &lt;/grindability&gt; [0..1]

'The Hardgrove Grindability Index value of the coal to be delivered.'

&lt;ashFusionTemperature&gt; CoalAttributeDecimal &lt;/ashFusionTemperature&gt; [0..1]

'The temperature at which the ash form of the coal product fuses completely in accordance with the ASTM International D1857 Standard Test Methodology.'

**Diagram****Schema Component Representation**

```

<xsd:group name="CommodityCoalProperties.model">
  <xsd:sequence>
    <xsd:element name="BTUpperLB" type="CoalAttributeDecimal" minOccurs="0"/>
    <xsd:element name="topSize" type="CoalAttributeDecimal" minOccurs="0"/>
    <xsd:element name="finesPassingScreen" type="CoalAttributeDecimal" minOccurs="0"/>
    <xsd:element name="grindability" type="CoalAttributeDecimal" minOccurs="0"/>
    <xsd:element name="ashFusionTemperature" type="CoalAttributeDecimal" minOccurs="0"/>
  </xsd:sequence>
</xsd:group>

```

top

**Model Group: CommodityCoalReducingAtmosphere.model**

Name	CommodityCoalReducingAtmosphere.model
Used by (from the same schema document)	Complex Type <a href="#">CoalStandardQuality</a>

**Documentation**

Items defining the attributes of the coal product determined by ash fusion tests.

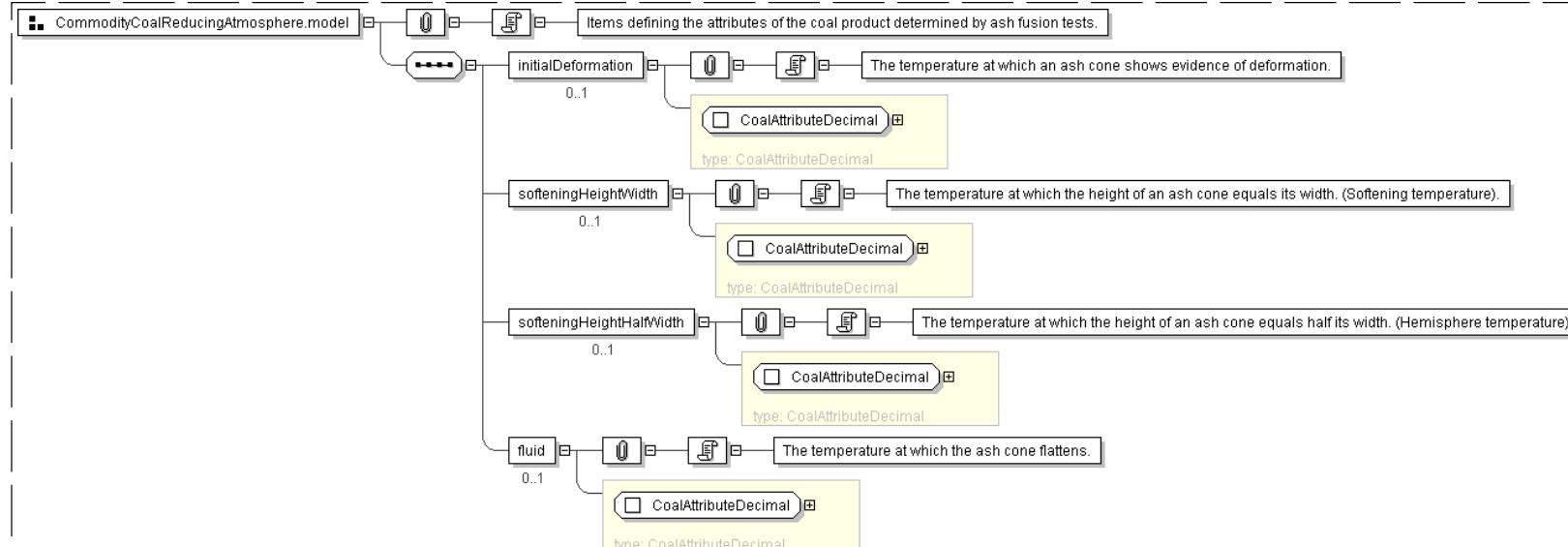
**XML Instance Representation**

```
<initialDeformation> CoalAttributeDecimal </initialDeformation> [0..1]
'The temperature at which an ash cone shows evidence of deformation.'

<softeningHeightWidth> CoalAttributeDecimal </softeningHeightWidth> [0..1]
'The temperature at which the height of an ash cone equals its width. (Softening temperature).'

<softeningHeightHalfWidth> CoalAttributeDecimal </softeningHeightHalfWidth> [0..1]
'The temperature at which the height of an ash cone equals half its width.
(Hemisphere temperature).'

<fluid> CoalAttributeDecimal </fluid> [0..1]
'The temperature at which the ash cone flattens.'
```

**Diagram****Schema Component Representation**

```
<xsd:group name="CommodityCoalReducingAtmosphere.model">
  <xsd:sequence>
    <xsd:element name="initialDeformation" type="CoalAttributeDecimal" minOccurs="0"/>
    <xsd:element name="softeningHeightWidth" type="CoalAttributeDecimal" minOccurs="0"/>
    <xsd:element name="softeningHeightHalfWidth" type="CoalAttributeDecimal" minOccurs="0"/>
    <xsd:element name="fluid" type="CoalAttributeDecimal" minOccurs="0"/>
  </xsd:sequence>
</xsd:group>
```

top

**Model Group: CommodityContent.model**

<b>Name</b>	CommodityContent.model
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">CommodityForward</a> , Complex Type <a href="#">CommodityOption</a> , Complex Type <a href="#">CommoditySwap</a>
<b>Documentation</b>	Items common to all Commodity Transactions.

**XML Instance Representation**

```
<commonPricing> xsd:boolean </commonPricing> [0..1]
'Common pricing may be relevant for a Transaction that references more than one
Commodity Reference Price. If Common Pricing is not specified as applicable, it will be
deemed not to apply.'
```

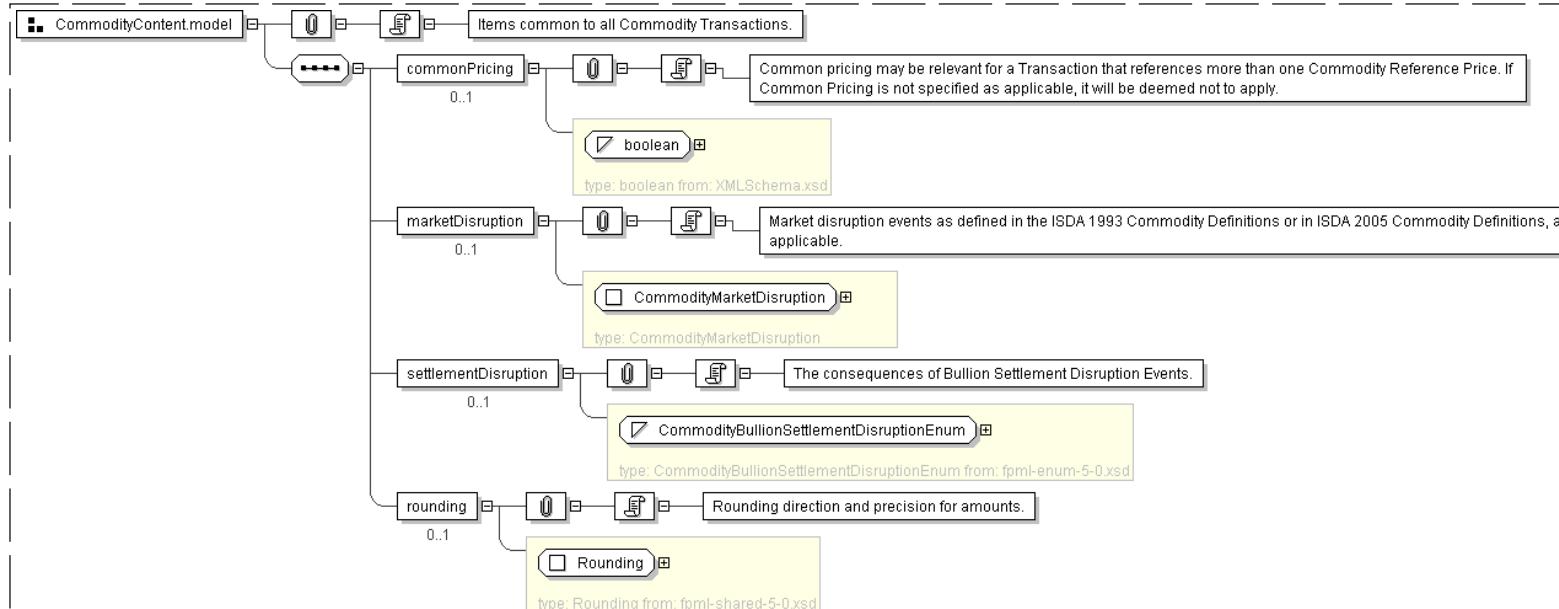
```

<marketDisruption> CommodityMarketDisruption </marketDisruption> [0..1]
'Market disruption events as defined in the ISDA 1993 Commodity Definitions or in ISDA
2005 Commodity Definitions, as applicable.'

<settlementDisruption> CommodityBullionSettlementDisruptionEnum </settlementDisruption> [0..1]
'The consequences of Bullion Settlement Disruption Events.'

<rounding> Rounding </rounding> [0..1]
'Rounding direction and precision for amounts.'

```

**Diagram****Schema Component Representation**

```

<xsd:group name="CommodityContent.model">
  <xsd:sequence>
    <xsd:element name="commonPricing" type="xsd:boolean" minOccurs="0"/>
    <xsd:element name="marketDisruption" type="CommodityMarketDisruption" minOccurs="0"/>
    <xsd:element name="settlementDisruption" type="CommodityBullionSettlementDisruptionEnum"
      " minOccurs="0"/>
    <xsd:element name="rounding" type="Rounding" minOccurs="0"/>
  </xsd:sequence>
</xsd:group>

```

top

**Model Group: CommodityDeliveryPeriodsPointer.model**

<b>Name</b>	CommodityDeliveryPeriodsPointer.model
<b>Used by (from the same schema document)</b>	Complex Type <b>CoalStandardQualitySchedule</b> , Complex Type <b>CommodityPhysicalQuantitySchedule</b> , Complex Type <b>SettlementPeriodsSchedule</b>
<b>Documentation</b>	Model group to enable users to reference a Delivery Periods schedule in the form of a series of actual dates in a deliveryPeriods container or in the form of a parameterised schedule in a deliveryPeriodsSchedule container.

**XML Instance Representation**

```

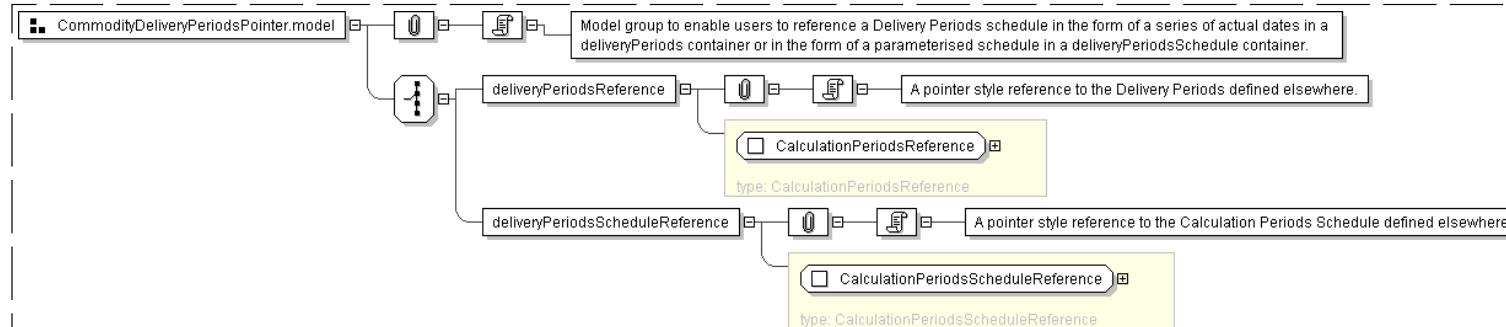
Start Choice [1]
<deliveryPeriodsReference> CalculationPeriodsReference </deliveryPeriodsReference> [1]
'A pointer style reference to the Delivery Periods defined elsewhere.'

<deliveryPeriodsScheduleReference> CalculationPeriodsScheduleReference

```

```
</deliveryPeriodsScheduleReference> [1]
'A pointer style reference to the Calculation Periods Schedule defined elsewhere.'
```

End Choice

**Diagram****Schema Component Representation**

```
<xsd:group name="CommodityDeliveryPeriodsPointer.model">
  <xsd:choice>
    <xsd:element name="deliveryPeriodsReference" type=" CalculationPeriodsReference " />
    <xsd:element name="deliveryPeriodsScheduleReference" type=" CalculationPeriodsScheduleReference " />
  </xsd:choice>
</xsd:group>
```

top

**Model Group: CommodityDeliveryPoints.model**

<b>Name</b>	CommodityDeliveryPoints.model
<b>Used by (from the same schema document)</b>	Complex Type <b>GasDelivery</b>
<b>Documentation</b>	A Delivery Point, applicable to physically settled commodity transactions.

**XML Instance Representation**

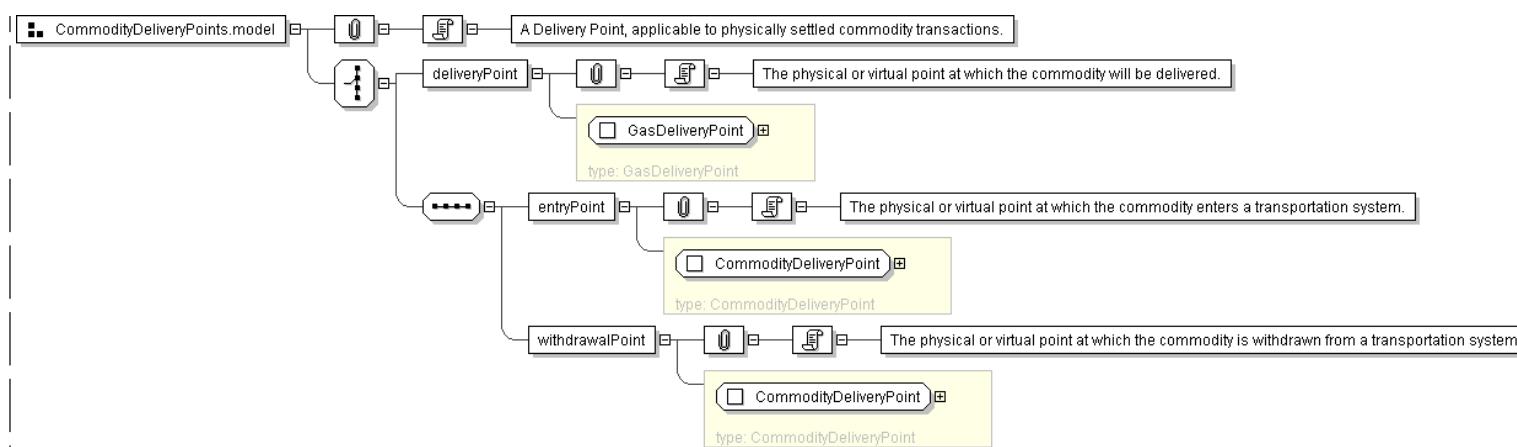
```
Start Choice [1]
<deliveryPoint> GasDeliveryPoint </deliveryPoint> [1]
'The physical or virtual point at which the commodity will be delivered.'

<entryPoint> CommodityDeliveryPoint </entryPoint> [1]
'The physical or virtual point at which the commodity enters a transportation system.'

<withdrawalPoint> CommodityDeliveryPoint </withdrawalPoint> [1]
'The physical or virtual point at which the commodity is withdrawn from a
transportation system.'
```

End Choice

**Diagram**

**Schema Component Representation**

```

<xsd:group name="CommodityDeliveryPoints.model">
  <xsd:choice>
    <xsd:element name="deliveryPoint" type=" GasDeliveryPoint " />
    <xsd:sequence>
      <xsd:element name="entryPoint" type=" CommodityDeliveryPoint " />
      <xsd:element name="withdrawalPoint" type=" CommodityDeliveryPoint " />
    </xsd:sequence>
  </xsd:choice>
</xsd:group>
  
```

top

**Model Group: CommodityFinancialOption.model**

Name	CommodityFinancialOption.model
Used by (from the same schema document)	Complex Type <a href="#">CommodityOption</a>
Documentation	Items specific to financially-settled commodity options.

**XML Instance Representation**

```

<commodity> Commodity </commodity> [1]
'Specifies the underlying component. At the time of the initial schema design, only
underlyers of type Commodity are supported; the choice group in the future could offer
the possibility of adding other types later.'

Start Group: CommodityAsian.model [0..1]
'A group containing properties specific to Asian options.'

<effectiveDate> AdjustableOrRelativeDate </effectiveDate> [1]
'The effective date of the Commodity Option Transaction. Note that the Termination/
Expiration Date should be specified in expirationDate within the CommodityAmericanExercise
type or the CommodityEuropeanExercise type, as applicable.'

Start Choice [1]
  <calculationPeriodsSchedule> CommodityCalculationPeriodsSchedule </
  calculationPeriodsSchedule> [1]
  'A parametric representation of the Calculation Periods of the Commodity Option Transaction.'

  <calculationPeriods> AdjustableDates </calculationPeriods> [1]
  'An absolute representation of the Calculation Period start dates of the Commodity
  Option Transaction.'

End Choice
<pricingDates> CommodityPricingDates </pricingDates> [1]
'The dates on which the option will price.'

<averagingMethod> AveragingMethodEnum </averagingMethod> [0..1]
  
```

'The Method of Averaging if there is more than one Pricing Date.'

End Group: [CommodityAsian.model](#)

Start [Choice](#) [1]

Start [Choice](#) [1]

<notionalQuantitySchedule> [CommodityNotionalQuantitySchedule](#) </notionalQuantitySchedule> [1]

'Allows the documentation of a shaped notional trade where the notional changes over the life of the transaction.'

<notionalQuantity> [CommodityNotionalQuantity](#) </notionalQuantity> [1]

'The Notional Quantity.'

<settlementPeriodsNotionalQuantity> [CommoditySettlementPeriodsNotionalQuantity](#)

</settlementPeriodsNotionalQuantity> [1..\*]

'For an electricity transaction, the Notional Quantity for a one or more groups of Settlement Periods to which the Notional Quantity is based. If the schedule differs for different groups of Settlement Periods, this element should be repeated.'

End Choice

<totalNotionalQuantity> [xsd:decimal](#) </totalNotionalQuantity> [0..1]

'The Total Notional Quantity.'

<quantityReference> [QuantityReference](#) </quantityReference> [1]

'A pointer style reference to a quantity defined on another leg.'

End Choice

<exercise> [CommodityExercise](#) </exercise> [1]

'The parameters for defining how the commodity option can be exercised and how it is settled.'

Start [Choice](#) [1]

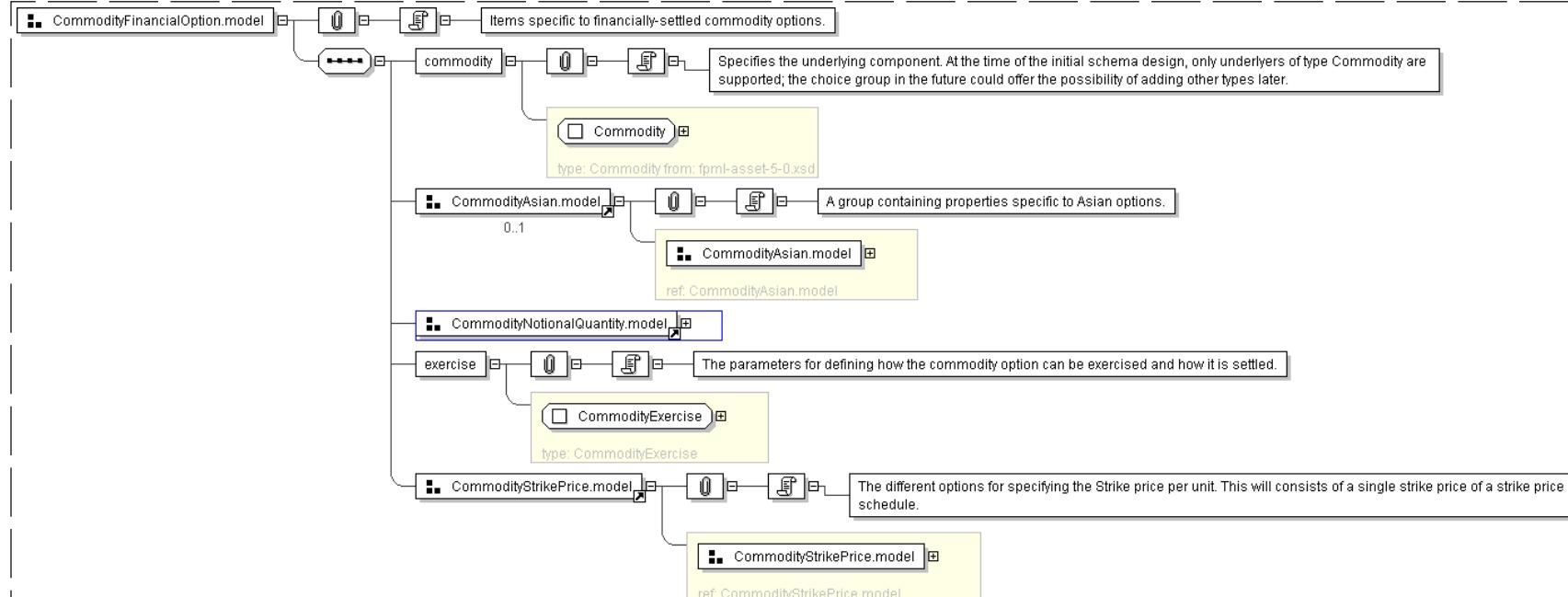
<strikePricePerUnit> [NonNegativeMoney](#) </strikePricePerUnit> [1]

'The currency amount of the strike price per unit.'

<strikePricePerUnitSchedule> [CommodityStrikeSchedule](#) </strikePricePerUnitSchedule> [1]

End Choice

#### Diagram



#### Schema Component Representation

```
<xsd:group name="CommodityFinancialOption.model">
```

```

<xsd:sequence>
  <xsd:element name="commodity" type=" Commodity "/>
  <xsd:group ref=" CommodityAsian.model " minOccurs="0"/>
  <xsd:group ref=" CommodityNotionalQuantity.model "/>
  <xsd:element name="exercise" type=" CommodityExercise "/>
  <xsd:group ref=" CommodityStrikePrice.model "/>
</xsd:sequence>
</xsd:group>

```

top

**Model Group: CommodityFixedPhysicalQuantity.model**

<b>Name</b>	CommodityFixedPhysicalQuantity.model
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">BullionPhysicalLeg</a> , Complex Type <a href="#">CommodityPhysicalQuantity</a> , Complex Type <a href="#">GasPhysicalQuantity</a>
<b>Documentation</b>	The different options for specifying a fixed physical quantity of commodity to be delivered.

**XML Instance Representation**

Start Choice [1]

<physicalQuantity> [CommodityNotionalQuantity](#) </physicalQuantity> [1]

'The Quantity per Delivery Period.'

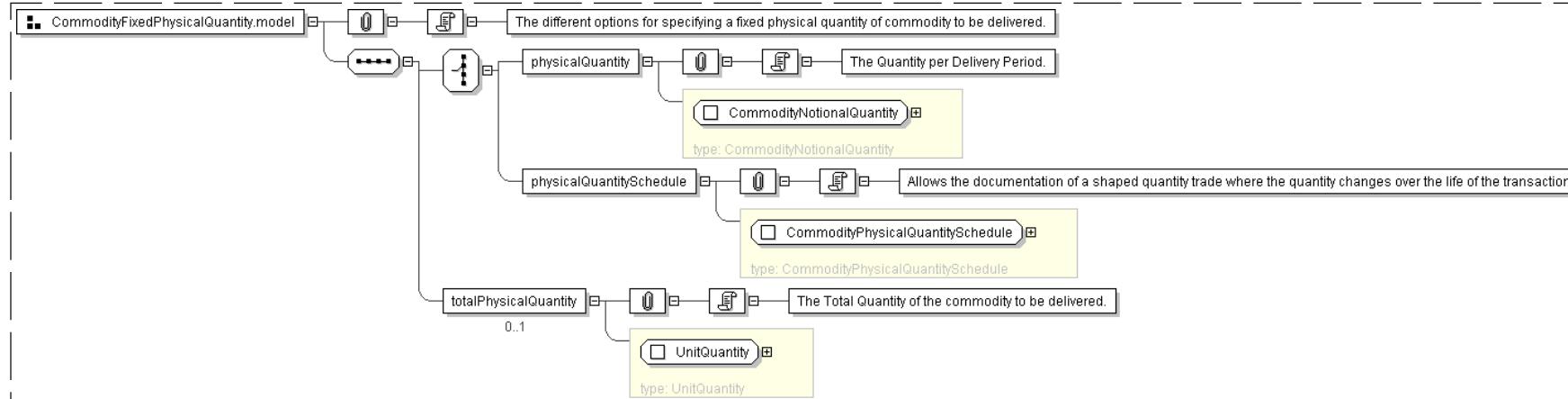
<physicalQuantitySchedule> [CommodityPhysicalQuantitySchedule](#) </physicalQuantitySchedule> [1]

'Allows the documentation of a shaped quantity trade where the quantity changes over the life of the transaction.'

End Choice

<totalPhysicalQuantity> [UnitQuantity](#) </totalPhysicalQuantity> [0..1]

'The Total Quantity of the commodity to be delivered.'

**Diagram****Schema Component Representation**

```

<xsd:group name="CommodityFixedPhysicalQuantity.model">
  <xsd:sequence>
    <xsd:choice>
      <xsd:element name="physicalQuantity" type=" CommodityNotionalQuantity "/>
      <xsd:element name="physicalQuantitySchedule" type=" CommodityPhysicalQuantitySchedule "/>
    </xsd:choice>
    <xsd:element name="totalPhysicalQuantity" type=" UnitQuantity " minOccurs="0"/>
  </xsd:sequence>
</xsd:group>

```

top

**Model Group: CommodityFixedPrice.model**

<b>Name</b>	CommodityFixedPrice.model
<b>Used by (from the same schema document)</b>	Complex Type <b>FixedPriceLeg</b>
<b>Documentation</b>	The different options for specifying the Fixed Price.

**XML Instance Representation**

```

Start Choice [1]
<fixedPriceSchedule> CommodityFixedPriceSchedule </fixedPriceSchedule> [1]
'Allows the specification of a Fixed Price that varies over the life of the trade.'

Start Choice [1]
<fixedPrice> FixedPrice </fixedPrice> [1]
'Fixed price on which fixed payments are based.'

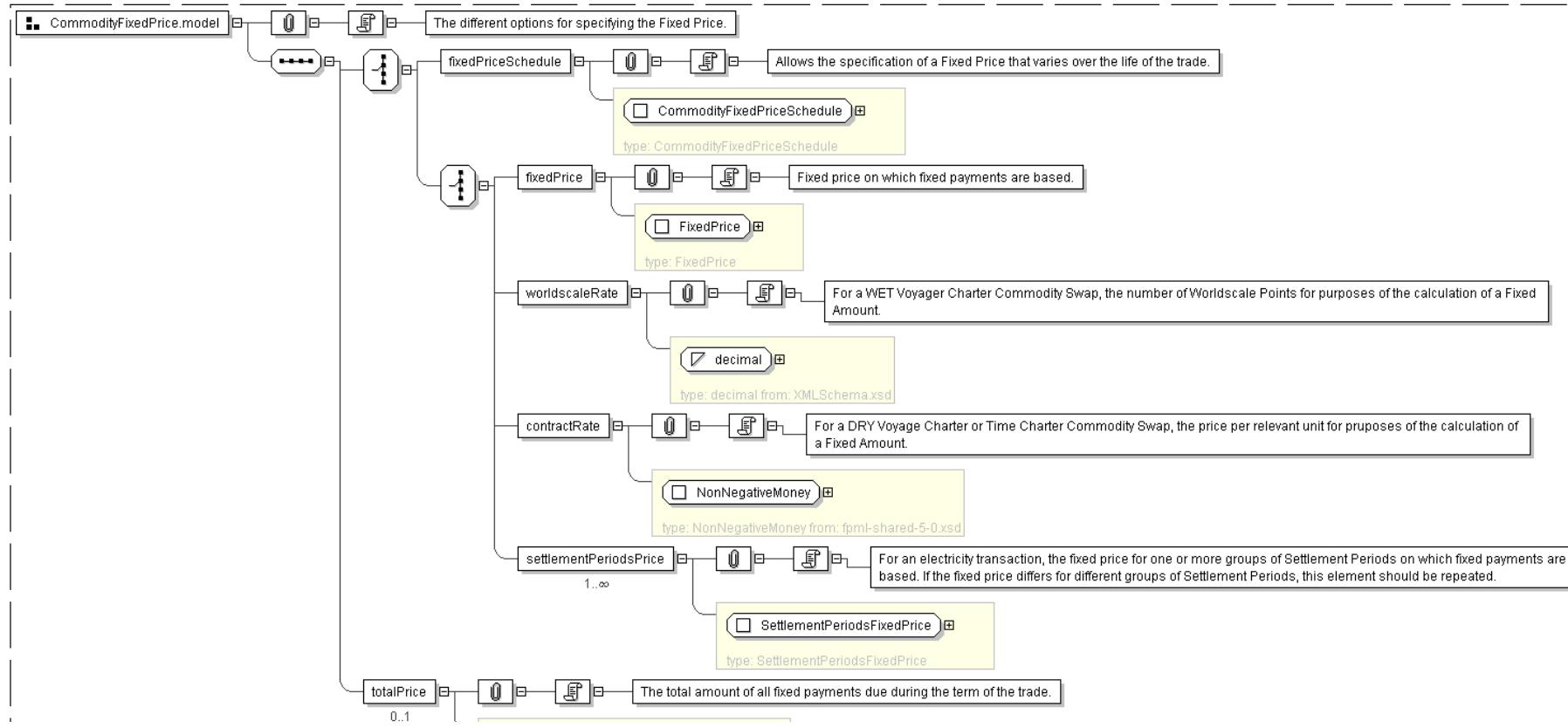
<worldscaleRate> xsd:decimal </worldscaleRate> [1]
'For a WET Voyager Charter Commodity Swap, the number of Worldscale Points for purposes of
the calculation of a Fixed Amount.'

<contractRate> NonNegativeMoney </contractRate> [1]
'For a DRY Voyage Charter or Time Charter Commodity Swap, the price per relevant unit
for purposes of the calculation of a Fixed Amount.'

<settlementPeriodsPrice> SettlementPeriodsFixedPrice </settlementPeriodsPrice> [1..*]
'For an electricity transaction, the fixed price for one or more groups of Settlement
Periods on which fixed payments are based. If the fixed price differs for different groups
of Settlement Periods, this element should be repeated.'

End Choice
End Choice
<totalPrice> NonNegativeMoney </totalPrice> [0..1]
'The total amount of all fixed payments due during the term of the trade.'

```

**Diagram**



## Schema Component Representation

```
<xsd:group name="CommodityFixedPrice.model">
  <xsd:sequence>
    <xsd:choice>
      <xsd:element name="fixedPriceSchedule" type=" CommodityFixedPriceSchedule " />
      <xsd:choice>
        <xsd:element name="fixedPrice" type=" FixedPrice " />
        <xsd:element name="worldscaleRate" type=" xsd:decimal " />
        <xsd:element name="contractRate" type=" NonNegativeMoney " />
        <xsd:element name="settlementPeriodsPrice" type=" SettlementPeriodsFixedPrice "
          " maxOccurs="unbounded" />
      </xsd:choice>
    </xsd:choice>
    <xsd:element name="totalPrice" type=" NonNegativeMoney " minOccurs="0 " />
  </xsd:sequence>
</xsd:group>
```

top

## Model Group: CommodityFreightFlatRate.model

Name	CommodityFreightFlatRate.model
Used by (from the same schema document)	Complex Type <b>FixedPriceLeg</b> , Complex Type <b>FloatingPriceLeg</b>
Documentation	The Flat Rate, applicable to Wet Voyager Charter Freight Swaps.

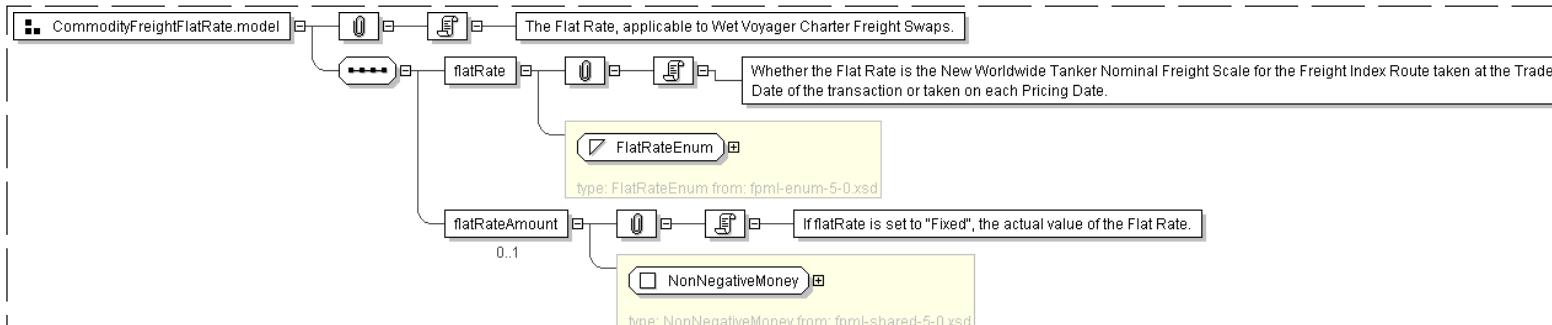
## XML Instance Representation

```
<flatRate> FlatRateEnum </flatRate> [1]
'Whether the Flat Rate is the New Worldwide Tanker Nominal Freight Scale for the Freight
Index Route taken at the Trade Date of the transaction or taken on each Pricing Date.'
```

```
<flatRateAmount> NonNegativeMoney </flatRateAmount> [0..1]
'If flatRate is set to "Fixed", the actual value of the Flat Rate.'
```

## Diagram



## Schema Component Representation

```
<xsd:group name="CommodityFreightFlatRate.model">
  <xsd:sequence>
    <xsd:element name="flatRate" type=" FlatRateEnum " />
    <xsd:element name="flatRateAmount" type=" NonNegativeMoney " minOccurs="0 " />
  </xsd:sequence>
</xsd:group>
```

top

## Model Group: CommodityNonPeriodicPaymentDates.model

<b>Name</b>	CommodityNonPeriodicPaymentDates.model
<b>Used by (from the same schema document)</b>	Model Group <b>CommodityPaymentDates.model</b>
<b>Documentation</b>	The different options for specifying the Payment Date.

**XML Instance Representation**

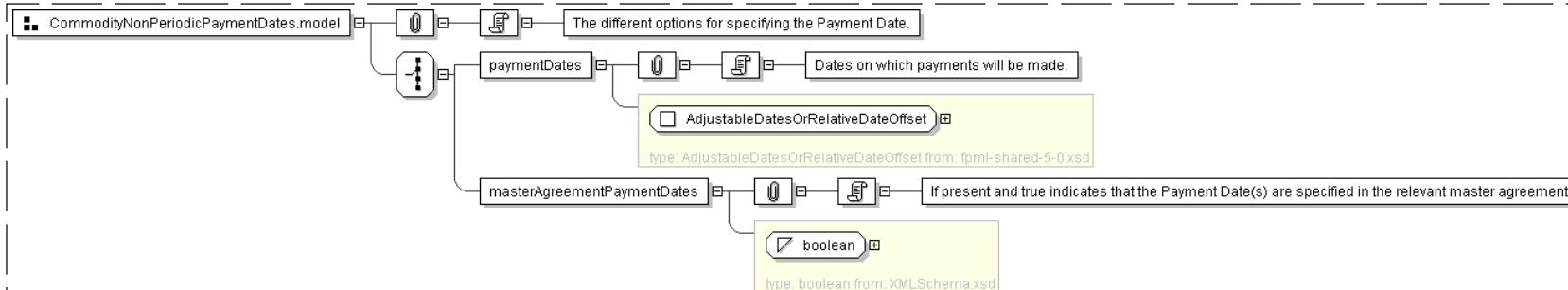
```

Start Choice [1]
<paymentDates> AdjustableDatesOrRelativeDateOffset </paymentDates> [1]
'Dates on which payments will be made.'

<masterAgreementPaymentDates> xsd:boolean </masterAgreementPaymentDates> [1]
'If present and true indicates that the Payment Date(s) are specified in the relevant
master agreement.'

End Choice

```

**Diagram****Schema Component Representation**

```

<xsd:group name="CommodityNonPeriodicPaymentDates.model">
  <xsd:choice>
    <xsd:element name="paymentDates" type="#AdjustableDatesOrRelativeDateOffset" />
    <xsd:element name="masterAgreementPaymentDates" type="xsd:boolean" />
  </xsd:choice>
</xsd:group>

```

top

**Model Group: CommodityNotionalQuantity.model**

<b>Name</b>	CommodityNotionalQuantity.model
<b>Used by (from the same schema document)</b>	Complex Type <b>FixedPriceLeg</b> , Complex Type <b>FloatingPriceLeg</b> , Model Group <b>CommodityFinancialOption.model</b>
<b>Documentation</b>	The different options for specifying the Notional Quantity. A flat notional for the term of the trade may be specified, or else the Notional Quantity per Calculation Period. In the latter case, there must be a notional quantity specified for each Calculation Period, regardless of whether the Notional Quantity changes or remains the same between periods.

**XML Instance Representation**

```

Start Choice [1]
Start Choice [1]
<notionalQuantitySchedule> CommodityNotionalQuantitySchedule </notionalQuantitySchedule> [1]
'Allows the documentation of a shaped notional trade where the notional changes over the
life of the transaction.'

<notionalQuantity> CommodityNotionalQuantity </notionalQuantity> [1]
'The Notional Quantity.'

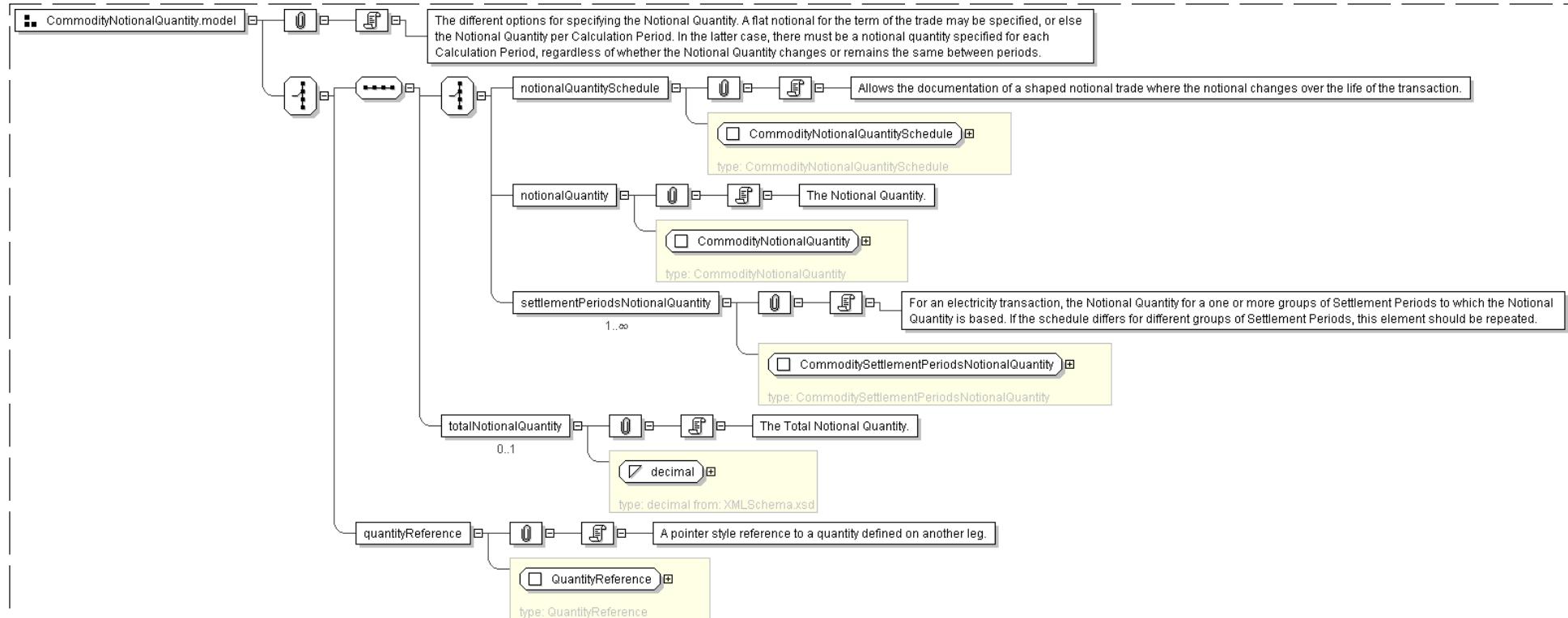
<settlementPeriodsNotionalQuantity> CommoditySettlementPeriodsNotionalQuantity
</settlementPeriodsNotionalQuantity> [1..*]
'For an electricity transaction, the Notional Quantity for one or more groups of
Settlement Periods to which the Notional Quantity is based. If the schedule differs
for different groups of Settlement Periods, this element should be repeated.'

End Choice
<totalNotionalQuantity> xsd:decimal </totalNotionalQuantity> [0..1]
'The Total Notional Quantity.'

```

```
<quantityReference> QuantityReference </quantityReference> [1]
'A pointer style reference to a quantity defined on another leg.'
```

End Choice

**Diagram****Schema Component Representation**

```
<xsd:group name="CommodityNotionalQuantity.model">
  <xsd:choice>
    <xsd:sequence>
      <xsd:choice>
        <xsd:element name="notionalQuantitySchedule" type=" CommodityNotionalQuantitySchedule " />
        <xsd:element name="notionalQuantity" type=" CommodityNotionalQuantity " />
        <xsd:element name="settlementPeriodsNotionalQuantity"
          type=" CommoditySettlementPeriodsNotionalQuantity " maxOccurs="unbounded" />
      </xsd:choice>
      <xsd:element name="totalNotionalQuantity" type=" xsd:decimal " minOccurs="0" />
    </xsd:sequence>
    <xsd:element name="quantityReference" type=" QuantityReference " />
  </xsd:choice>
</xsd:group>
```

top

**Model Group: CommodityPaymentDates.model**

<b>Name</b>	CommodityPaymentDates.model
<b>Used by (from the same schema document)</b>	Complex Type <code>CommodityExercise</code> , Complex Type <code>FixedPriceLeg</code> , Complex Type <code>FloatingPriceLeg</code> , Complex Type <code>NonPeriodicFixedPriceLeg</code>
<b>Documentation</b>	The different options for specifying the Payment Date. This will consist of either a set of Payment Dates or else a Payment Date schedule.

**XML Instance Representation**

Start **Choice** [1]  
**<relativePaymentDates>** **CommodityRelativePaymentDates** **</relativePaymentDates>** [1]

'The Payment Dates of the trade relative to the Calculation Periods.'

Start **Choice** [1]  
**<paymentDates>** **AdjustableDatesOrRelativeDateOffset** **</paymentDates>** [1]

'Dates on which payments will be made.'

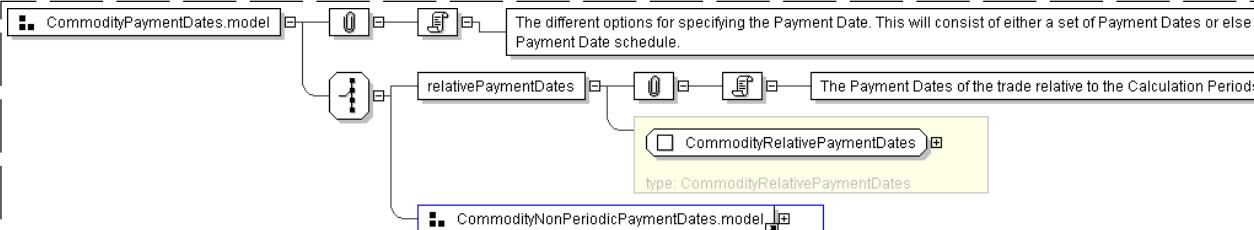
**<masterAgreementPaymentDates>** **xsd:boolean** **</masterAgreementPaymentDates>** [1]

'If present and true indicates that the Payment Date(s) are specified in the relevant master agreement.'

End Choice

End Choice

#### Diagram



#### Schema Component Representation

```

<xsd:group name="CommodityPaymentDates.model">
  <xsd:choice>
    <xsd:element name="relativePaymentDates" type="CommodityRelativePaymentDates" />
    <xsd:group ref="CommodityNonPeriodicPaymentDates.model" />
  </xsd:choice>
</xsd:group>

```

top

#### Model Group: CommodityPhysicalOption.model

Name	CommodityPhysicalOption.model
------	-------------------------------

Used by (from the same schema document)	Complex Type <b>CommodityOption</b>
---	-------------------------------------

Documentation	Items specific to financially-settled commodity options.
---------------	--

#### XML Instance Representation

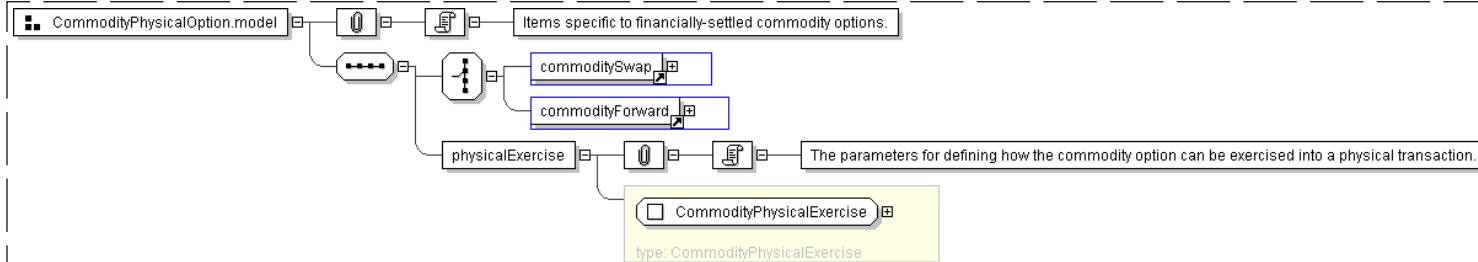
Start **Choice** [1]  
**<commoditySwap>** ... **</commoditySwap>** [1]  
**<commodityForward>** ... **</commodityForward>** [1]

End Choice

**<physicalExercise>** **CommodityPhysicalExercise** **</physicalExercise>** [1]

'The parameters for defining how the commodity option can be exercised into a physical transaction.'

#### Diagram



**Schema Component Representation**

```
<xsd:group name="CommodityPhysicalOption.model">
  <xsd:sequence>
    <xsd:choice>
      <xsd:element ref="# commoditySwap "# />
      <xsd:element ref="# commodityForward "# />
    </xsd:choice>
    <xsd:element name="physicalExercise" type="# CommodityPhysicalExercise "# />
  </xsd:sequence>
</xsd:group>
```

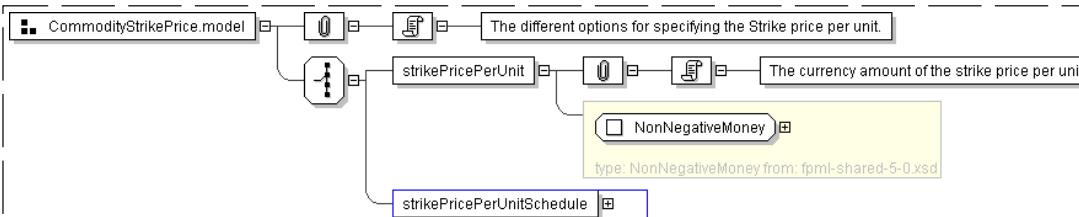
[top](#)**Model Group: CommodityStrikePrice.model**

Name	CommodityStrikePrice.model
Used by (from the same schema document)	Model Group <b>CommodityFinancialOption.model</b>
Documentation	The different options for specifying the Strike price per unit.

**XML Instance Representation**

```
Start Choice [1]
  <strikePricePerUnit> NonNegativeMoney </strikePricePerUnit> [1]
  'The currency amount of the strike price per unit..'

  <strikePricePerUnitSchedule> CommodityStrikeSchedule </strikePricePerUnitSchedule> [1]
End Choice
```

**Diagram****Schema Component Representation**

```
<xsd:group name="CommodityStrikePrice.model">
  <xsd:choice>
    <xsd:element name="strikePricePerUnit" type="# NonNegativeMoney "# />
    <xsd:element name="strikePricePerUnitschedule" type="# CommodityStrikeSchedule "# />
  </xsd:choice>
</xsd:group>
```

[top](#)**Model Group: CommodityUSCoalDelivery.model**

Name	CommodityUSCoalDelivery.model
Used by (from the same schema document)	Complex Type <b>CoalDelivery</b>
Documentation	Items specific to the definition of the delivery of a US Coal Product.

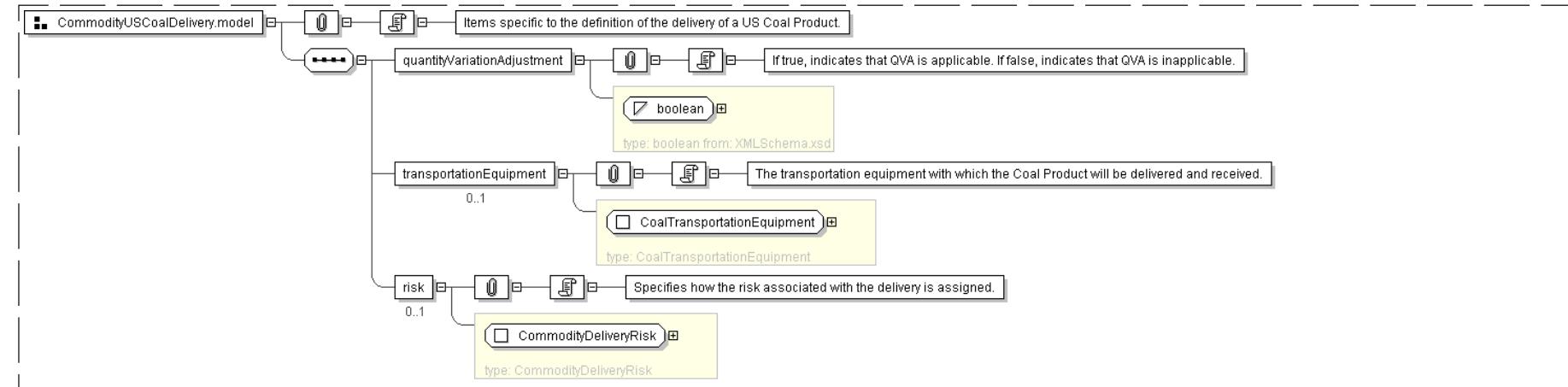
**XML Instance Representation**

```
<quantityVariationAdjustment> xsd:boolean </quantityVariationAdjustment> [1]
'If true, indicates that QVA is applicable. If false, indicates that QVA is inapplicable.'

<transportationEquipment> CoalTransportationEquipment </transportationEquipment> [0..1]
'The transportation equipment with which the Coal Product will be delivered and received.'

<risk> CommodityDeliveryRisk </risk> [0..1]
'Specifies how the risk associated with the delivery is assigned.'
```

**Diagram**

**Schema Component Representation**

```
<xsd:group name="CommodityUSCoalDelivery.model">
  <xsd:sequence>
    <xsd:element name="quantityVariationAdjustment" type="xsd:boolean" />
    <xsd:element name="transportationEquipment" type="CoalTransportationEquipment" minOccurs="0"/>
    <xsd:element name="risk" type="CommodityDeliveryRisk" minOccurs="0"/>
  </xsd:sequence>
</xsd:group>
```

top

**Model Group: CommodityUSCoalProduct.model**

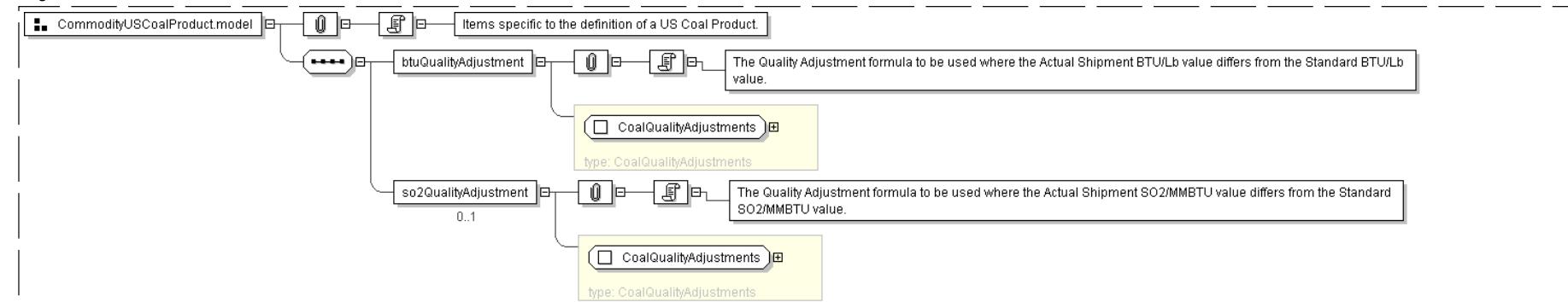
Name	CommodityUSCoalProduct.model
Used by (from the same schema document)	Complex Type <b>CoalProduct</b>
Documentation	Items specific to the definition of a US Coal Product.

**XML Instance Representation**

```
<btuQualityAdjustment> CoalQualityAdjustments </btuQualityAdjustment> [1]
'The Quality Adjustment formula to be used where the Actual Shipment BTU/Lb value differs
from the Standard BTU/Lb value.'
```

```
<so2QualityAdjustment> CoalQualityAdjustments </so2QualityAdjustment> [0..1]
'The Quality Adjustment formula to be used where the Actual Shipment SO2/MMBTU value
differs from the Standard SO2/MMBTU value.'
```

**Diagram****Schema Component Representation**

```
<xsd:group name="CommodityUSCoalProduct.model">
```

```

<xsd:sequence>
  <xsd:element name="btuQualityAdjustment" type=" CoalQualityAdjustments " />
  <xsd:element name="so2QualityAdjustment" type=" CoalQualityAdjustments " minOccurs="0" />
</xsd:sequence>
</xsd:group>

```

[top](#)

## Model Group: LagOrReference.model

Name	LagOrReference.model
Used by (from the same schema document)	Complex Type <a href="#">CommodityFx</a>
Documentation	Allows a Lag or a LagReference to be specified.

## XML Instance Representation

```

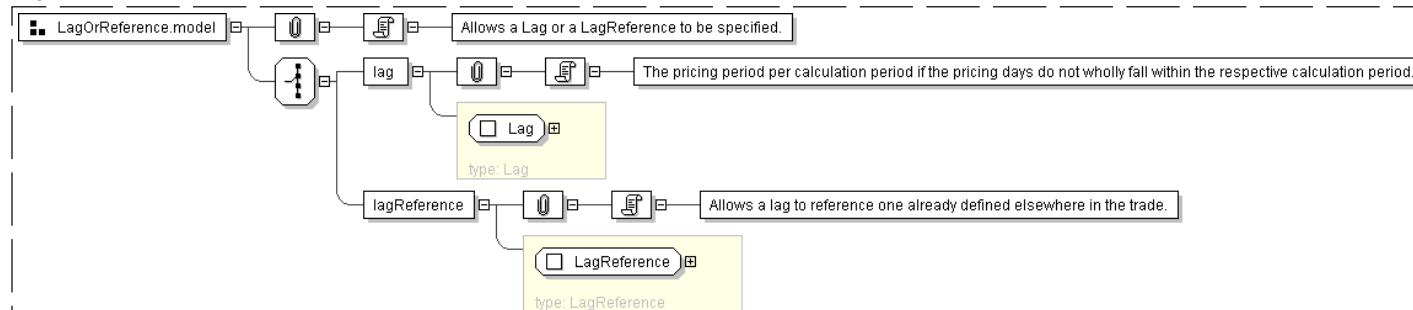
Start Choice [1]
<lag> Lag </lag> [1]
'The pricing period per calculation period if the pricing days do not wholly fall within
the respective calculation period.'

<lagReference> LagReference </lagReference> [1]
'Allows a lag to reference one already defined elsewhere in the trade.'

```

End Choice

## Diagram



## Schema Component Representation

```

<xsd:group name="LagOrReference.model">
  <xsd:choice>
    <xsd:element name="lag" type=" Lag " />
    <xsd:element name="lagReference" type=" LagReference " />
  </xsd:choice>
</xsd:group>

```

[top](#)

## Model Group: Price.model

Name	Price.model
Used by (from the same schema document)	Complex Type <a href="#">FixedPrice</a>
Documentation	Price model group.

## XML Instance Representation

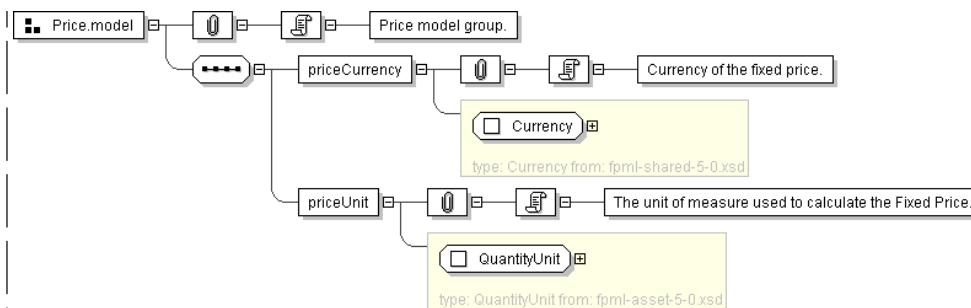
```

<priceCurrency> Currency </priceCurrency> [1]
'Currency of the fixed price.'

<priceUnit> QuantityUnit </priceUnit> [1]
'The unit of measure used to calculate the Fixed Price.'

```

## Diagram

**Schema Component Representation**

```

<xsd:group name="Price.model">
  <xsd:sequence>
    <xsd:element name="priceCurrency" type="Currency" />
    <xsd:element name="priceUnit" type="QuantityUnit" />
  </xsd:sequence>
</xsd:group>
  
```

[top](#)**Model Group: PricingDays.model**

<b>Name</b>	PricingDays.model
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">CommodityFx</a> , Complex Type <a href="#">CommodityPricingDates</a>
<b>Documentation</b>	The different options for specifying which days are pricing days within a pricing period. Unless a lag element is present, the pricing period will be the calculation period.

**XML Instance Representation**

```

<dayType> CommodityDayTypeEnum </dayType> [1]
'The type of day on which pricing occurs.'

Start Choice [1]
<dayDistribution> CommodityFrequencyType </dayDistribution> [1]
'The method by which the pricing days are distributed across the pricing period.'

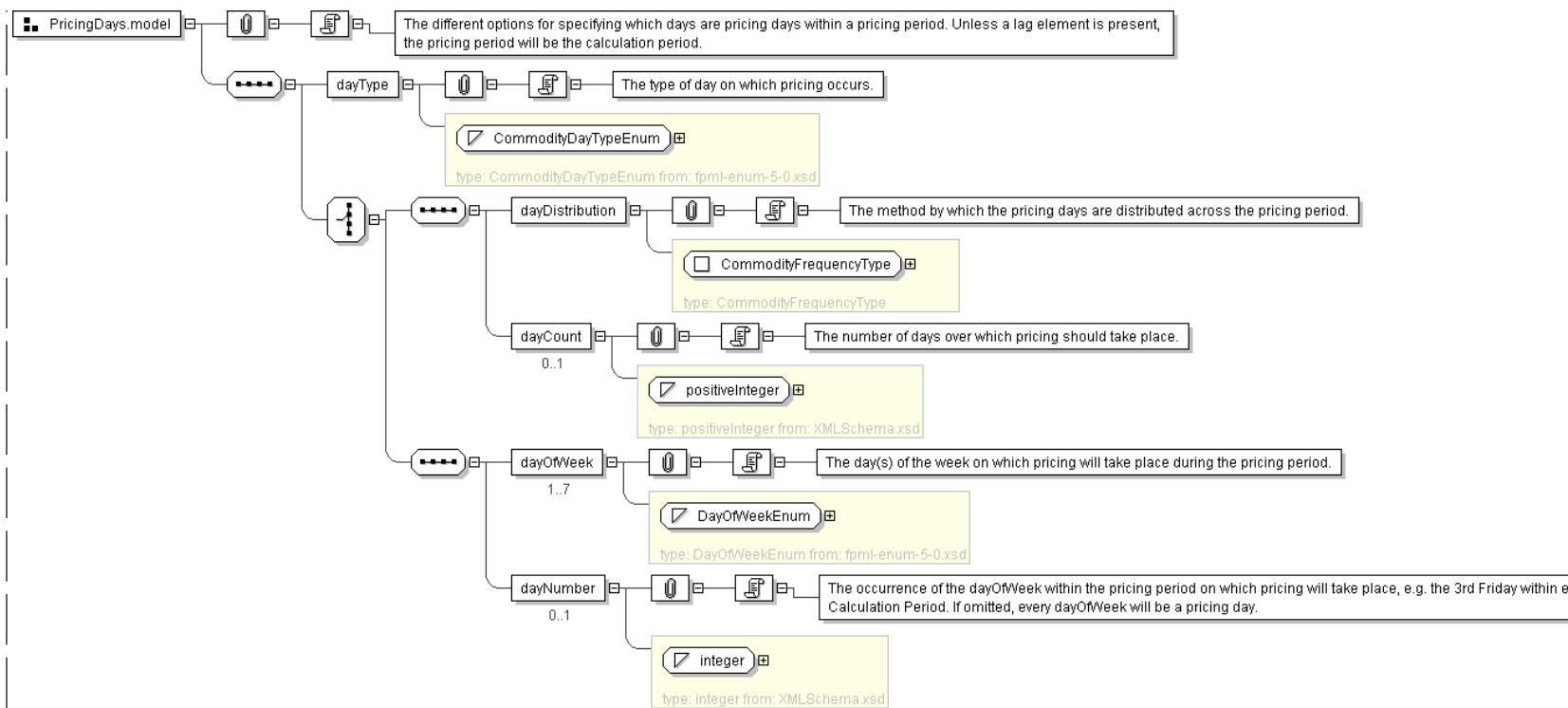
<dayCount> xsd:positiveInteger </dayCount> [0..1]
'The number of days over which pricing should take place.'

<dayOfWeek> DayOfWeekEnum </dayOfWeek> [1..7]
'The day(s) of the week on which pricing will take place during the pricing period.'

<dayNumber> xsd:integer </dayNumber> [0..1]
'The occurrence of the dayOfWeek within the pricing period on which pricing will take place,
e.g. the 3rd Friday within each Calculation Period. If omitted, every dayOfWeek will be
a pricing day.'
  
```

End Choice

**Diagram**



#### Schema Component Representation

```

<xsd:group name="PricingDays.model">
  <xsd:sequence>
    <xsd:element name="dayType" type="CommodityDayTypeEnum" />
    <xsd:choice>
      <xsd:sequence>
        <xsd:element name="dayDistribution" type="CommodityFrequencyType" />
        <xsd:element name="dayCount" type="xsd:positiveInteger" minOccurs="0" />
      </xsd:sequence>
      <xsd:sequence>
        <xsd:element name="dayOfWeek" type="DayOfWeekEnum" maxOccurs="7" />
        <xsd:element name="dayNumber" type="xsd:integer" minOccurs="0" />
      </xsd:sequence>
    </xsd:choice>
  </xsd:sequence>
</xsd:group>

```

top

#### Legend

**Complex Type:** AusAddress  
Schema Component Type

AusAddress  
Schema Component Name

<b>Super-types:</b>	<a href="#">Address</a> < AusAddress (by extension)
<b>Sub-types:</b>	• <a href="#">QLDAddress</a> (by restriction)

If this schema component is a type definition, its type hierarchy is shown in a gray-bordered box.

<b>Name</b>	AusAddress
<b>Abstract</b>	no

The table above displays the properties of this schema component.

**XML Instance Representation**

```
<... country="Australia" >
<unitNo> string </unitNo> [0..1]
<houseNo> string </houseNo> [1]
<street> string </street> [1]
Start Choice [1]
<city> string </city> [1]
<town> string </town> [1]
End Choice
<state> AusStates </state> [1]
<postcode> string <>pattern = [1-9][0-9]{3}</> </postcode> [1]
</...>
```

The XML Instance Representation table above shows the schema component's content as an XML instance.

- The minimum and maximum occurrence of elements and attributes are provided in square brackets, e.g. [0..1].
- Model group information are shown in gray, e.g. Start Choice ... End Choice.
- For type derivations, the elements and attributes that have been added to or changed from the base type's content are shown in **bold**.
- If an element/attribute has a fixed value, the fixed value is shown in green, e.g. country="Australia".
- Otherwise, the type of the element/attribute is displayed.
  - If the element/attribute's type is in the schema, a link is provided to it.
  - For local simple type definitions, the constraints are displayed in angle brackets, e.g. <>pattern = [1-9][0-9]{3}</>.

**Schema Component Representation**

```
<complexType name="AusAddress">
<complexContent>
<extension base=" Address " >
<sequence>
<element name="state" type=" AusStates " />
<element name="postcode">
<simpleType>
<restriction base=" string ">
<pattern value="[1-9][0-9]{3}" />
</restriction>
</simpleType>
</element>
</sequence>
<attribute name="country" type=" string " fixed="Australia" />
</extension>
</complexContent>
</complexType>
```

The Schema Component Representation table above displays the underlying XML representation of the schema component. (Annotations are not shown.)

top

**Glossary**

**Abstract** (Applies to complex type definitions and element declarations). An abstract element or complex type cannot be used to validate an element instance. If there is a reference to an abstract element, only element declarations that can substitute the abstract element can be used to validate the instance. For references to abstract type definitions, only derived types can be used.

**All Model Group** Child elements can be provided *in any order* in instances. See: <http://www.w3.org/TR/xmlschema-1/#element-all>.

**Choice Model Group** *Only one* from the list of child elements and model groups can be provided in instances. See: <http://www.w3.org/TR/xmlschema-1/#element-choice>.

**Collapse Whitespace Policy** Replace tab, line feed, and carriage return characters with space character (Unicode character 32). Then, collapse contiguous sequences of space characters into single space character, and remove leading and trailing space characters.

**Disallowed Substitutions** (Applies to element declarations). If *substitution* is specified, then *substitution group* members cannot be used in place of the given element declaration to validate element instances. If *derivation methods*, e.g. extension, restriction, are specified, then the given element declaration will not validate element instances that have types derived from the element declaration's type using the specified derivation methods. Normally, element instances can override their declaration's type by specifying an *xsi:type* attribute.

**Key Constraint** Like *Uniqueness Constraint*, but additionally requires that the specified value(s) must be provided. See: [http://www.w3.org/TR/xmlschema-1/#identity-constraint\\_Definitions](http://www.w3.org/TR/xmlschema-1/#identity-constraint_Definitions).

**Key Reference Constraint** Ensures that the specified value(s) must match value(s) from a *Key Constraint* or *Uniqueness Constraint*. See: [http://www.w3.org/TR/xmlschema-1/#clidentity-constraint\\_Definitions](http://www.w3.org/TR/xmlschema-1/#clidentity-constraint_Definitions).

**Model Group** Groups together element content, specifying the order in which the element content can occur and the number of times the group of

element content may be repeated. See: [http://www.w3.org/TR/xmlschema-1/#Model\\_Groups](http://www.w3.org/TR/xmlschema-1/#Model_Groups).

**Nillable** (Applies to element declarations). If an element declaration is nillable, instances can use the `xsi:nil` attribute. The `xsi:nil` attribute is the boolean attribute, `nil`, from the `http://www.w3.org/2001/XMLSchema-instance` namespace. If an element instance has an `xsi:nil` attribute set to true, it can be left empty, even though its element declaration may have required content.

**Notation** A notation is used to identify the format of a piece of data. Values of elements and attributes that are of type, NOTATION, must come from the names of declared notations. See: [http://www.w3.org/TR/xmlschema-1/#cNotation\\_Declarations](http://www.w3.org/TR/xmlschema-1/#cNotation_Declarations).

**Preserve Whitespace Policy** Preserve whitespaces exactly as they appear in instances.

**Prohibited Derivations** (Applies to type definitions). Derivation methods that cannot be used to create sub-types from a given type definition.

**Prohibited Substitutions** (Applies to complex type definitions). Prevents sub-types that have been derived using the specified derivation methods from validating element instances in place of the given type definition.

**Replace Whitespace Policy** Replace tab, line feed, and carriage return characters with space character (Unicode character 32).

**Sequence Model Group** Child elements and model groups must be provided *in the specified order* in instances. See: <http://www.w3.org/TR/xmlschema-1/#element-sequence>.

**Substitution Group** Elements that are *members* of a substitution group can be used wherever the `head` element of the substitution group is referenced.

**Substitution Group Exclusions** (Applies to element declarations). Prohibits element declarations from nominating themselves as being able to substitute a given element declaration, if they have types that are derived from the original element's type using the specified derivation methods.

**Target Namespace** The target namespace identifies the namespace that components in this schema belongs to. If no target namespace is provided, then the schema components do not belong to any namespace.

**Uniqueness Constraint** Ensures uniqueness of an element/attribute value, or a combination of values, within a specified scope. See: [http://www.w3.org/TR/xmlschema-1/#cIdentity-constraint\\_Definitions](http://www.w3.org/TR/xmlschema-1/#cIdentity-constraint_Definitions).

[top](#)

---

Generated by [soXygen!](#) using a modified version of [xs3p](#) that adds schema diagrams and chunking support.

# XML Schema Documentation

---

## Table of Contents

- [Schema Document Properties](#)
- [Global Declarations](#)
  - [Element: allocationAcknowledgement](#)
  - [Element: allocationApproved](#)
  - [Element: allocationException](#)
  - [Element: allocationRefused](#)
  - [Element: clearingAcknowledgement](#)
  - [Element: clearingConfirmed](#)
  - [Element: clearingException](#)
  - [Element: clearingRefused](#)
  - [Element: confirmationAcknowledgement](#)
  - [Element: confirmationAgreed](#)
  - [Element: confirmationDisputed](#)
  - [Element: confirmationException](#)
  - [Element: confirmationStatus](#)
  - [Element: consentAcknowledgement](#)
  - [Element: consentException](#)
  - [Element: consentGranted](#)
  - [Element: consentRefused](#)
  - [Element: executionAcknowledgement](#)
  - [Element: executionAdvice](#)
  - [Element: executionAdviceAcknowledgement](#)
  - [Element: executionAdviceException](#)
  - [Element: executionAdviceRetracted](#)
  - [Element: executionException](#)
  - [Element: executionNotification](#)
  - [Element: executionRetracted](#)
  - [Element: requestAllocation](#)
  - [Element: requestAllocationRetracted](#)
  - [Element: requestClearing](#)
  - [Element: requestClearingRetracted](#)
  - [Element: requestConfirmation](#)
  - [Element: requestConfirmationRetracted](#)
  - [Element: requestConsent](#)
  - [Element: requestConsentRetracted](#)
  - [Element: tradeChangeAdvice](#)
  - [Element: tradeChangeAdviceAcknowledgement](#)
  - [Element: tradeChangeAdviceException](#)
  - [Element: tradeChangeAdviceRetracted](#)
- [Global Definitions](#)
  - [Complex Type: AllocationApproved](#)
  - [Complex Type: AllocationRefused](#)
  - [Complex Type: ClearingConfirmed](#)
  - [Complex Type: ClearingRefused](#)
  - [Complex Type: ConfirmationAgreed](#)
  - [Complex Type: ConfirmationDisputed](#)
  - [Complex Type: ConfirmationRetracted](#)
  - [Complex Type: ConfirmationStatus](#)
  - [Complex Type: ConsentGranted](#)

- [Complex Type: ConsentRefused](#)
- [Complex Type: ExecutionAdvice](#)
- [Complex Type: ExecutionAdviceRetracted](#)
- [Complex Type: ExecutionNotification](#)
- [Complex Type: ExecutionRetracted](#)
- [Complex Type: RequestAllocation](#)
- [Complex Type: RequestAllocationRetracted](#)
- [Complex Type: RequestClearing](#)
- [Complex Type: RequestClearingRetracted](#)
- [Complex Type: RequestConfirmation](#)
- [Complex Type: RequestConsent](#)
- [Complex Type: RequestConsentRetracted](#)
- [Complex Type: TradeChangeAdvice](#)
- [Complex Type: TradeChangeAdviceRetracted](#)

- [Legend](#)

- [Glossary](#)

[top](#)

## Schema Document Properties

<b>Target Namespace</b>	<a href="http://www.fpml.org/FpML-5/confirmation">http://www.fpml.org/FpML-5/confirmation</a>
<b>Version</b>	\$Revision: 2864 \$
<b>Element and Attribute Namespaces</b>	<ul style="list-style-type: none"> <li>• Global element and attribute declarations belong to this schema's target namespace.</li> <li>• By default, local element declarations belong to this schema's target namespace.</li> <li>• By default, local attribute declarations have no namespace.</li> </ul>
<b>Schema Composition</b>	<ul style="list-style-type: none"> <li>• This schema includes components from the following schema document(s):           <ul style="list-style-type: none"> <li>◦ fpml-business-events-5-0.xsd</li> </ul> </li> </ul>
<b>Documentation</b>	Confirmation messages.

## Declared Namespaces

Prefix	Namespace
Default namespace	<a href="http://www.fpml.org/FpML-5/confirmation">http://www.fpml.org/FpML-5/confirmation</a>
xml	http://www.w3.org/XML/1998/namespace
dsig	http://www.w3.org/2000/09/xmldsig#
fpml-annotation	http://www.fpml.org/annotation
xsd	http://www.w3.org/2001/XMLSchema
ecore	http://www.eclipse.org/emf/2002/Ecore

## Schema Component Representation

```

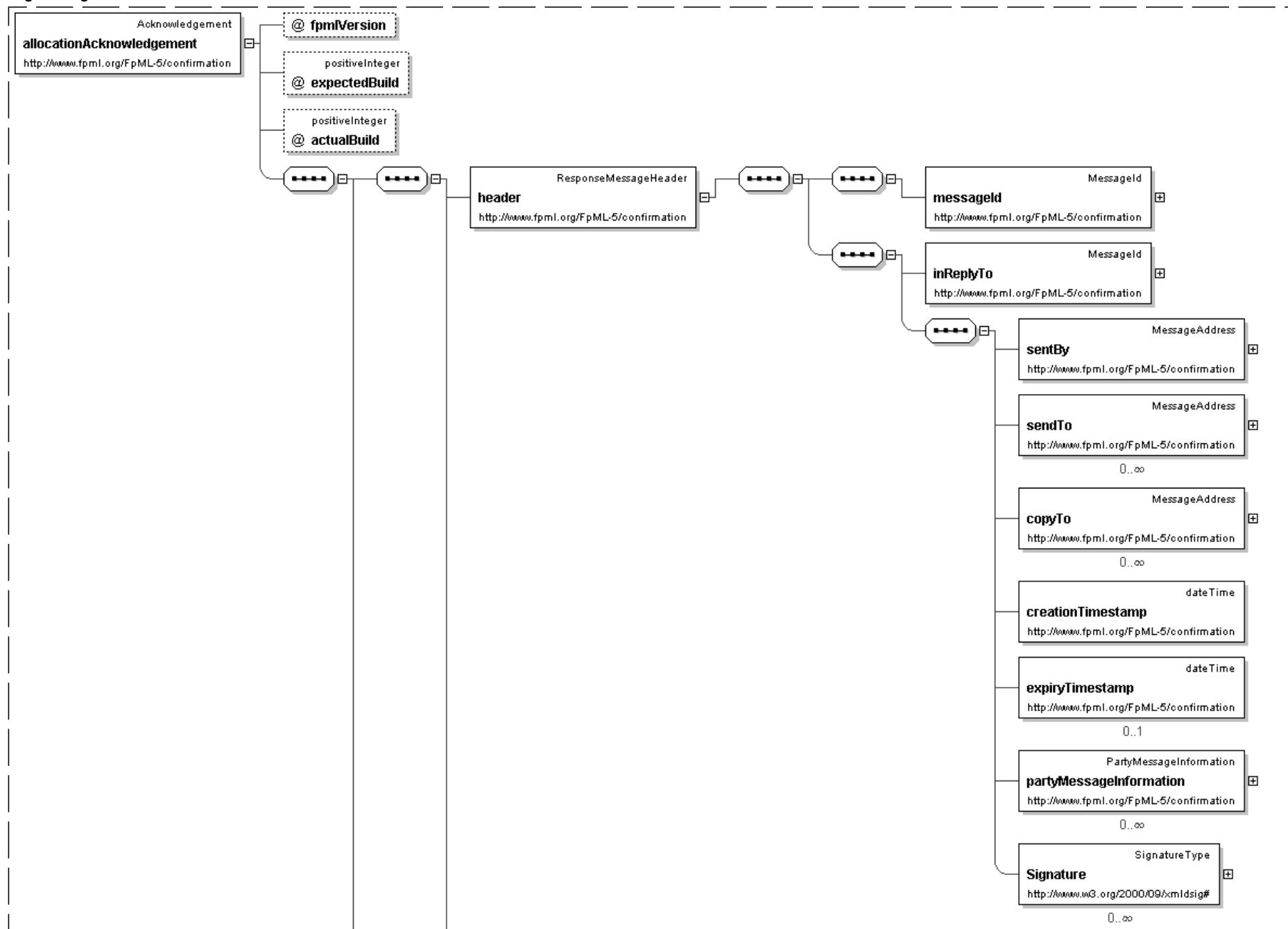
<xsd:schema targetNamespace="http://www.fpml.org/FpML-5/confirmation"
  documentRoot="FpML" nsPrefix="conf" package="org.fpml.confirmation" version="$Revision: 2864 "
  elementFormDefault="qualified" attributeFormDefault="unqualified">
  ...
  <xsd:include schemaLocation="fpml-business-events-5-0.xsd"/>
</xsd:schema>
  
```

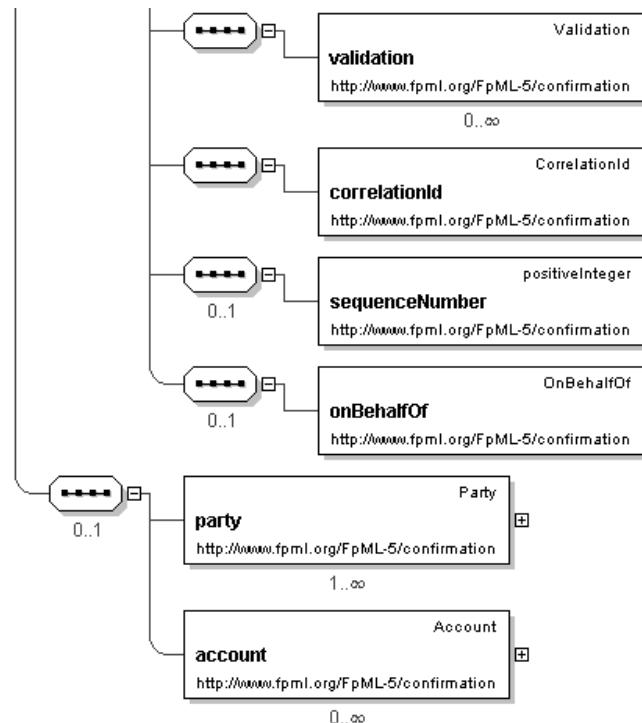
[top](#)

## Global Declarations

### Element: allocationAcknowledgement

<b>Name</b>	allocationAcknowledgement
<b>Type</b>	<a href="#">Acknowledgement</a>
<b>Nillable</b>	no
<b>Abstract</b>	no

**Logical Diagram**

**XML Instance Representation**

```

<allocationAcknowledgement
  fpmlVersion=" xsd:token (value comes from list: {'5-0'}) [1]
  'Indicate which version of the FpML Schema an FpML message adheres to.'

  "
  expectedBuild=" xsd:positiveInteger [0..1]
  'This optional attribute can be supplied by a message creator in an FpML instance to
  specify which build number of the schema was used to define the message when it was generated.'

  "
  actualBuild="8 [0..1]
  'The specific build number of this schema version. This attribute is not included in
  an instance document. Instead, it is supplied by the XML parser when the document is
  validated against the FpML schema and indicates the build number of the schema file. Every
  time FpML publishes a change to the schema, validation rules, or examples within a version
  (e.g., version 4.2) the actual build number is incremented. If no changes have been
  made between releases within a version (i.e. from Trial Recommendation to Recommendation)
  the actual build number stays the same.'

  ">
  <header> ResponseMessageHeader </header> [1]
  <validation> Validation </validation> [0..*]
  <correlationId> CorrelationId </correlationId> [1]
  'A qualified identifier used to correlate between messages'

  Start Group: Sequence.model [0..1]
  <sequenceNumber> xsd:positiveInteger </sequenceNumber> [1]
  
```

'A numeric value that can be used to order messages with the same correlation identifier from the same sender.'

End Group: Sequence.model

Start Group: OnBehalfOf.model [0..1]

<onBehalfOf> OnBehalfOf </onBehalfOf> [1]

'Indicates which party (and accounts) a trade is being processed for.'

End Group: OnBehalfOf.model

Start Group: PartiesAndAccounts.model [0..1]

<party> Party </party> [1..\*]

'A legal entity or a subdivision of a legal entity.', 'Parties can perform multiple roles in a trade lifecycle. For example, the principal parties obligated to make payments from time to time during the term of the trade, but may include other parties involved in, or incidental to, the trade, such as parties acting in the role of novation transferor/ transferee, broker, calculation agent, etc. In FpML roles are defined in multiple places within a document.'

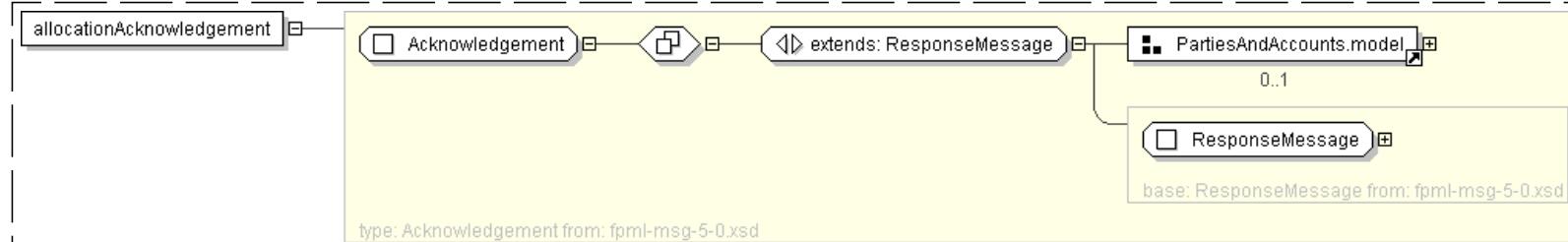
<account> Account </account> [0..\*]

'Optional account information used to precisely define the origination and destination of financial instruments.'

End Group: PartiesAndAccounts.model

</allocationAcknowledgement>

#### Diagram



#### Schema Component Representation

```
<xsd:element name="allocationAcknowledgement" type=" Acknowledgement " />
```

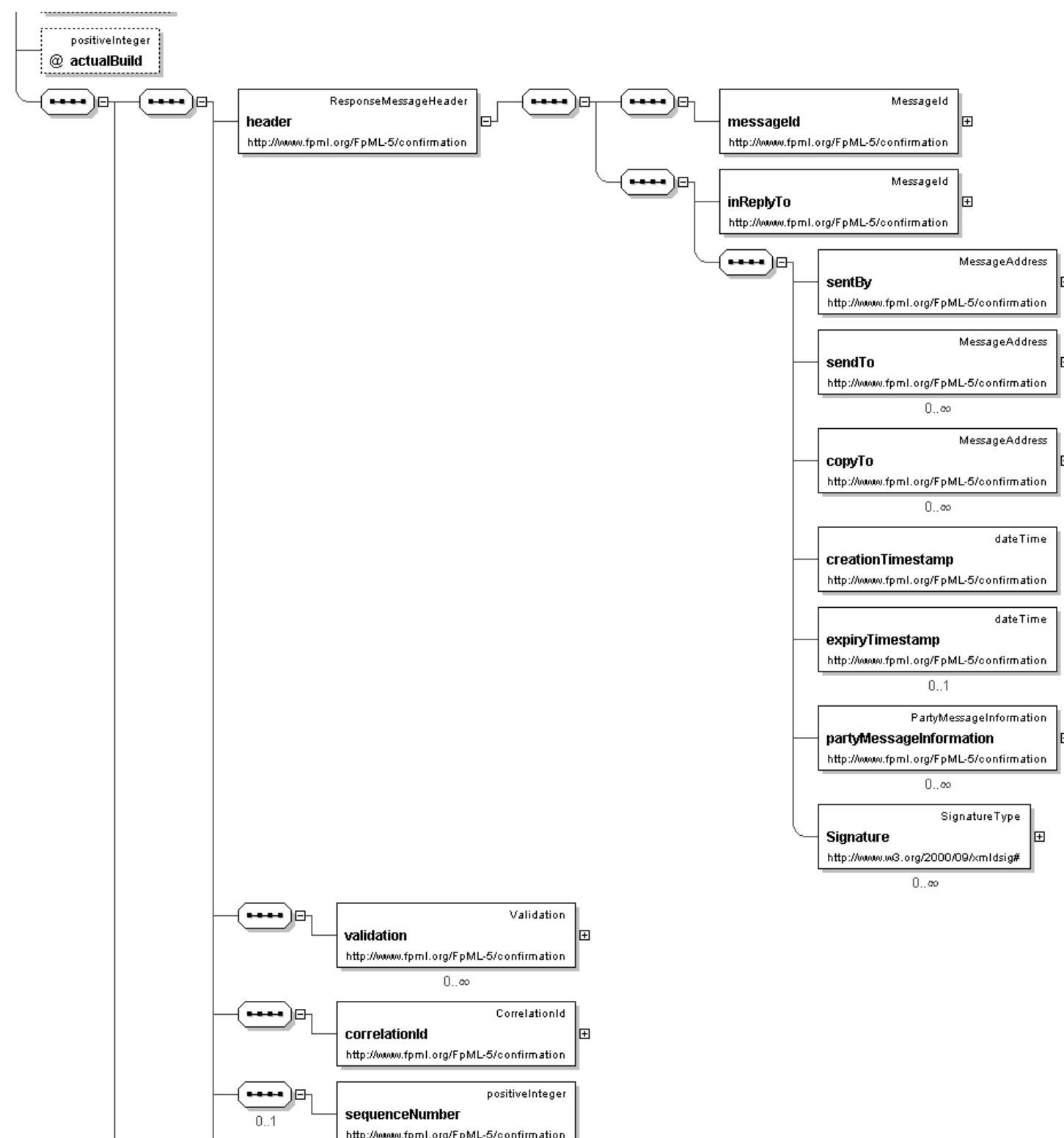
[top](#)

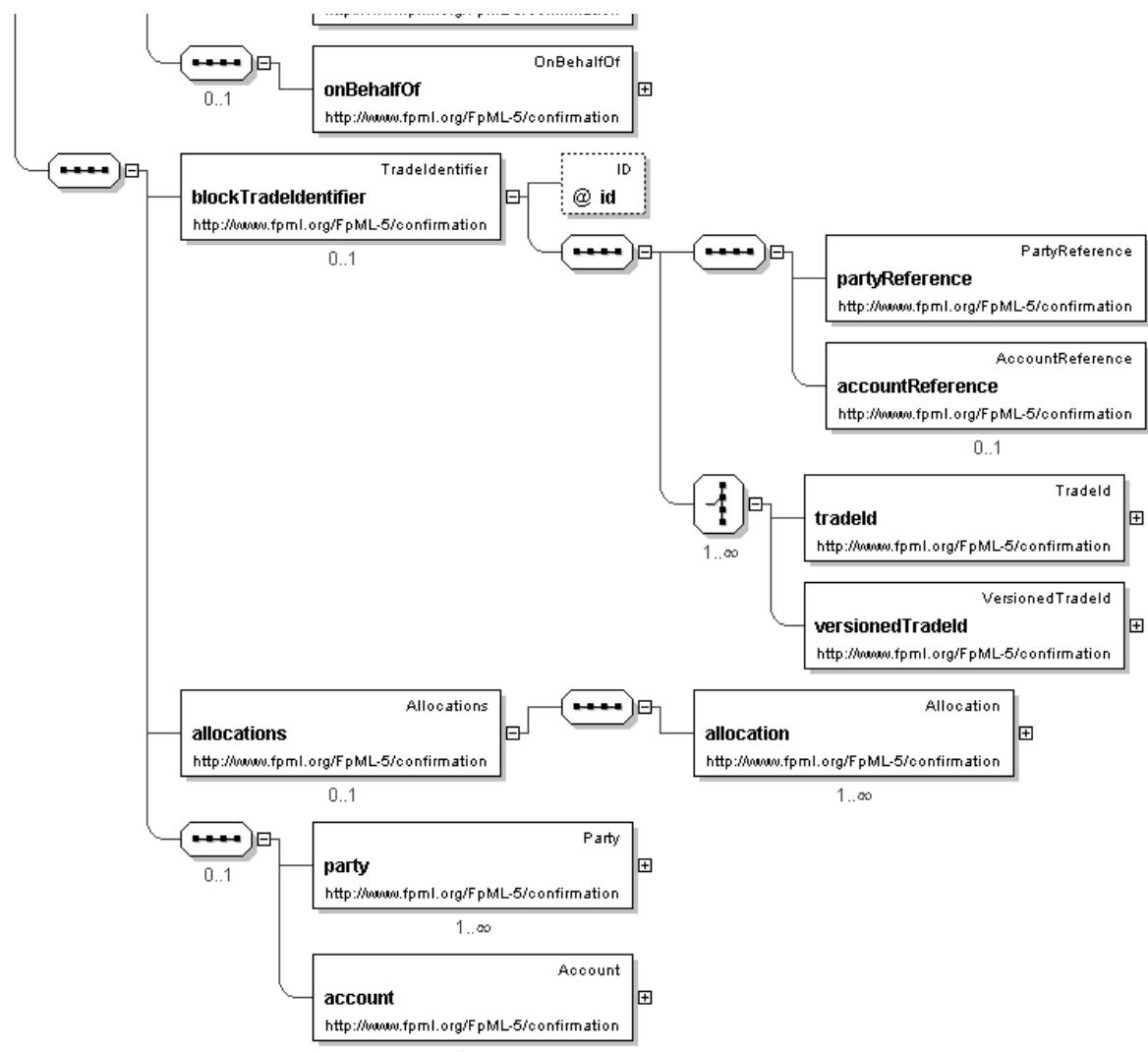
## Element: allocationApproved

Name	allocationApproved
Type	AllocationApproved
Nullable	no
Abstract	no

#### Logical Diagram





**XML Instance Representation**

```

<allocationApproved
fpmlVersion=" xsd:token (value comes from list: {'5-0'}) [1]
'Indicate which version of the FpML Schema an FpML message adheres to.'

"
expectedBuild=" xsd:positiveInteger [0..1]
'This optional attribute can be supplied by a message creator in an FpML instance to
specify which build number of the schema was used to define the message when it was generated.'

"
actualBuild="8 [0..1]
'The specific build number of this schema version. This attribute is not included in
an instance document. Instead, it is supplied by the XML parser when the document is
validated against the FpML schema and indicates the build number of the schema file. Every
  
```

time FpML publishes a change to the schema, validation rules, or examples within a version (e.g., version 4.2) the actual build number is incremented. If no changes have been made between releases within a version (i.e. from Trial Recommendation to Recommendation) the actual build number stays the same.'

">  
 <header> ResponseMessageHeader </header> [1]  
 <validation> Validation </validation> [0..\*]  
 <correlationId> CorrelationId </correlationId> [1]  
 'A qualified identifier used to correlate between messages'

Start Group: Sequence.model [0..1]  
 <sequenceNumber> xsd:positiveInteger </sequenceNumber> [1]  
 'A numeric value that can be used to order messages with the same correlation identifier from the same sender.'

End Group: Sequence.model  
 Start Group: OnBehalfOf.model [0..1]  
 <onBehalfOf> OnBehalfOf </onBehalfOf> [1]  
 'Indicates which party (and accounts) a trade is being processed for.'

End Group: OnBehalfOf.model  
 <blockTradeIdentifier> TradeIdentifier </blockTradeIdentifier> [0..1]  
 <allocations> Allocations </allocations> [0..1]  
 Start Group: PartiesAndAccounts.model [0..1]  
 <party> Party </party> [1..\*]  
 'A legal entity or a subdivision of a legal entity.', 'Parties can perform multiple roles in a trade lifecycle. For example, the principal parties obligated to make payments from time to time during the term of the trade, but may include other parties involved in, or incidental to, the trade, such as parties acting in the role of novation transferor/ transferee, broker, calculation agent, etc. In FpML roles are defined in multiple places within a document.'

<account> Account </account> [0..\*]  
 'Optional account information used to precisely define the origination and destination of financial instruments.'

End Group: PartiesAndAccounts.model  
 </allocationApproved>

## Diagram



## Schema Component Representation

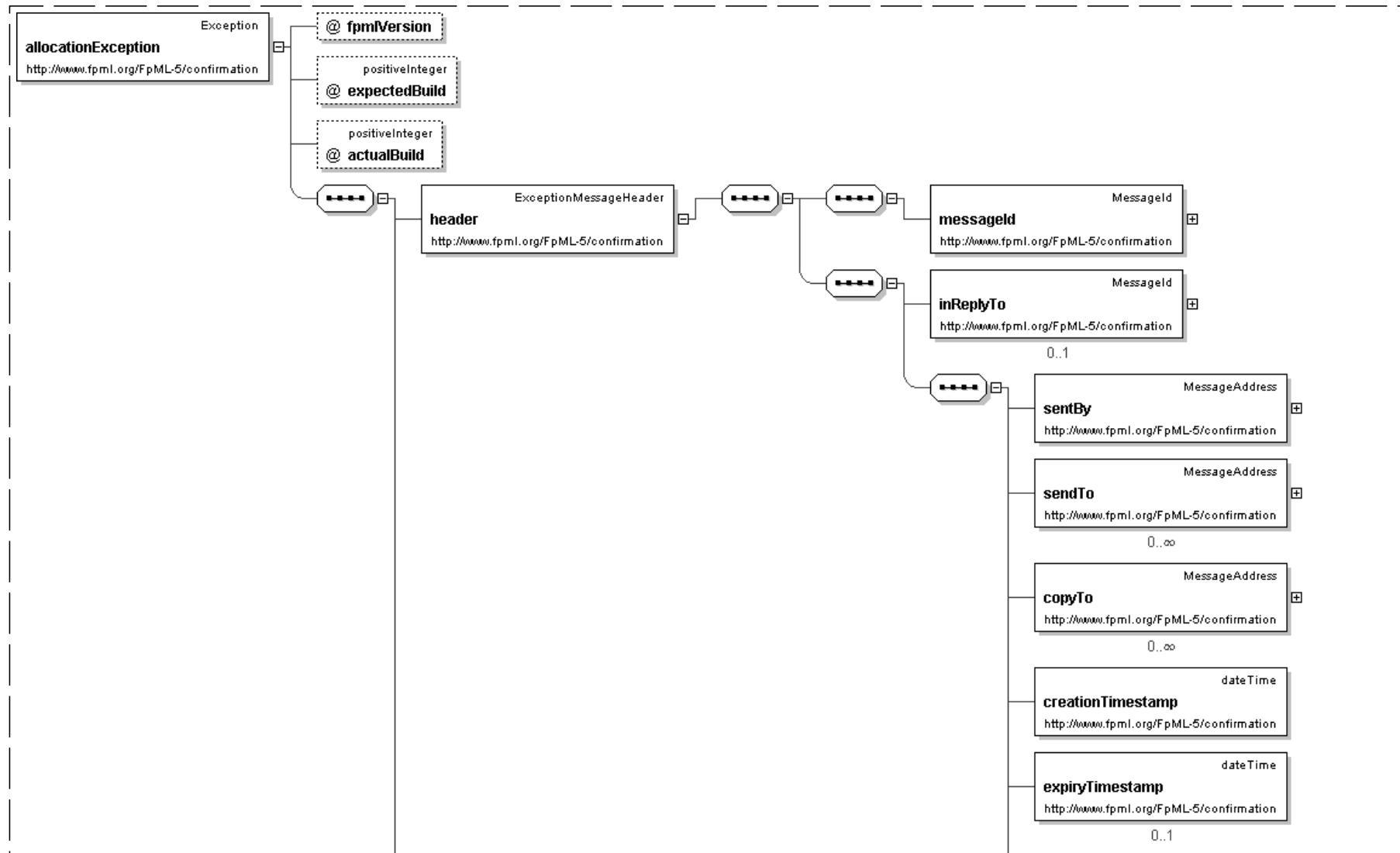
```
<xsd:element name="allocationApproved" type=" AllocationApproved " />
```

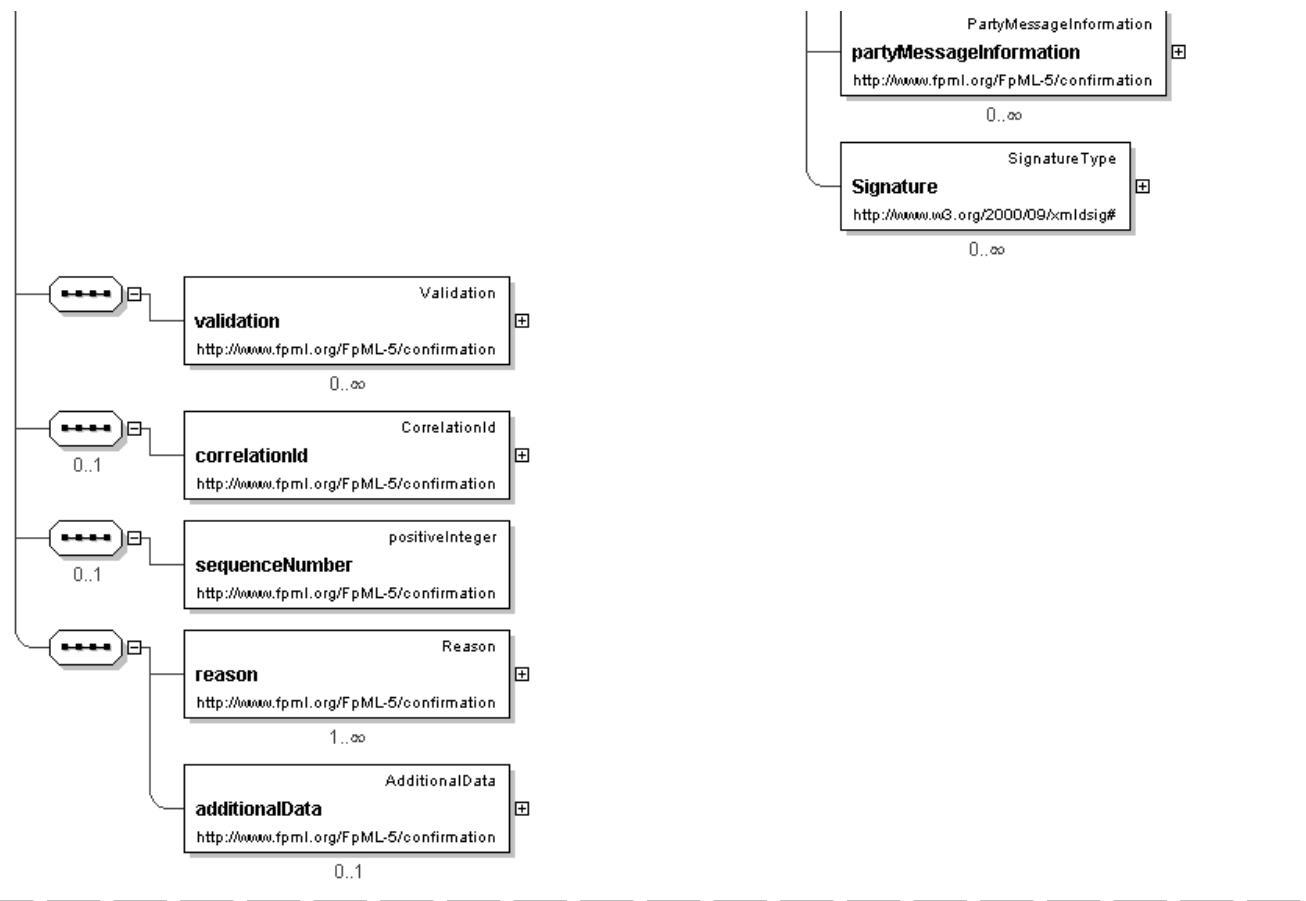
[top](#)

## Element: allocationException

Name	allocationException
Type	Exception
Nillable	no
Abstract	no

## Logical Diagram



**XML Instance Representation**

```

<allocationException
  fpmVersion=" xsd:token (value comes from list: {'5-0'}) [1]
  'Indicate which version of the FpML Schema an FpML message adheres to.'
  "
  expectedBuild=" xsd:positiveInteger [0..1]
  'This optional attribute can be supplied by a message creator in an FpML instance to
  specify which build number of the schema was used to define the message when it was generated.'
  "
  actualBuild="8 [0..1]
  'The specific build number of this schema version. This attribute is not included in
  an instance document. Instead, it is supplied by the XML parser when the document is
  validated against the FpML schema and indicates the build number of the schema file. Every
  time FpML publishes a change to the schema, validation rules, or examples within a version
  (e.g., version 4.2) the actual build number is incremented. If no changes have been
  made between releases within a version (i.e. from Trial Recommendation to Recommendation)
  the actual build number stays the same.'
  ">
  <header> ExceptionMessageHeader </header> [1]
  <validation> Validation </validation> [0..*]

```

```

| Start Group: Correlation.model [0..1]
|   <correlationId> CorrelationId </correlationId> [1]
|     'A qualified identifier used to correlate between messages'

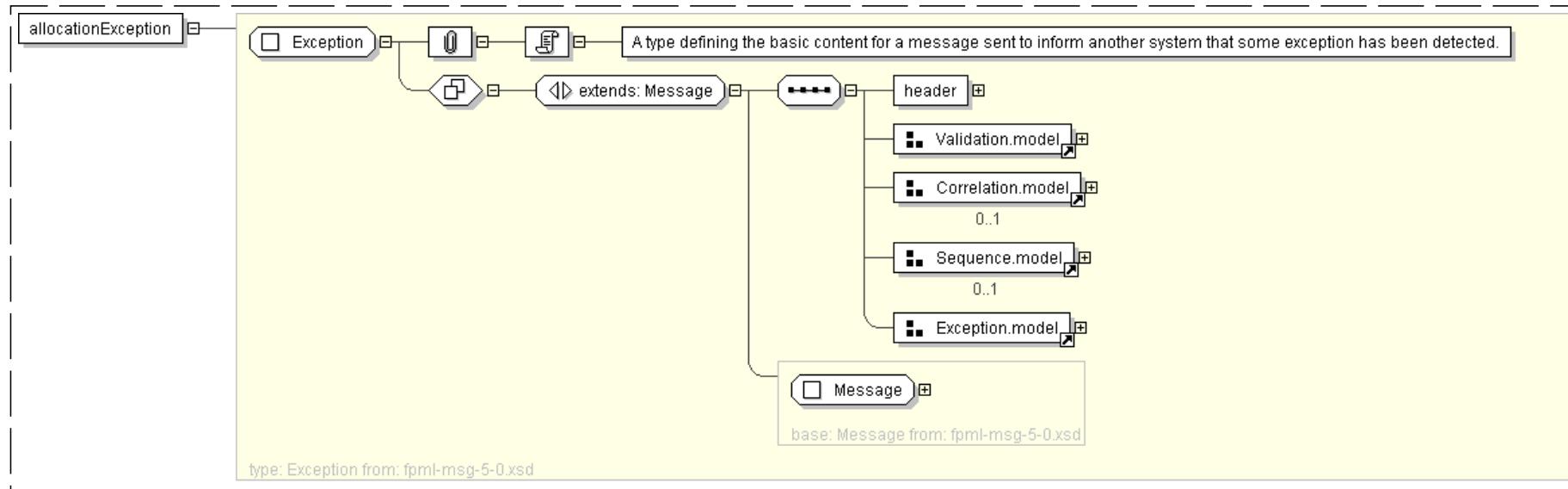
End Group: Correlation.model
Start Group: Sequence.model [0..1]
  <sequenceNumber> xsd:positiveInteger </sequenceNumber> [1]
  'A numeric value that can be used to order messages with the same correlation identifier
  from the same sender.'

End Group: Sequence.model
<reason> Reason </reason> [1..*]
'An instance of the Reason type used to record the nature of any errors associated with
a message.'

<additionalData> AdditionalData </additionalData> [0..1]
'Any string of additional data that may help the message processor, for example in a
rejection message this might contain a code value or the text of the original request (within
a CDATA section).'

</allocationException>

```

**Diagram****Schema Component Representation**

```
<xsd:element name="allocationException" type="<u>Exception</u>" />
```

top

**Element: allocationRefused**

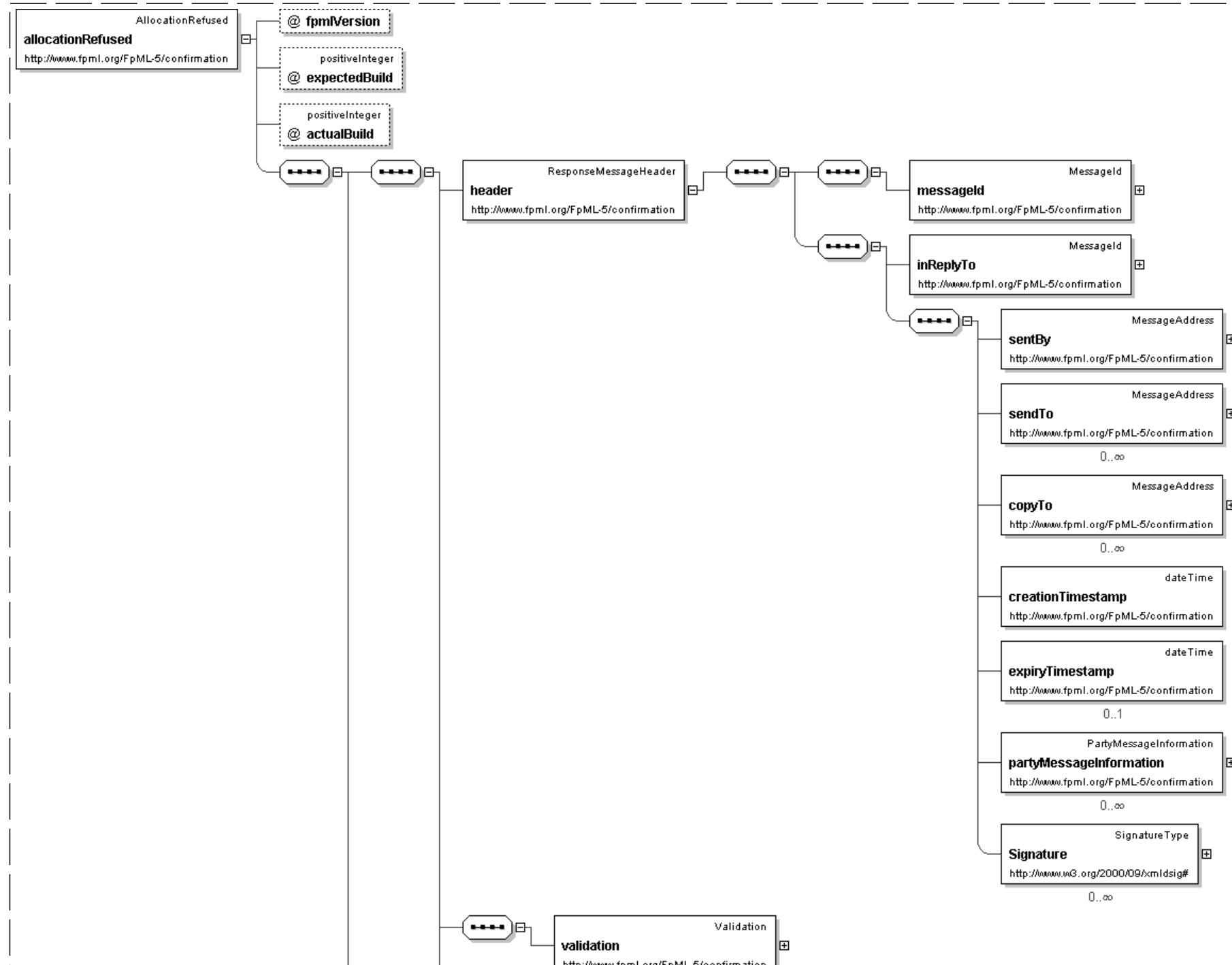
Name	allocationRefused
Type	<a href="#">AllocationRefused</a>

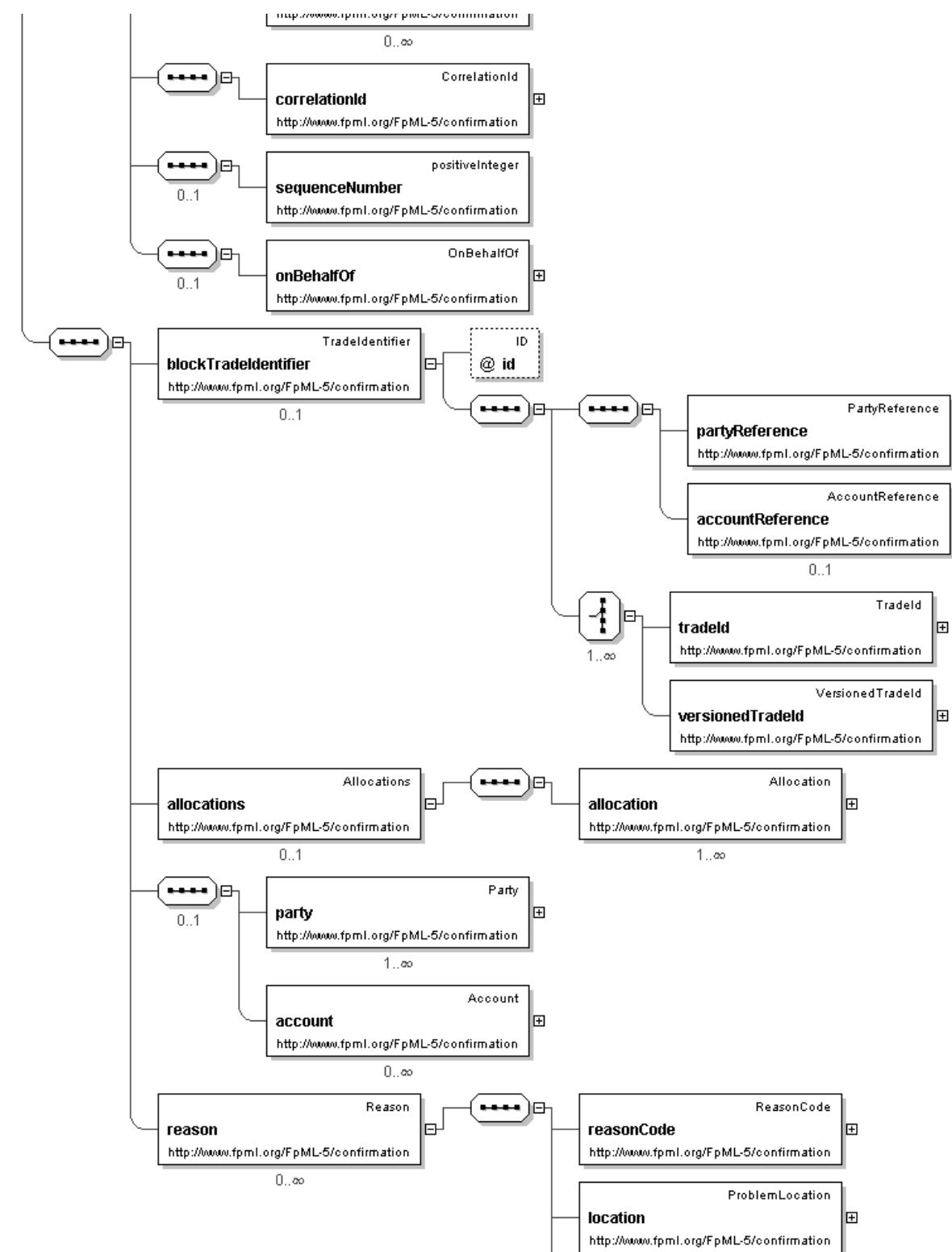
**Nillable**

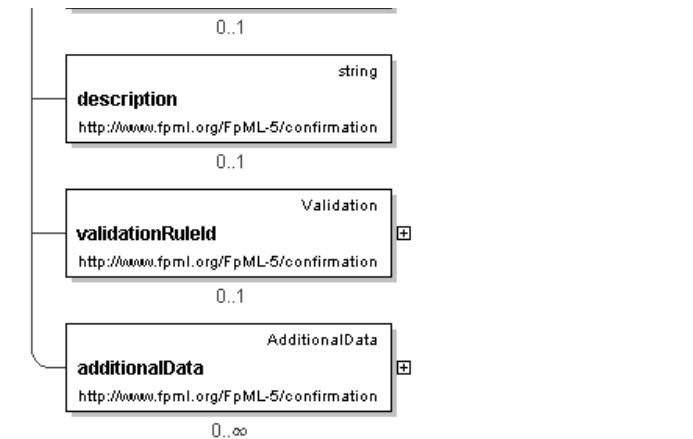
no

**Abstract**

no

**Logical Diagram**



**XML Instance Representation**

```

<allocationRefused
fpmVersion=" xsd:token (value comes from list: {'5-0'}) [1]
'Indicate which version of the FpML Schema an FpML message adheres to.'

"
expectedBuild=" xsd:positiveInteger [0..1]
'This optional attribute can be supplied by a message creator in an FpML instance to
specify which build number of the schema was used to define the message when it was generated.'

"
actualBuild="8 [0..1]
'The specific build number of this schema version. This attribute is not included in
an instance document. Instead, it is supplied by the XML parser when the document is
validated against the FpML schema and indicates the build number of the schema file. Every
time FpML publishes a change to the schema, validation rules, or examples within a version
(e.g., version 4.2) the actual build number is incremented. If no changes have been
made between releases within a version (i.e. from Trial Recommendation to Recommendation)
the actual build number stays the same.'

">
<header> ResponseMessageHeader </header> [1]
<validation> Validation </validation> [0..*]
<correlationId> CorrelationId </correlationId> [1]
'A qualified identifier used to correlate between messages'

Start Group: Sequence.model [0..1]
<sequenceNumber> xsd:positiveInteger </sequenceNumber> [1]
'A numeric value that can be used to order messages with the same correlation identifier
from the same sender.'

End Group: Sequence.model
Start Group: OnBehalfOf.model [0..1]
<onBehalfOf> OnBehalfOf </onBehalfOf> [1]
'Indicates which party (and accounts) a trade is being processed for.'

End Group: OnBehalfOf.model
<blockTradeIdentifier> TradeIdentifier </blockTradeIdentifier> [0..1]
<allocations> Allocations </allocations> [0..1]

```

Start Group: PartiesAndAccounts.model [0..1]  
| <party> Party </party> [1..\*]

'A legal entity or a subdivision of a legal entity.', 'Parties can perform multiple roles in a trade lifecycle. For example, the principal parties obligated to make payments from time to time during the term of the trade, but may include other parties involved in, or incidental to, the trade, such as parties acting in the role of novation transferor/transferee, broker, calculation agent, etc. In FpML roles are defined in multiple places within a document.'

<account> Account </account> [0..\*]

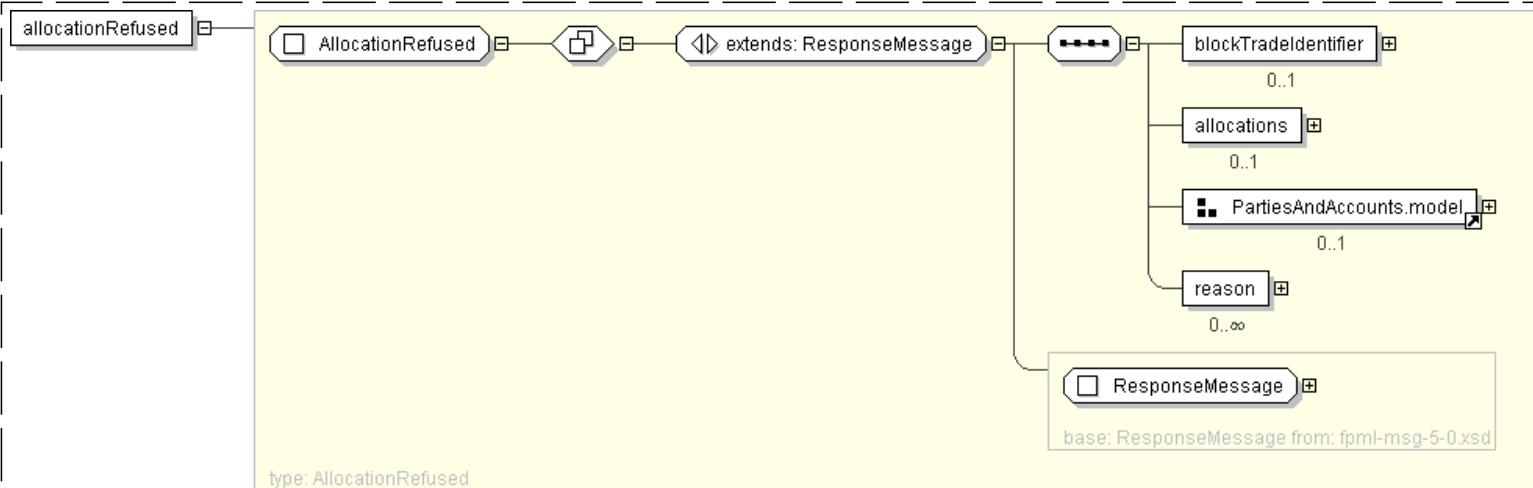
*'Optional account information used to precisely define the origination and destination of financial instruments.'*

End Group: PartiesAndAccounts.model

<reason> Reason </reason> [0..\*]

</allocationRefused>

## Diagram



## Schema Component Representations

```
<xsd:element name="allocationRefused" type=" AllocationRefused "/>
```

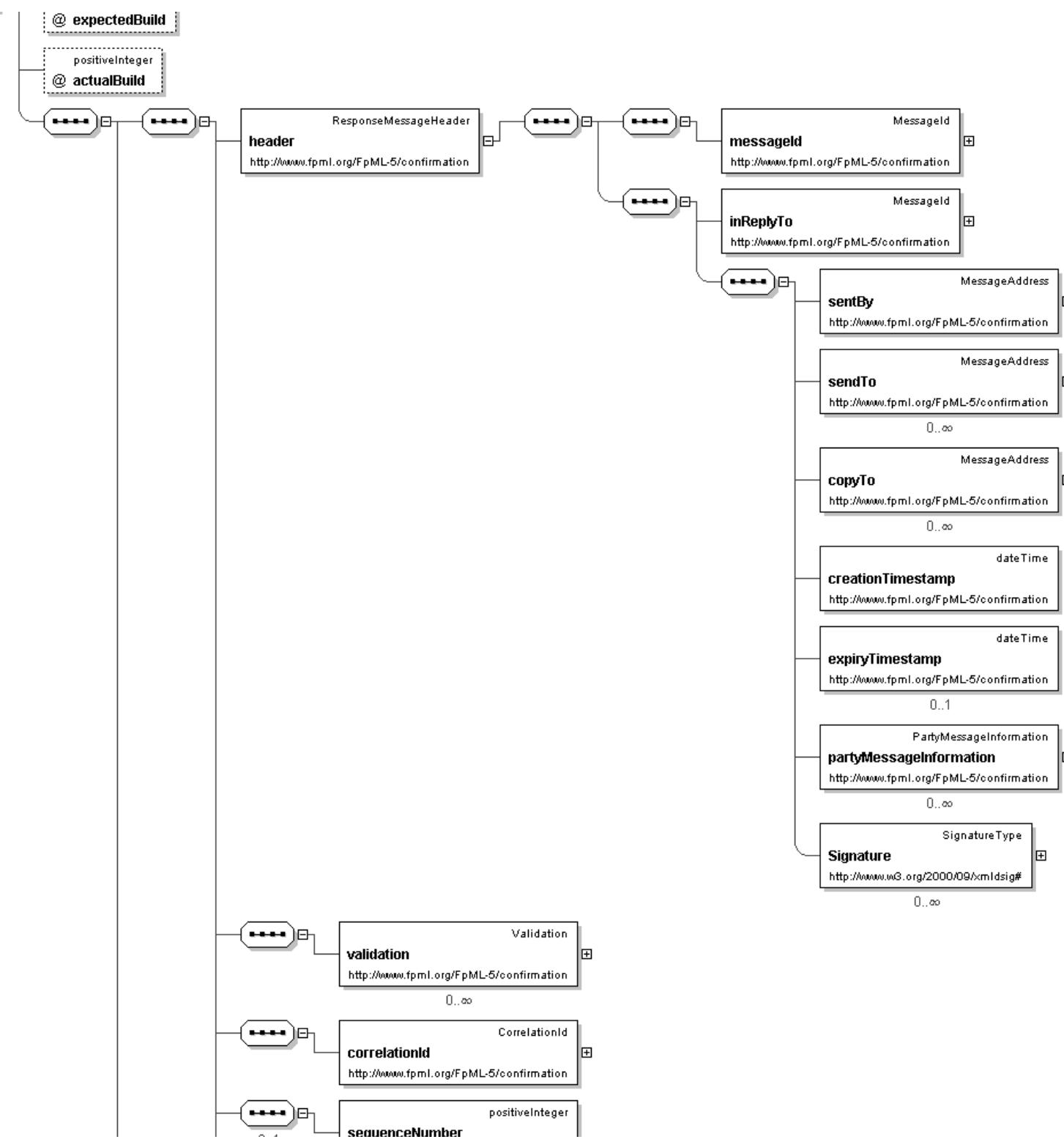
top

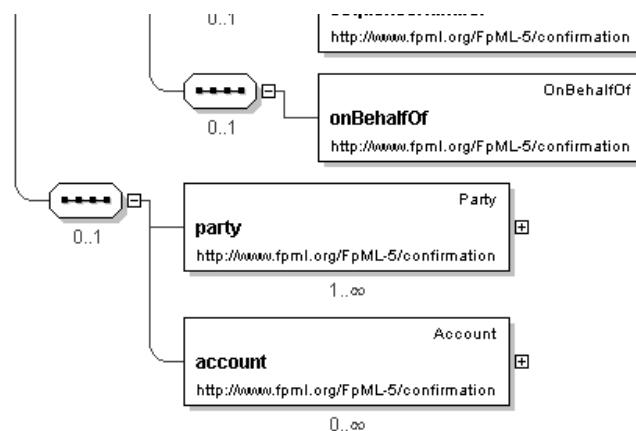
## **Element: clearingAcknowledgement**

<b>Name</b>	clearingAcknowledgement
<b>Type</b>	<a href="#">Acknowledgement</a>
<b>Nullable</b>	no
<b>Abstract</b>	no

## Logical Diagram





**XML Instance Representation**

```

<clearingAcknowledgement
fpmlVersion=" xsd:token (value comes from list: {'5-0'}) [1]
'Indicate which version of the FpML Schema an FpML message adheres to.'

"
expectedBuild=" xsd:positiveInteger [0..1]
'This optional attribute can be supplied by a message creator in an FpML instance to
specify which build number of the schema was used to define the message when it was generated.

"
actualBuild="8 [0..1]
'The specific build number of this schema version. This attribute is not included in
an instance document. Instead, it is supplied by the XML parser when the document is
validated against the FpML schema and indicates the build number of the schema file. Every
time FpML publishes a change to the schema, validation rules, or examples within a version
(e.g., version 4.2) the actual build number is incremented. If no changes have been
made between releases within a version (i.e. from Trial Recommendation to Recommendation)
the actual build number stays the same.

">
<header> ResponseMessageHeader </header> [1]
<validation> Validation </validation> [0..*]
<correlationId> CorrelationId </correlationId> [1]
'A qualified identifier used to correlate between messages'

Start Group: Sequence.model [0..1]
<sequenceNumber> xsd:positiveInteger </sequenceNumber> [1]
'A numeric value that can be used to order messages with the same correlation identifier
from the same sender.'

End Group: Sequence.model
Start Group: OnBehalfOf.model [0..1]
<onBehalfOf> OnBehalfOf </onBehalfOf> [1]
'Indicates which party (and accounts) a trade is being processed for.

End Group: OnBehalfOf.model
Start Group: PartiesAndAccounts.model [0..1]
<party> Party </party> [1..*]

```

'A legal entity or a subdivision of a legal entity.' , 'Parties can perform multiple roles in a trade lifecycle. For example, the principal parties obligated to make payments from time to time during the term of the trade, but may include other parties involved in, or incidental to, the trade, such as parties acting in the role of novation transferor/transferee, broker, calculation agent, etc. In FpML roles are defined in multiple places within a document.'

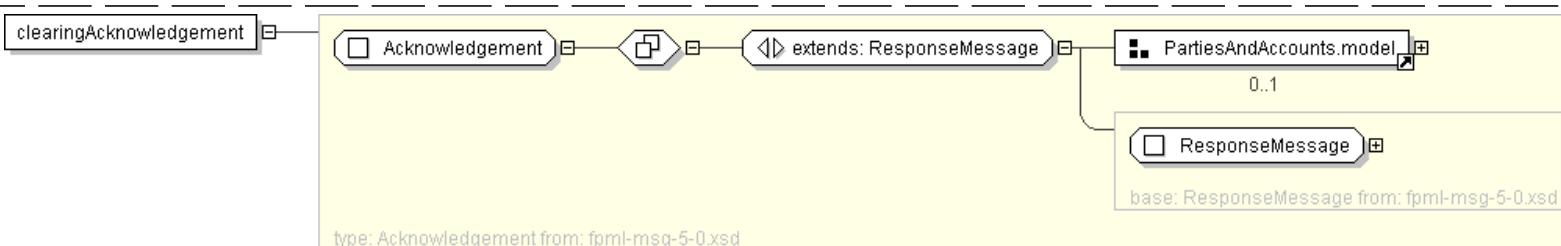
<account> Account </account> [0..\*]

'Optional account information used to precisely define the origination and destination of financial instruments.'

End Group: PartiesAndAccounts.model

</clearingAcknowledgement>

#### Diagram



#### Schema Component Representation

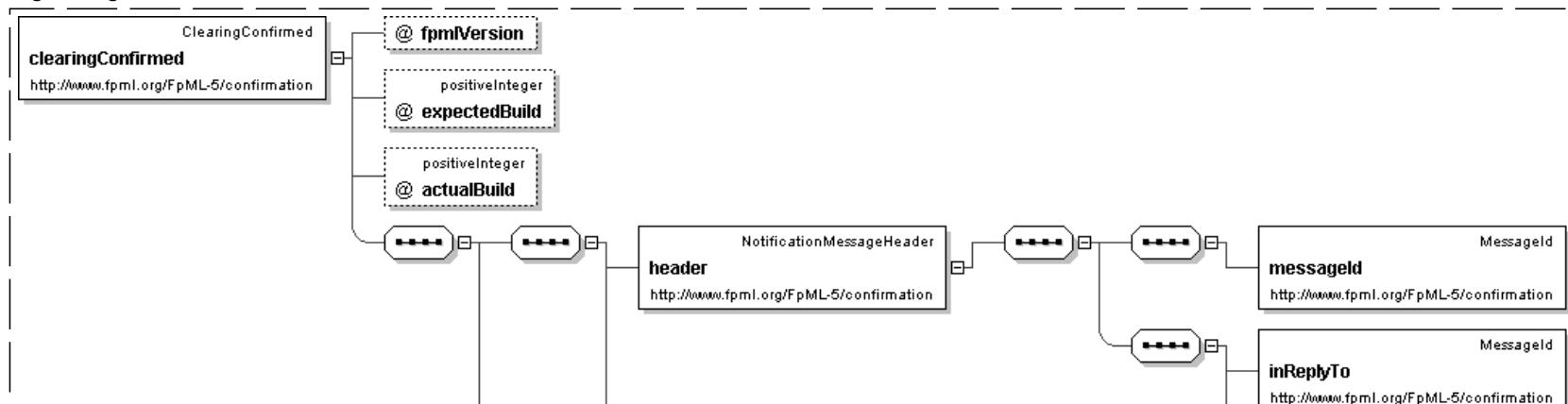
```
<xsd:element name="clearingAcknowledgement" type=" Acknowledgement " />
```

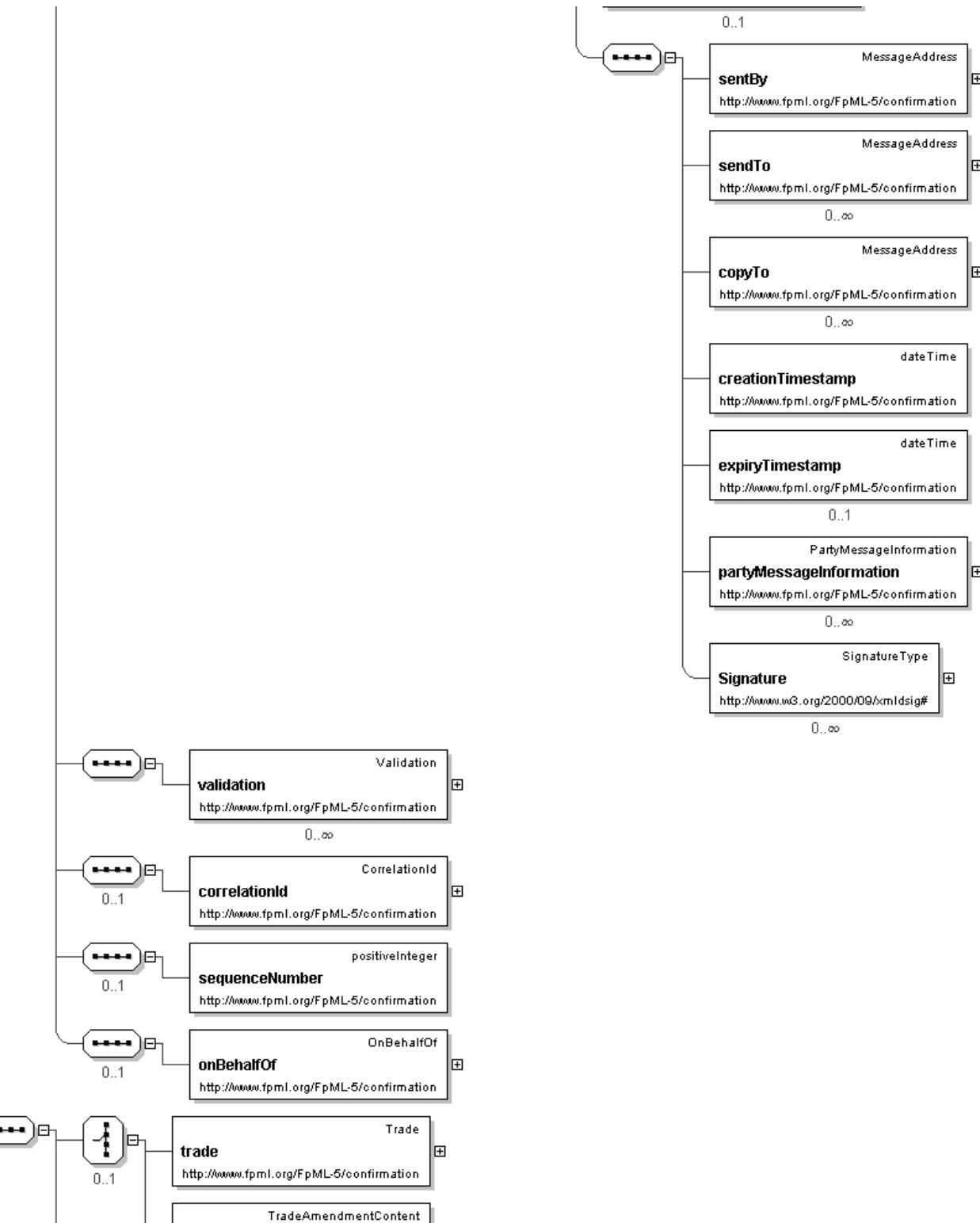
[top](#)

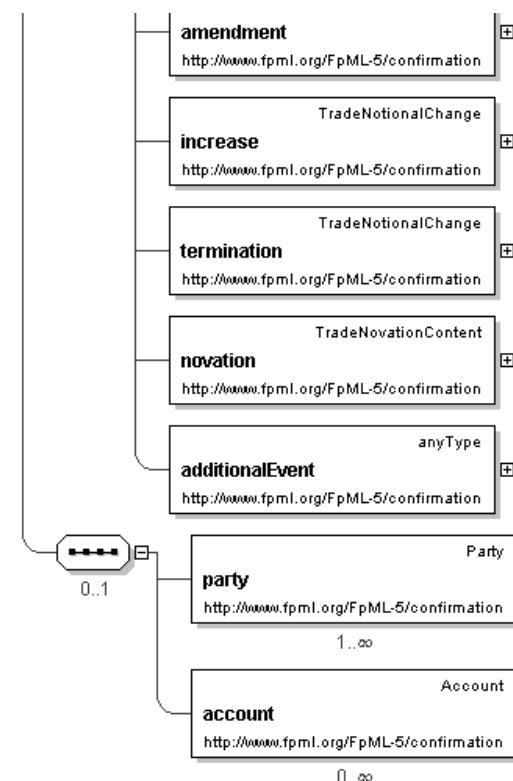
#### Element: clearingConfirmed

Name	clearingConfirmed
Type	ClearingConfirmed
Nillable	no
Abstract	no

#### Logical Diagram





**XML Instance Representation**

```

<clearingConfirmed
fpmlVersion=" xsd:token (value comes from list: {'5-0'}) [1]
'Indicate which version of the FpML Schema an FpML message adheres to.'

"
expectedBuild=" xsd:positiveInteger [0..1]
'This optional attribute can be supplied by a message creator in an FpML instance to
specify which build number of the schema was used to define the message when it was generated.'

"
actualBuild="8 [0..1]
'The specific build number of this schema version. This attribute is not included in
an instance document. Instead, it is supplied by the XML parser when the document is
validated against the FpML schema and indicates the build number of the schema file. Every
time FpML publishes a change to the schema, validation rules, or examples within a version
(e.g., version 4.2) the actual build number is incremented. If no changes have been
made between releases within a version (i.e. from Trial Recommendation to Recommendation)
the actual build number stays the same.'


">
<header> NotificationMessageHeader </header> [1]
<validation> Validation </validation> [0..*]
Start Group: Correlation.model [0..1]
<correlationId> CorrelationId </correlationId> [1]
'A qualified identifier used to correlate between messages'
  
```

End Group: Correlation.model  
Start Group: Sequence.model [0..1]  
  <sequenceNumber> xsd:positiveInteger </sequenceNumber> [1]  
  'A numeric value that can be used to order messages with the same correlation identifier from the same sender.'

End Group: Sequence.model  
Start Group: OnBehalfOf.model [0..1]  
  <onBehalfOf> OnBehalfOf </onBehalfOf> [1]  
  'Indicates which party (and accounts) a trade is being processed for.'

End Group: OnBehalfOf.model  
Start Group: Events.model [0..1]  
Start Choice [1]  
  <trade> Trade </trade> [1]  
  <amendment> TradeAmendmentContent </amendment> [1]  
  <increase> TradeNotionalChange </increase> [1]  
  <termination> TradeNotionalChange </termination> [1]  
  <novation> TradeNovationContent </novation> [1]  
  <additionalEvent> ... </additionalEvent> [1]

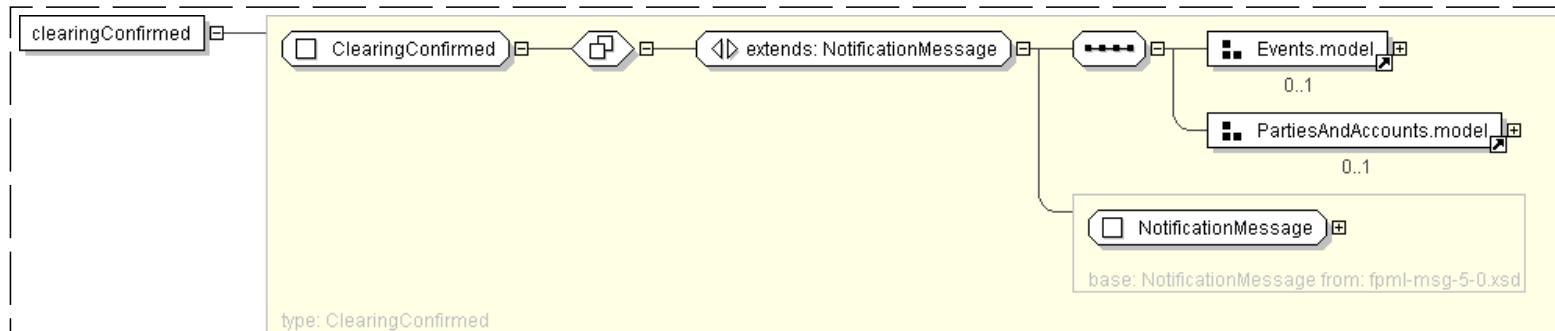
End Choice

End Group: Events.model  
Start Group: PartiesAndAccounts.model [0..1]  
  <party> Party </party> [1..\*]

'A legal entity or a subdivision of a legal entity.', 'Parties can perform multiple roles in a trade lifecycle. For example, the principal parties obligated to make payments from time to time during the term of the trade, but may include other parties involved in, or incidental to, the trade, such as parties acting in the role of novation transferor/transferee, broker, calculation agent, etc. In FpML roles are defined in multiple places within a document.'

<account> Account </account> [0..\*]  
'Optional account information used to precisely define the origination and destination of financial instruments.'

End Group: PartiesAndAccounts.model  
</clearingConfirmed>

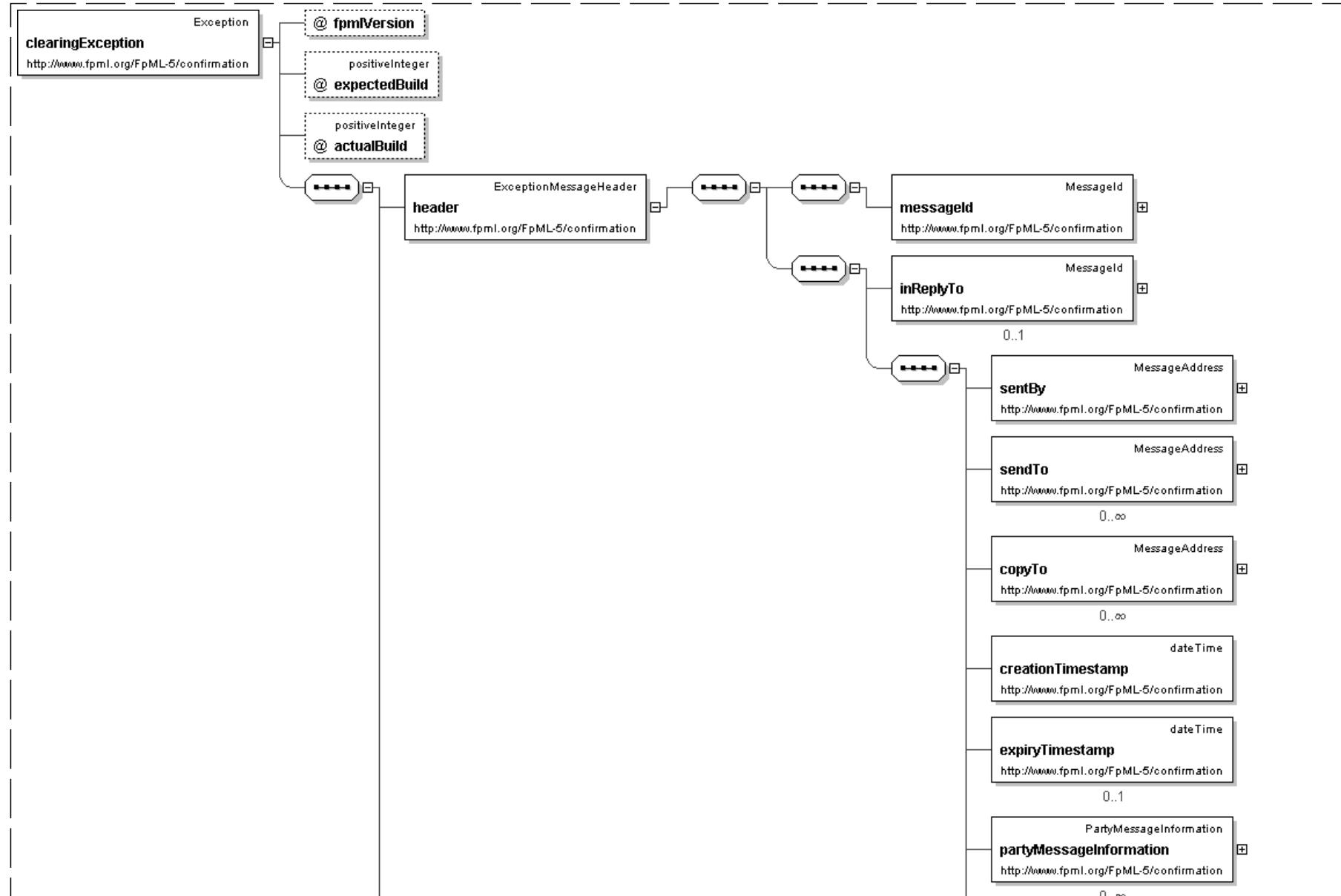
**Diagram****Schema Component Representation**

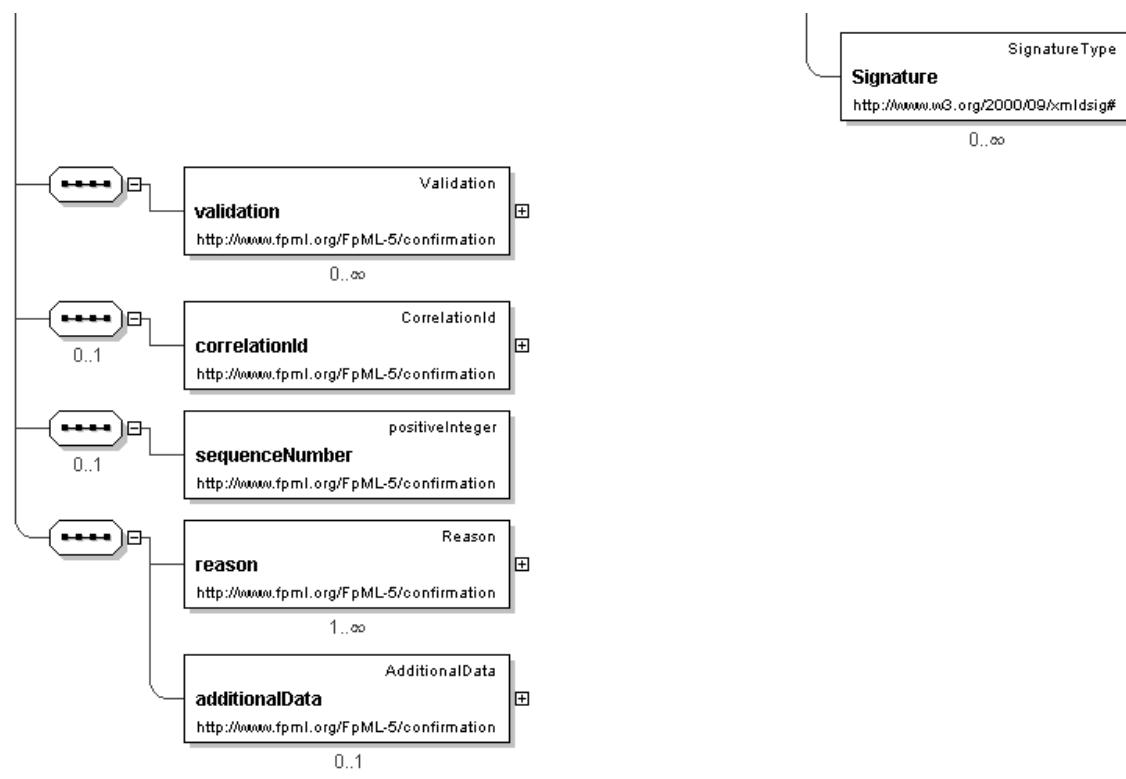
```
<xsd:element name="clearingConfirmed" type=" ClearingConfirmed "/>
```

## Element: clearingException

Name	clearingException
Type	Exception
Nillable	no
Abstract	no

### Logical Diagram



**XML Instance Representation**

```

<clearingException
fpmlVersion=" xsd:token (value comes from list: {'5-0'}) [1]
'Indicate which version of the FpML Schema an FpML message adheres to.'

"
expectedBuild=" xsd:positiveInteger [0..1]
'This optional attribute can be supplied by a message creator in an FpML instance to
specify which build number of the schema was used to define the message when it was generated.'

"
actualBuild="8 [0..1]
'The specific build number of this schema version. This attribute is not included in
an instance document. Instead, it is supplied by the XML parser when the document is
validated against the FpML schema and indicates the build number of the schema file. Every
time FpML publishes a change to the schema, validation rules, or examples within a version
(e.g., version 4.2) the actual build number is incremented. If no changes have been
made between releases within a version (i.e. from Trial Recommendation to Recommendation)
the actual build number stays the same.'

">
<header> ExceptionMessageHeader </header> [1]
<validation> Validation </validation> [0..*]
Start Group: Correlation.model [0..1]
<correlationId> CorrelationId </correlationId> [1]
'A qualified identifier used to correlate between messages'
  
```

```

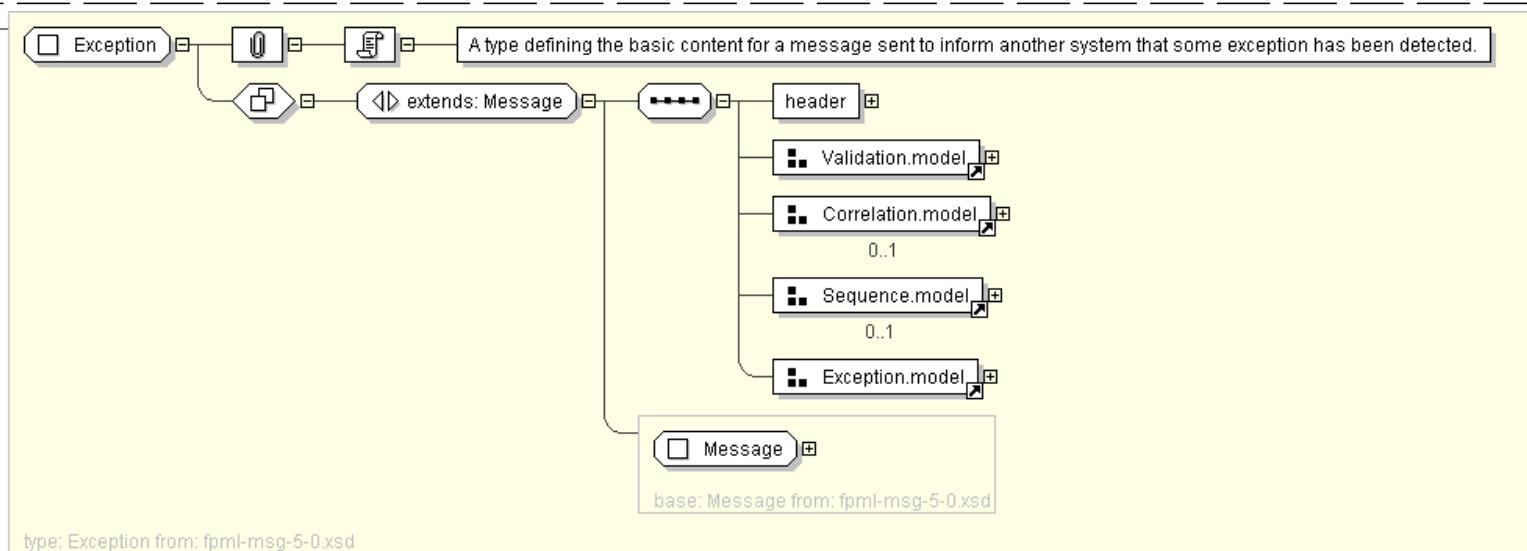
End Group: Correlation.model
Start Group: Sequence.model [0..1]
<sequenceNumber> xsd:positiveInteger </sequenceNumber> [1]
'A numeric value that can be used to order messages with the same correlation identifier
from the same sender.'

End Group: Sequence.model
<reason> Reason </reason> [1..*]
'An instance of the Reason type used to record the nature of any errors associated with
a message.'

<additionalData> AdditionalData </additionalData> [0..1]
'Any string of additional data that may help the message processor, for example in a
rejection message this might contain a code value or the text of the original request (within
a CDATA section).'

</clearingException>

```

**Diagram****Schema Component Representation**

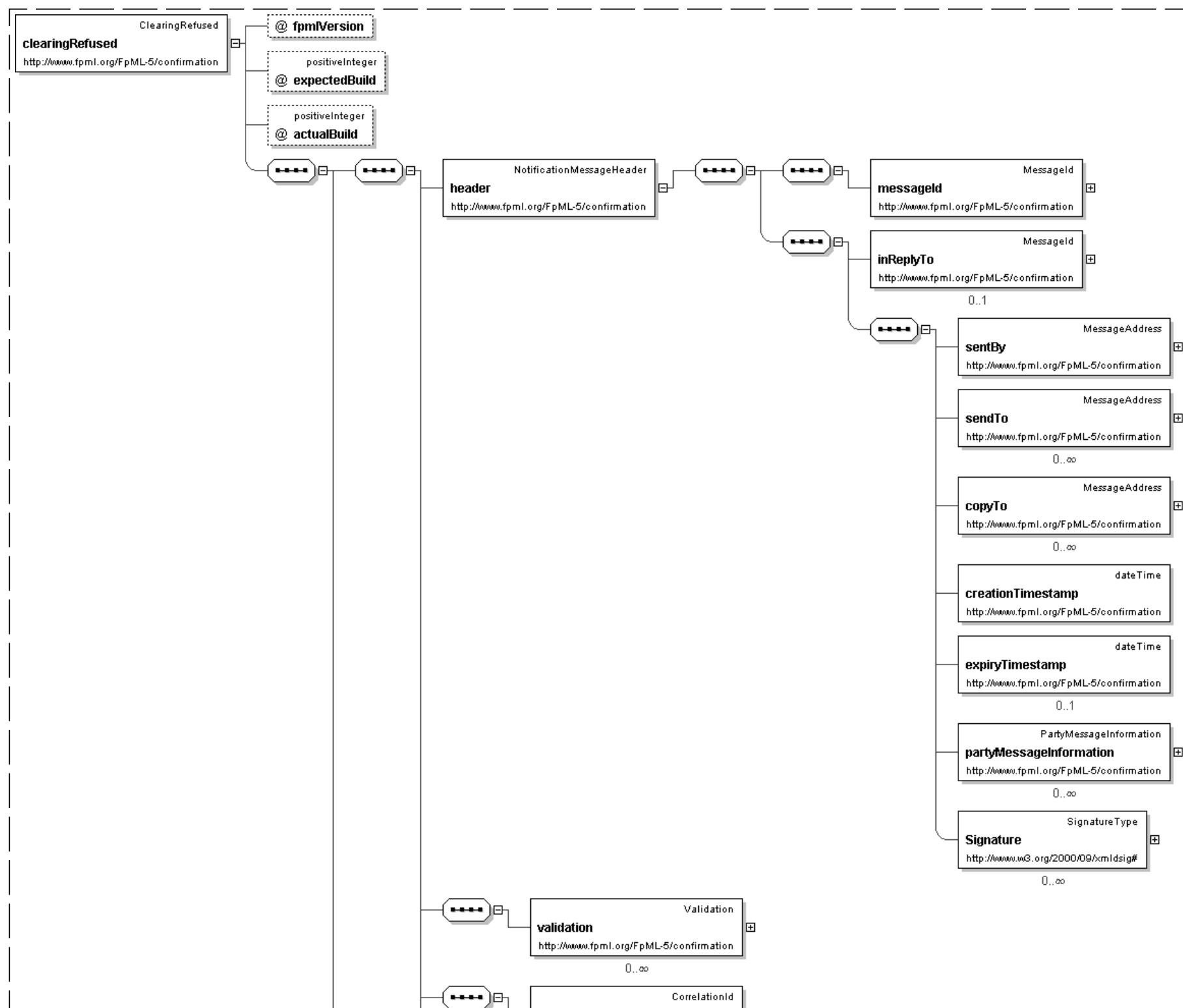
```
<xsd:element name="clearingException" type="#Exception" />
```

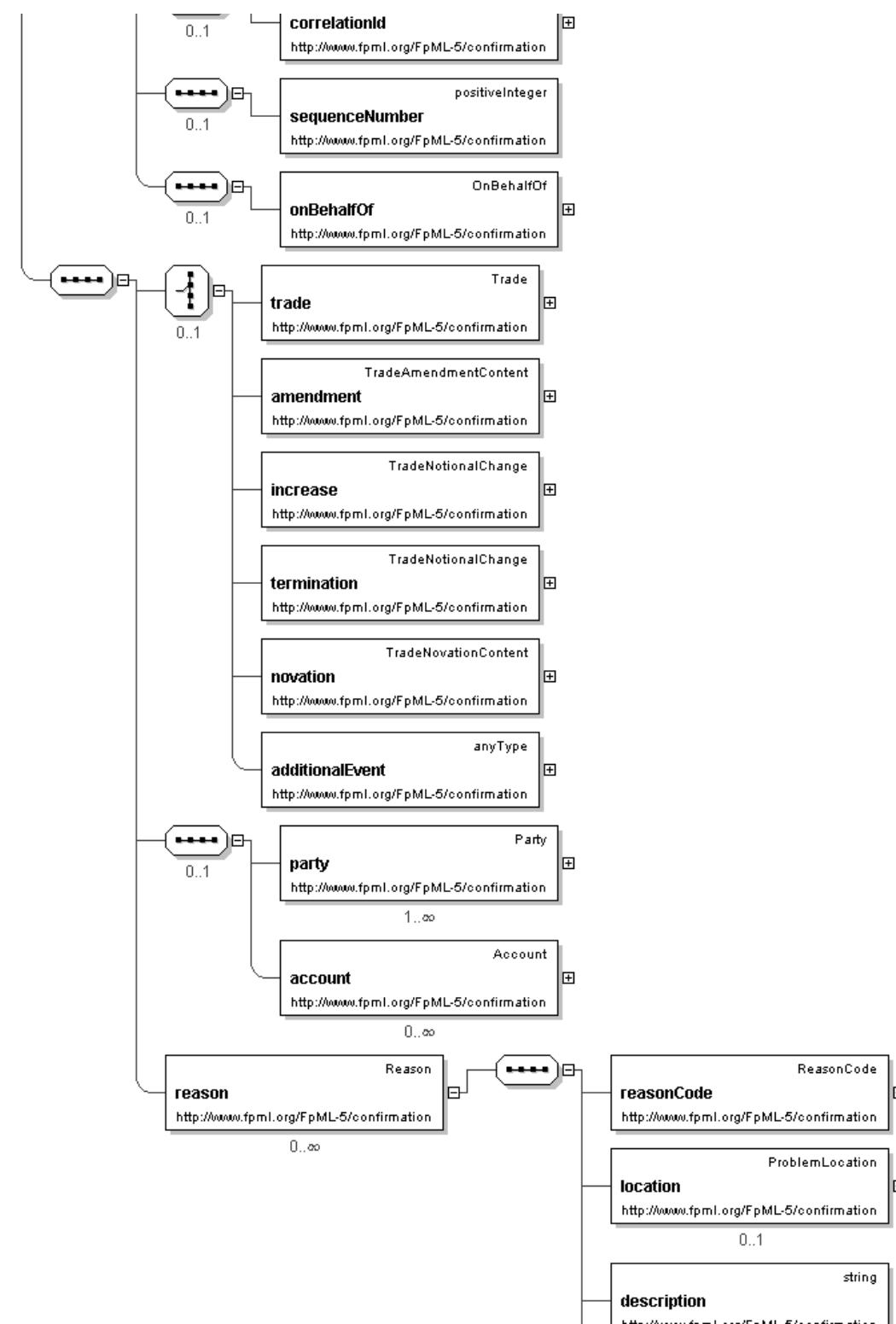
top

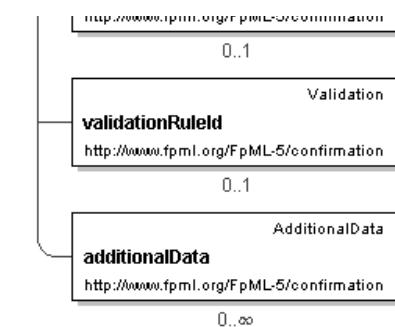
**Element: clearingRefused**

Name	clearingRefused
Type	ClearingRefused
Nillable	no
Abstract	no

**Logical Diagram**





**XML Instance Representation**

```

<clearingRefused
fpmlVersion=" xsd:token (value comes from list: {'5-0'}) [1]
'Indicate which version of the FpML Schema an FpML message adheres to.'

"
expectedBuild=" xsd:positiveInteger [0..1]
'This optional attribute can be supplied by a message creator in an FpML instance to
specify which build number of the schema was used to define the message when it was generated.

"
actualBuild="8 [0..1]

```

*'The specific build number of this schema version. This attribute is not included in an instance document. Instead, it is supplied by the XML parser when the document is validated against the FpML schema and indicates the build number of the schema file. Every time FpML publishes a change to the schema, validation rules, or examples within a version (e.g., version 4.2) the actual build number is incremented. If no changes have been made between releases within a version (i.e. from Trial Recommendation to Recommendation) the actual build number stays the same.'*

```

">
<header> NotificationMessageHeader </header> [1]
<validation> Validation </validation> [0..*]
Start Group: Correlation.model [0..1]
<correlationId> CorrelationId </correlationId> [1]
'A qualified identifier used to correlate between messages'

End Group: Correlation.model
Start Group: Sequence.model [0..1]
<sequenceNumber> xsd:positiveInteger </sequenceNumber> [1]
'A numeric value that can be used to order messages with the same correlation identifier
from the same sender.'

```

```

End Group: Sequence.model
Start Group: OnBehalfOf.model [0..1]
<onBehalfOf> OnBehalfOf </onBehalfOf> [1]
'Indicates which party (and accounts) a trade is being processed for.'

```

```

End Group: OnBehalfOf.model
Start Group: Events.model [0..1]
Start Choice [1]
<trade> Trade </trade> [1]
<amendment> TradeAmendmentContent </amendment> [1]

```

```

<increase> TradeNotionalChange </increase> [1]
<termination> TradeNotionalChange </termination> [1]
<novation> TradeNovationContent </novation> [1]
<additionalEvent> ... </additionalEvent> [1]

```

End Choice

End Group: Events.model

Start Group: PartiesAndAccounts.model [0..1]

&lt;party&gt; Party &lt;/party&gt; [1..\*]

'A legal entity or a subdivision of a legal entity.', 'Parties can perform multiple roles in a trade lifecycle. For example, the principal parties obligated to make payments from time to time during the term of the trade, but may include other parties involved in, or incidental to, the trade, such as parties acting in the role of novation transferor/transferee, broker, calculation agent, etc. In FpML roles are defined in multiple places within a document.'

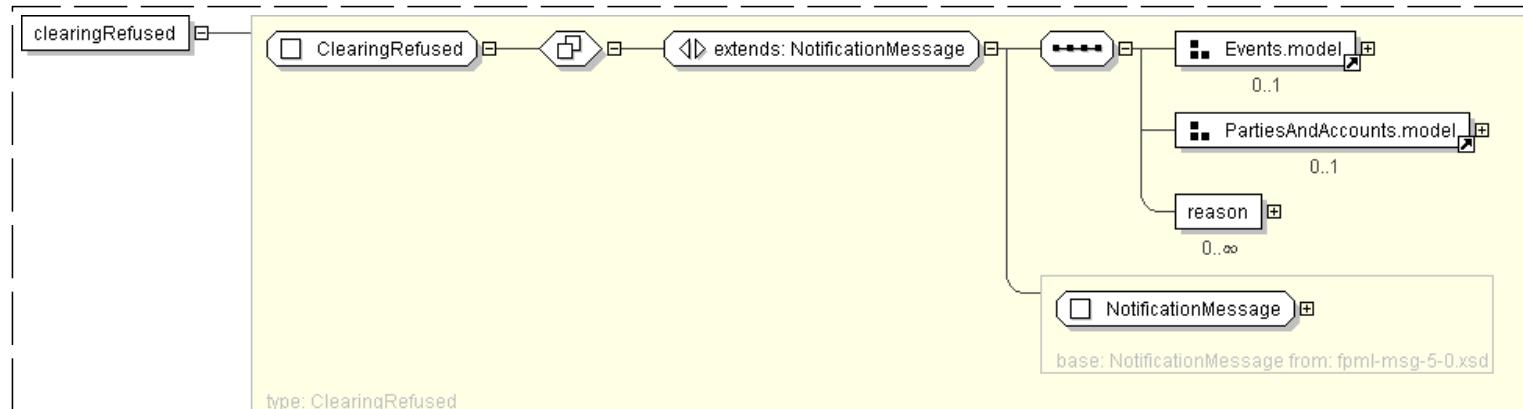
&lt;account&gt; Account &lt;/account&gt; [0..\*]

'Optional account information used to precisely define the origination and destination of financial instruments.'

End Group: PartiesAndAccounts.model

&lt;reason&gt; Reason &lt;/reason&gt; [0..\*]

&lt;/clearingRefused&gt;

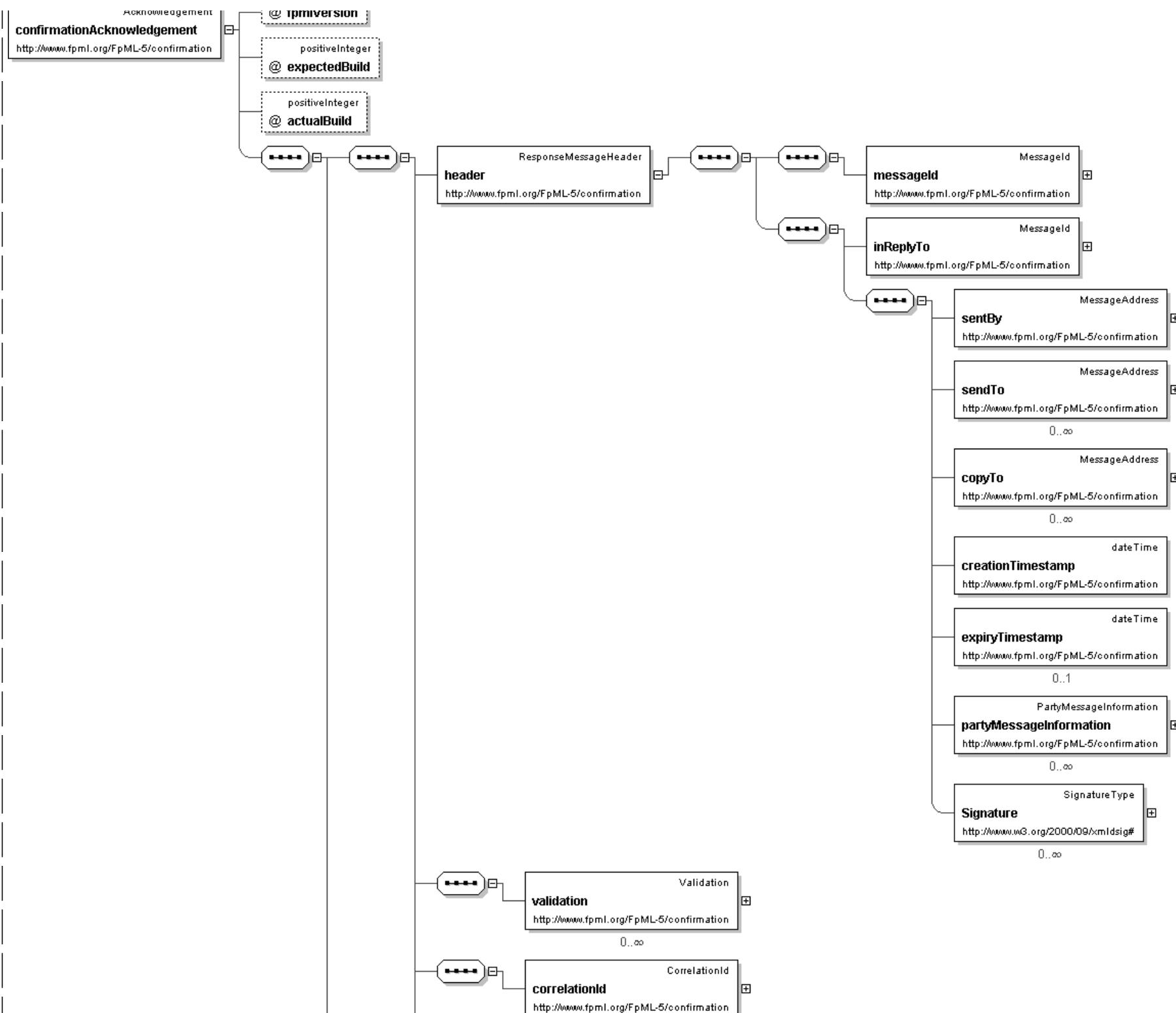
**Diagram****Schema Component Representation**

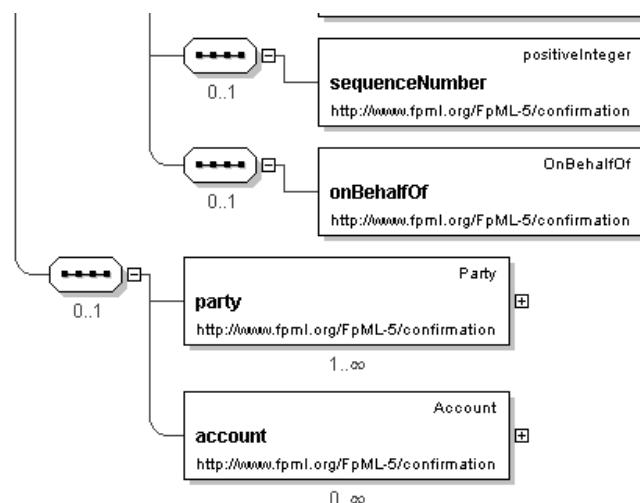
```
<xsd:element name="clearingRefused" type=" ClearingRefused "/>
```

[top](#)**Element: confirmationAcknowledgement**

Name	confirmationAcknowledgement
Type	Acknowledgement
Nillable	no
Abstract	no

**Logical Diagram**



**XML Instance Representation**

```

<confirmationAcknowledgement
fpmlVersion=" xsd:token (value comes from list: {'5-0'}) [1]
'Indicate which version of the FpML Schema an FpML message adheres to.'

"
expectedBuild=" xsd:positiveInteger [0..1]
'This optional attribute can be supplied by a message creator in an FpML instance to
specify which build number of the schema was used to define the message when it was generated.'

"
actualBuild="8 [0..1]
'The specific build number of this schema version. This attribute is not included in
an instance document. Instead, it is supplied by the XML parser when the document is
validated against the FpML schema and indicates the build number of the schema file. Every
time FpML publishes a change to the schema, validation rules, or examples within a version
(e.g., version 4.2) the actual build number is incremented. If no changes have been
made between releases within a version (i.e. from Trial Recommendation to Recommendation)
the actual build number stays the same.'

">
<header> ResponseMessageHeader </header> [1]
<validation> Validation </validation> [0..*]
<correlationId> CorrelationId </correlationId> [1]
'A qualified identifier used to correlate between messages'

Start Group: Sequence.model [0..1]
<sequenceNumber> xsd:positiveInteger </sequenceNumber> [1]
'A numeric value that can be used to order messages with the same correlation identifier
from the same sender.'

End Group: Sequence.model
Start Group: OnBehalfOf.model [0..1]
<onBehalfOf> OnBehalfOf </onBehalfOf> [1]
'Indicates which party (and accounts) a trade is being processed for.'

End Group: OnBehalfOf.model
  
```

Start Group: [PartiesAndAccounts.model](#) [0..1]

<party> [Party](#) </party> [1..\*]

'A legal entity or a subdivision of a legal entity.', 'Parties can perform multiple roles in a trade lifecycle. For example, the principal parties obligated to make payments from time to time during the term of the trade, but may include other parties involved in, or incidental to, the trade, such as parties acting in the role of novation transferor/transferee, broker, calculation agent, etc. In FpML roles are defined in multiple places within a document.'

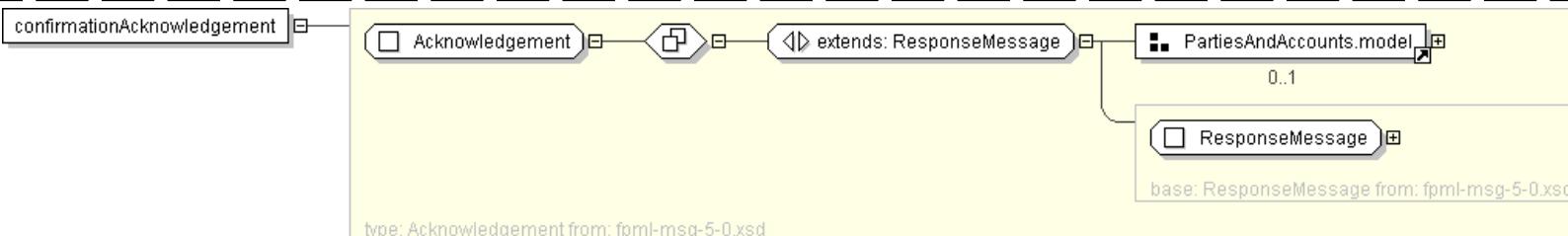
<account> [Account](#) </account> [0..\*]

'Optional account information used to precisely define the origination and destination of financial instruments.'

End Group: [PartiesAndAccounts.model](#)

</confirmationAcknowledgement>

#### Diagram



#### Schema Component Representation

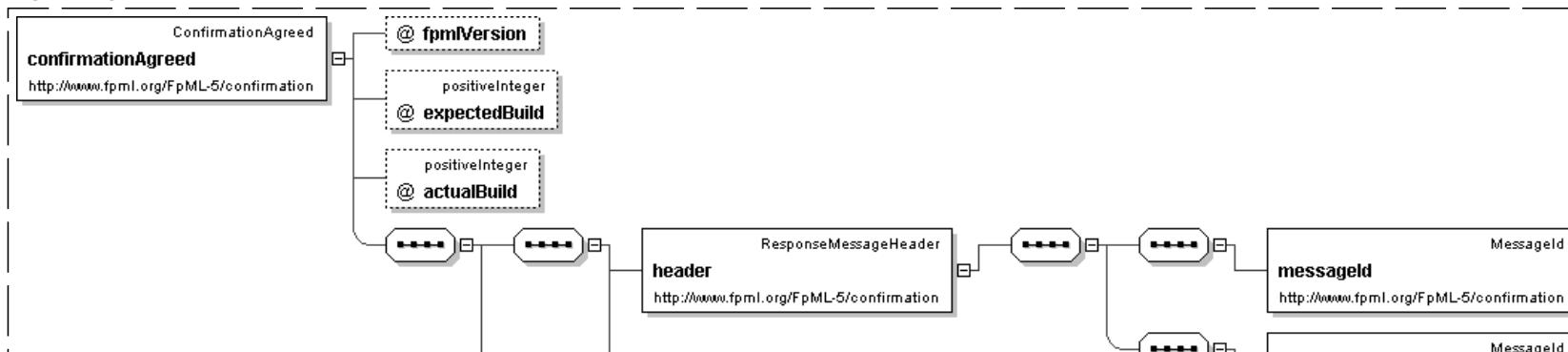
```
<xsd:element name="confirmationAcknowledgement" type="Acknowledgement" />
```

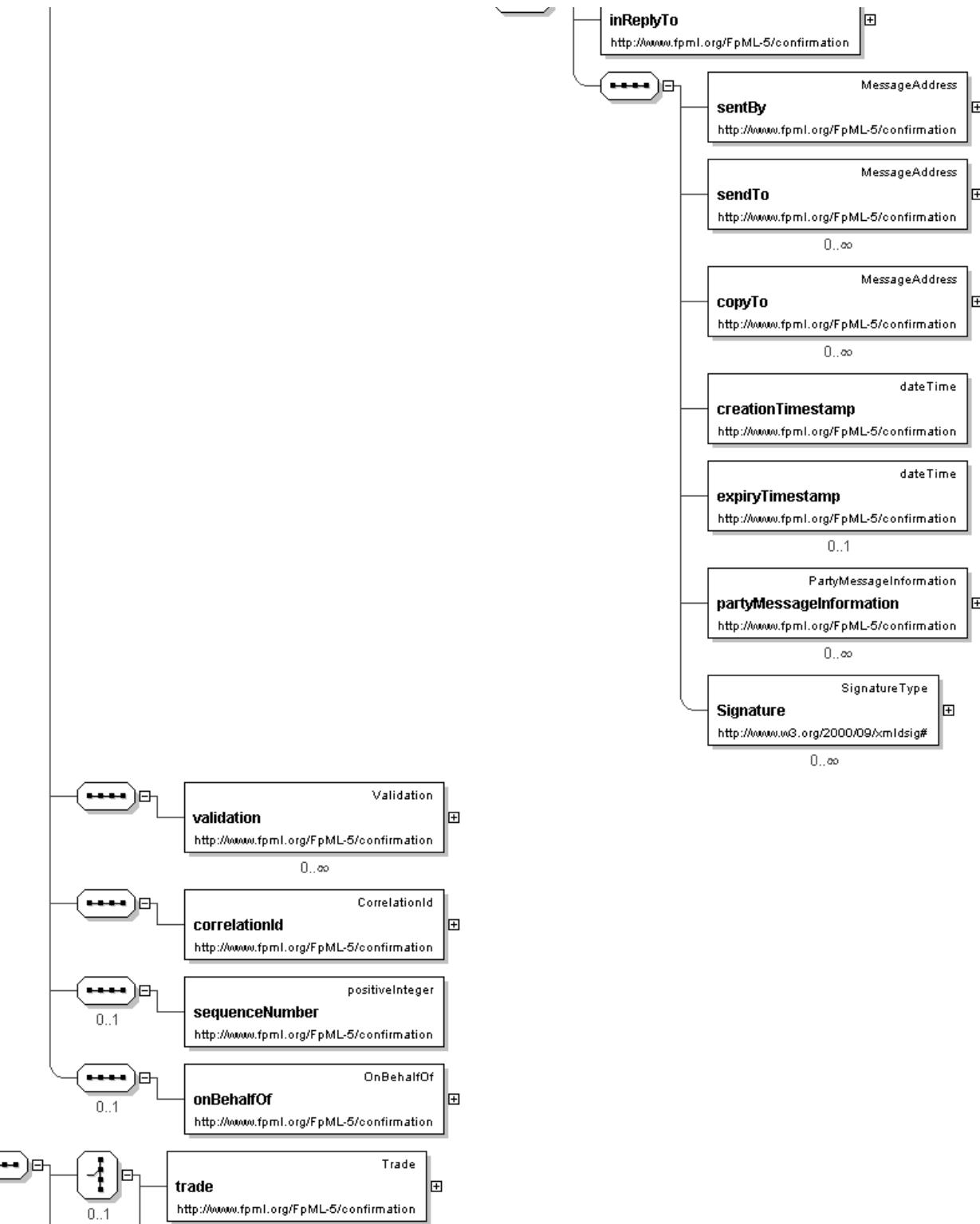
[top](#)

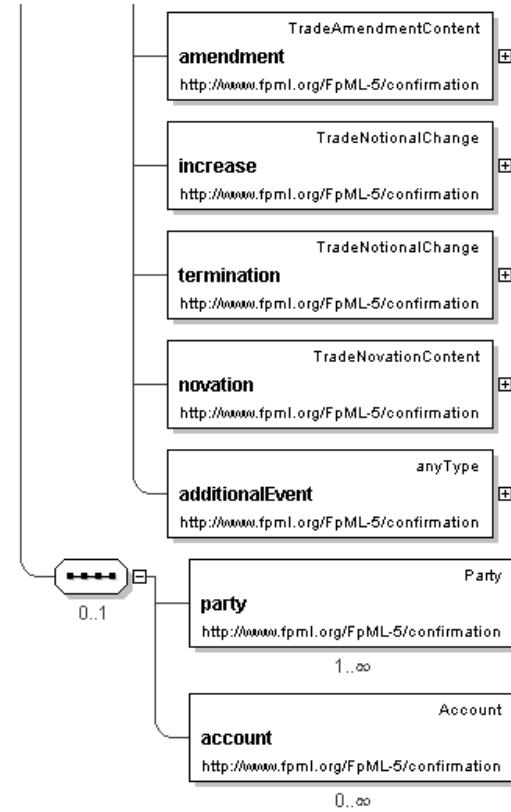
## Element: confirmationAgreed

Name	confirmationAgreed
Type	<a href="#">ConfirmationAgreed</a>
Nillable	no
Abstract	no

#### Logical Diagram







### XML Instance Representation

```

<confirmationAgreed
  fpmlVersion=" xsd:token (value comes from list: {'5-0'}) [1]
  'Indicate which version of the FpML Schema an FpML message adheres to.'

  "
  expectedBuild=" xsd:positiveInteger [0..1]
  'This optional attribute can be supplied by a message creator in an FpML instance to
  specify which build number of the schema was used to define the message when it was generated.'

  "
  actualBuild="8 [0..1]
  'The specific build number of this schema version. This attribute is not included in
  an instance document. Instead, it is supplied by the XML parser when the document is
  validated against the FpML schema and indicates the build number of the schema file. Every
  time FpML publishes a change to the schema, validation rules, or examples within a version
  (e.g., version 4.2) the actual build number is incremented. If no changes have been
  made between releases within a version (i.e. from Trial Recommendation to Recommendation)
  the actual build number stays the same.'


  ">
  <header> ResponseMessageHeader </header> [1]
  <validation> Validation </validation> [0..*]
  <correlationId> CorrelationId </correlationId> [1]
  'A qualified identifier used to correlate between messages'

```

Start Group: Sequence.model [0..1]  
 <sequenceNumber> xsd:positiveInteger </sequenceNumber> [1]  
*'A numeric value that can be used to order messages with the same correlation identifier from the same sender.'*

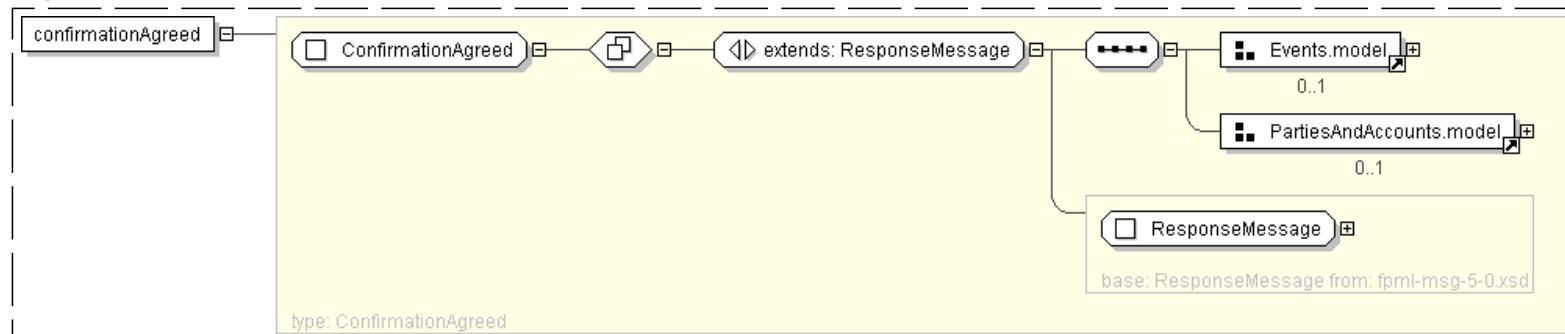
End Group: Sequence.model  
 Start Group: OnBehalfOf.model [0..1]  
 <onBehalfOf> OnBehalfOf </onBehalfOf> [1]  
*'Indicates which party (and accounts) a trade is being processed for.'*

End Group: OnBehalfOf.model  
 Start Group: Events.model [0..1]  
 Start Choice [1]  
 <trade> Trade </trade> [1]  
 <amendment> TradeAmendmentContent </amendment> [1]  
 <increase> TradeNotionalChange </increase> [1]  
 <termination> TradeNotionalChange </termination> [1]  
 <novation> TradeNovationContent </novation> [1]  
 <additionalEvent> ... </additionalEvent> [1]

End Choice  
 End Group: Events.model  
 Start Group: PartiesAndAccounts.model [0..1]  
 <party> Party </party> [1..\*]  
*'A legal entity or a subdivision of a legal entity.', 'Parties can perform multiple roles in a trade lifecycle. For example, the principal parties obligated to make payments from time to time during the term of the trade, but may include other parties involved in, or incidental to, the trade, such as parties acting in the role of novation transferor/transferee, broker, calculation agent, etc. In FpML roles are defined in multiple places within a document.'*  
 <account> Account </account> [0..\*]  
*'Optional account information used to precisely define the origination and destination of financial instruments.'*

End Group: PartiesAndAccounts.model  
</confirmationAgreed>

#### Diagram

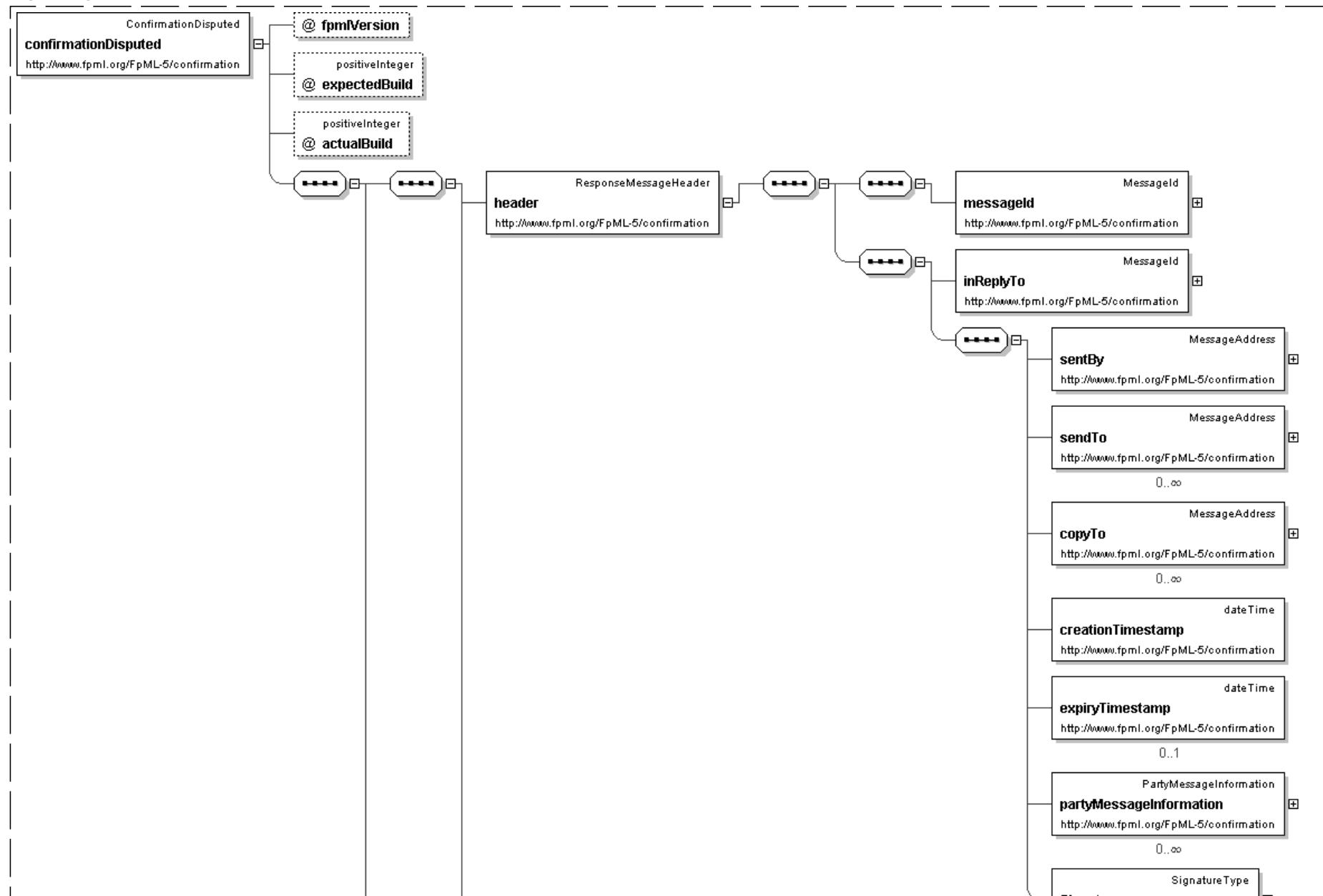


#### Schema Component Representation

```
<xsd:element name="confirmationAgreed" type=" ConfirmationAgreed "/>
```

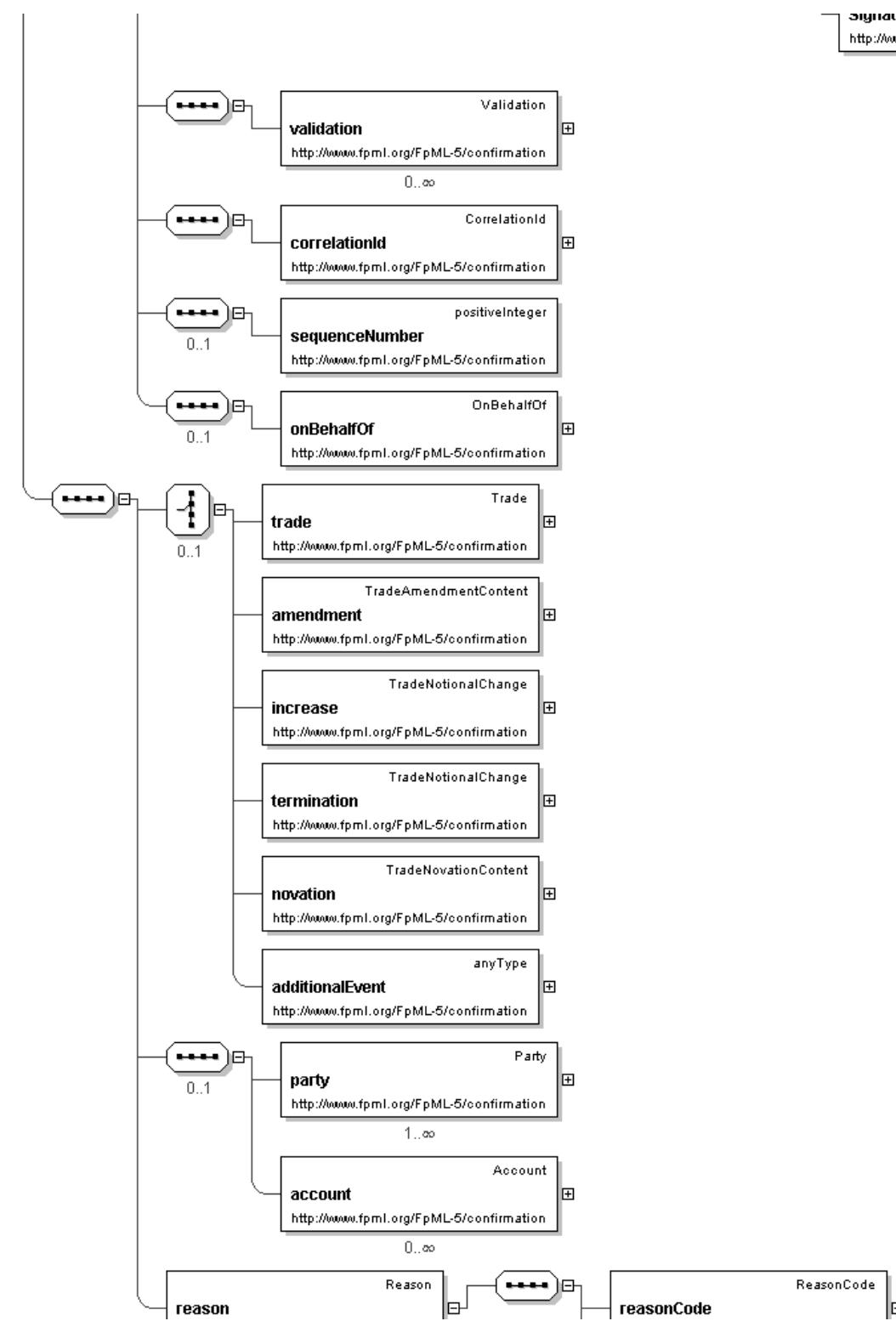
**Element: confirmationDisputed**

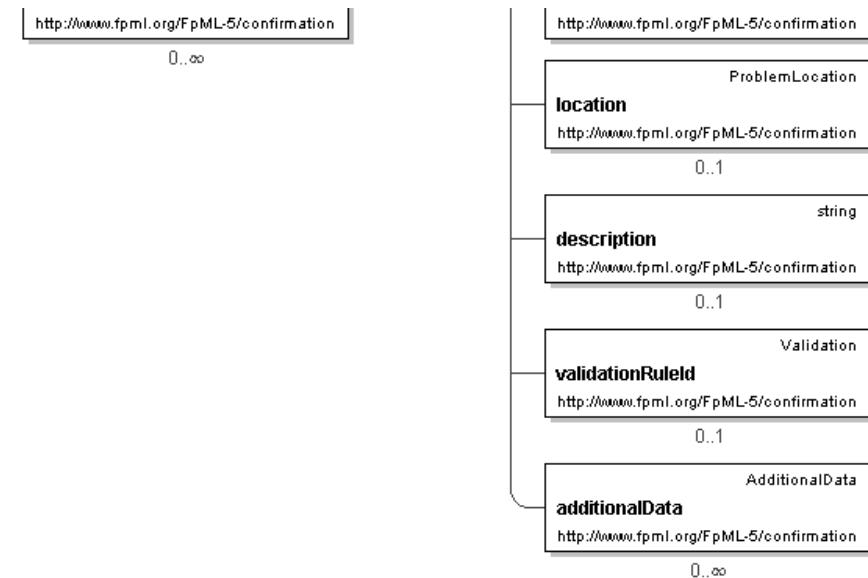
<b>Name</b>	confirmationDisputed
<b>Type</b>	<a href="#">ConfirmationDisputed</a>
<b>Nillable</b>	no
<b>Abstract</b>	no

**Logical Diagram**

**Signature**  
<http://www.w3.org/2000/09/xmldsig#>

0..oo



**XML Instance Representation**

```

<confirmationDisputed
fpmlVersion=" xsd:token (value comes from list: {'5-0'}) [1]
'Indicate which version of the FpML Schema an FpML message adheres to.'

"
expectedBuild=" xsd:positiveInteger [0..1]
'This optional attribute can be supplied by a message creator in an FpML instance to
specify which build number of the schema was used to define the message when it was generated.'

"
actualBuild="8 [0..1]
'The specific build number of this schema version. This attribute is not included in
an instance document. Instead, it is supplied by the XML parser when the document is
validated against the FpML schema and indicates the build number of the schema file. Every
time FpML publishes a change to the schema, validation rules, or examples within a version
(e.g., version 4.2) the actual build number is incremented. If no changes have been
made between releases within a version (i.e. from Trial Recommendation to Recommendation)
the actual build number stays the same.'

">
<header> ResponseMessageHeader </header> [1]
<validation> Validation </validation> [0..*]
<correlationId> CorrelationId </correlationId> [1]
'A qualified identifier used to correlate between messages'

Start Group: Sequence.model [0..1]
<sequenceNumber> xsd:positiveInteger </sequenceNumber> [1]
'A numeric value that can be used to order messages with the same correlation identifier
from the same sender.'

End Group: Sequence.model
Start Group: OnBehalfOf.model [0..1]
<onBehalfOf> OnBehalfOf </onBehalfOf> [1]

```

'Indicates which party (and accounts) a trade is being processed for.'

End Group: OnBehalfOf.model

Start Group: Events.model [0..1]

Start Choice [1]

```
<trade> Trade </trade> [1]
<amendment> TradeAmendmentContent </amendment> [1]
<increase> TradeNotionalChange </increase> [1]
<termination> TradeNotionalChange </termination> [1]
<novation> TradeNovationContent </novation> [1]
<additionalEvent> ... </additionalEvent> [1]
```

End Choice

End Group: Events.model

Start Group: PartiesAndAccounts.model [0..1]

<party> Party </party> [1..\*]

'A legal entity or a subdivision of a legal entity.', 'Parties can perform multiple roles in a trade lifecycle. For example, the principal parties obligated to make payments from time to time during the term of the trade, but may include other parties involved in, or incidental to, the trade, such as parties acting in the role of novation transferor/transferee, broker, calculation agent, etc. In FpML roles are defined in multiple places within a document.'

<account> Account </account> [0..\*]

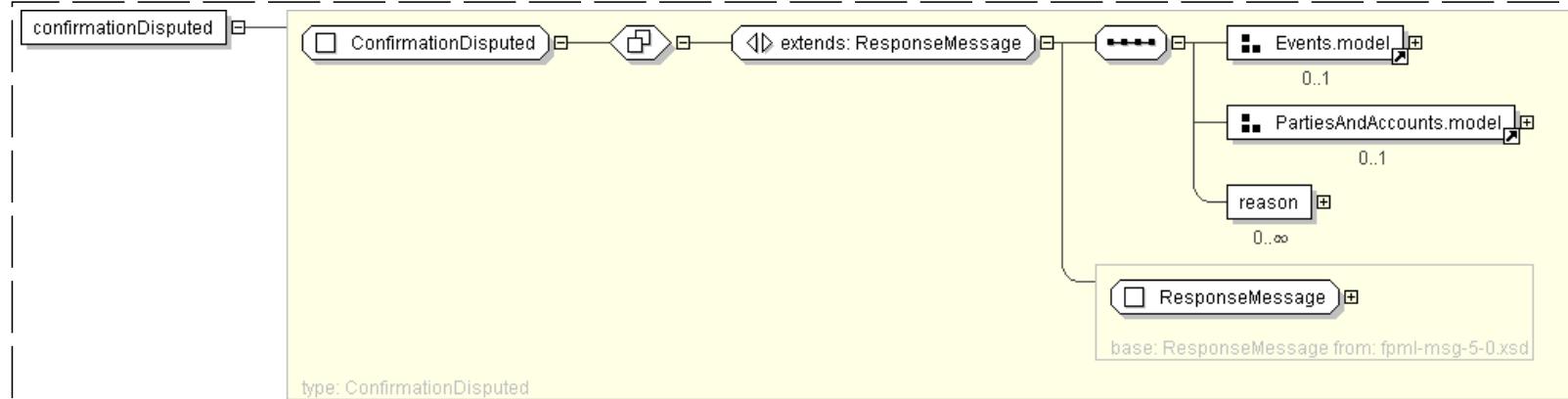
'Optional account information used to precisely define the origination and destination of financial instruments.'

End Group: PartiesAndAccounts.model

<reason> Reason </reason> [0..\*]

</confirmationDisputed>

#### Diagram



#### Schema Component Representation

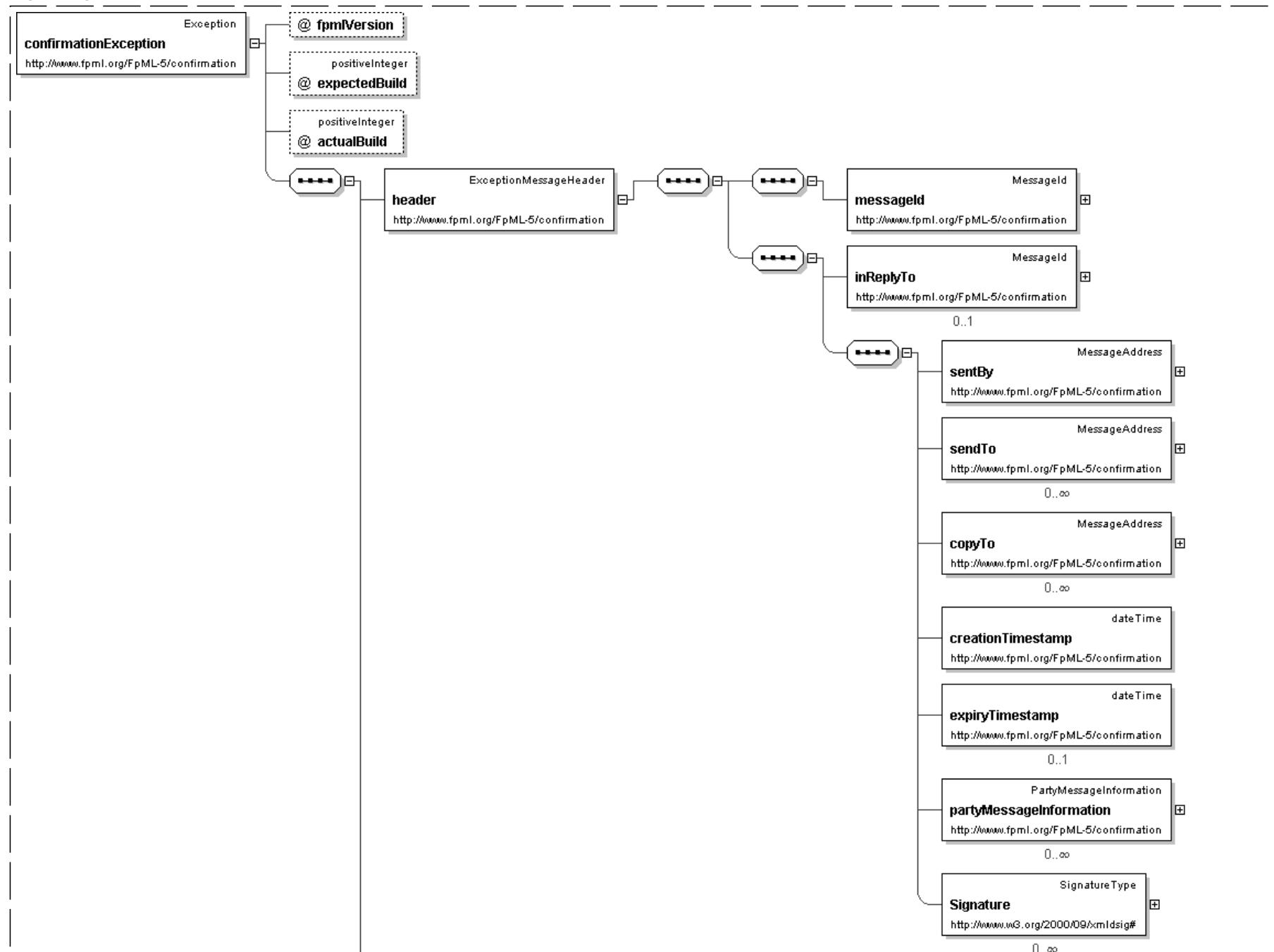
```
<xsd:element name="confirmationDisputed" type="ConfirmationDisputed" />
```

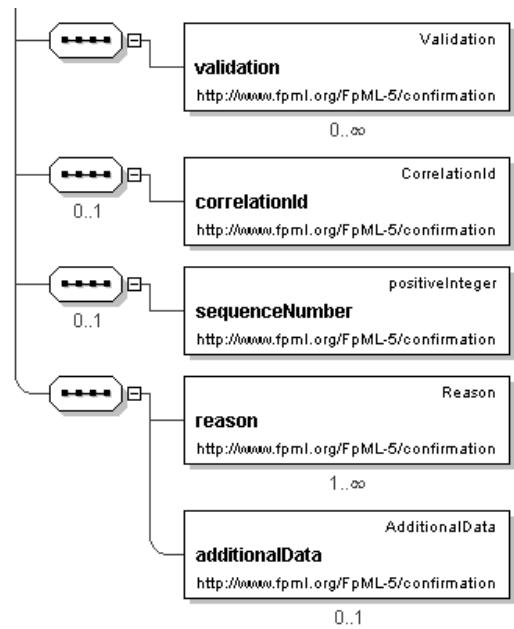
[top](#)

#### Element: confirmationException

Name	confirmationException
Type	Exception
Nillable	no
Abstract	no

## Logical Diagram



**XML Instance Representation**

```

<confirmationException
  fpmlVersion=" xsd:token (value comes from list: {'5-0'}) [1]
  'Indicate which version of the FpML Schema an FpML message adheres to.'

  "
  expectedBuild=" xsd:positiveInteger [0..1]
  'This optional attribute can be supplied by a message creator in an FpML instance to
  specify which build number of the schema was used to define the message when it was generated.'

  "
  actualBuild="8 [0..1]
  'The specific build number of this schema version. This attribute is not included in
  an instance document. Instead, it is supplied by the XML parser when the document is
  validated against the FpML schema and indicates the build number of the schema file. Every
  time FpML publishes a change to the schema, validation rules, or examples within a version
  (e.g., version 4.2) the actual build number is incremented. If no changes have been
  made between releases within a version (i.e. from Trial Recommendation to Recommendation)
  the actual build number stays the same.'

  ">
  <header> ExceptionMessageHeader </header> [1]
  <validation> Validation </validation> [0..*]
Start Group: Correlation.model [0..1]
  <correlationId> CorrelationId </correlationId> [1]
  'A qualified identifier used to correlate between messages'

End Group: Correlation.model
Start Group: Sequence.model [0..1]
  <sequenceNumber> xsd:positiveInteger </sequenceNumber> [1]
  'A numeric value that can be used to order messages with the same correlation identifier
  from the same sender.'

```

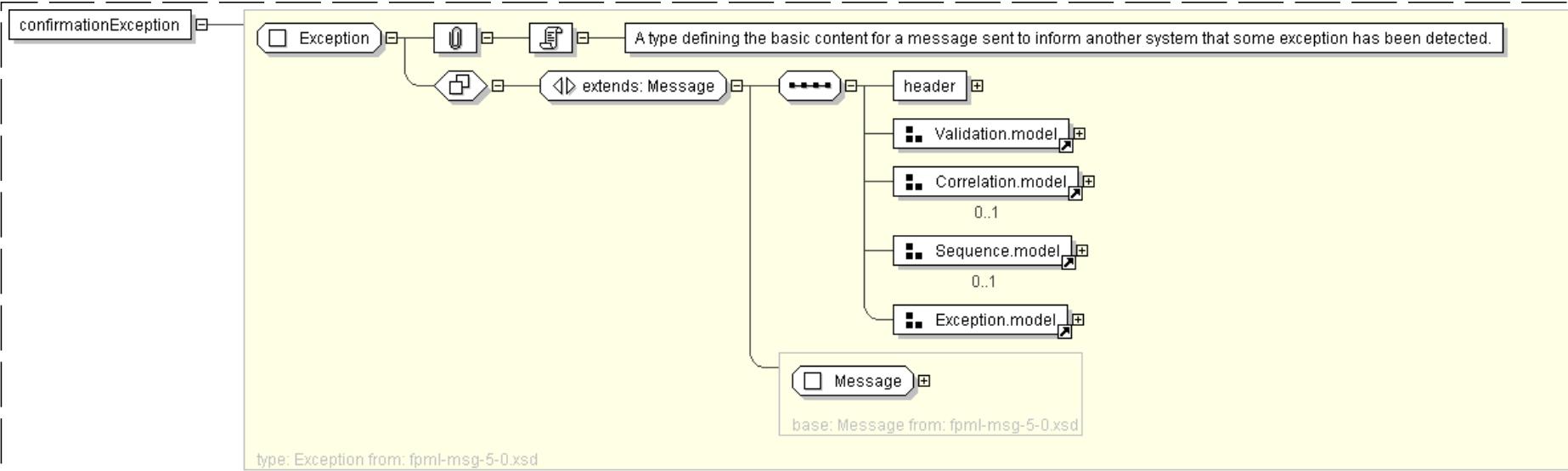
```

End Group: Sequence.model
<reason> Reason </reason> [1..*]
'An instance of the Reason type used to record the nature of any errors associated with
a message.'

<additionalData> AdditionalData </additionalData> [0..1]
'Any string of additional data that may help the message processor, for example in a
rejection message this might contain a code value or the text of the original request (within
a CDATA section).'

</confirmationException>

```

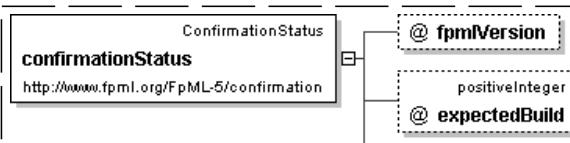
**Diagram****Schema Component Representation**

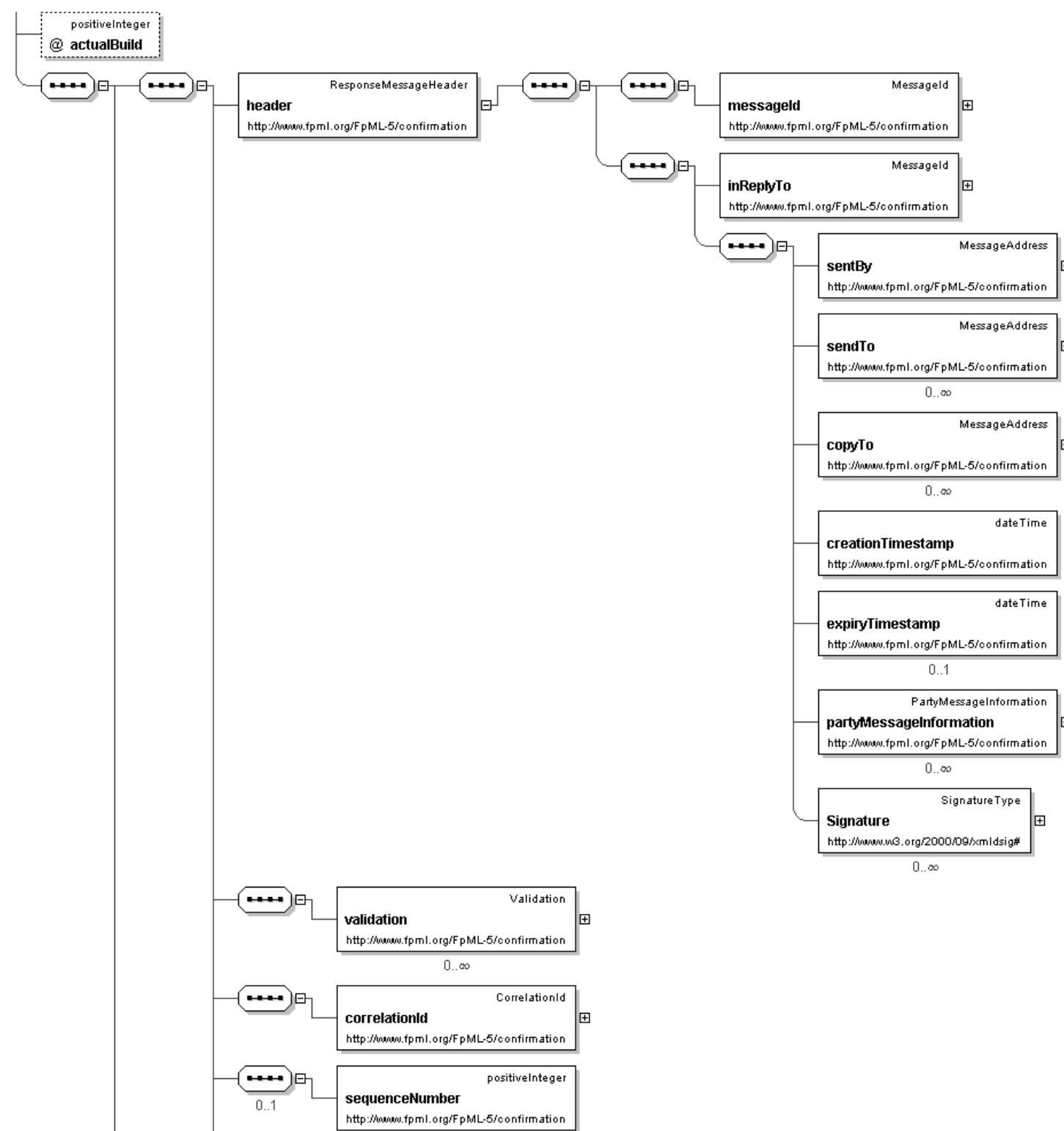
```
<xsd:element name="confirmationException" type=" Exception " />
```

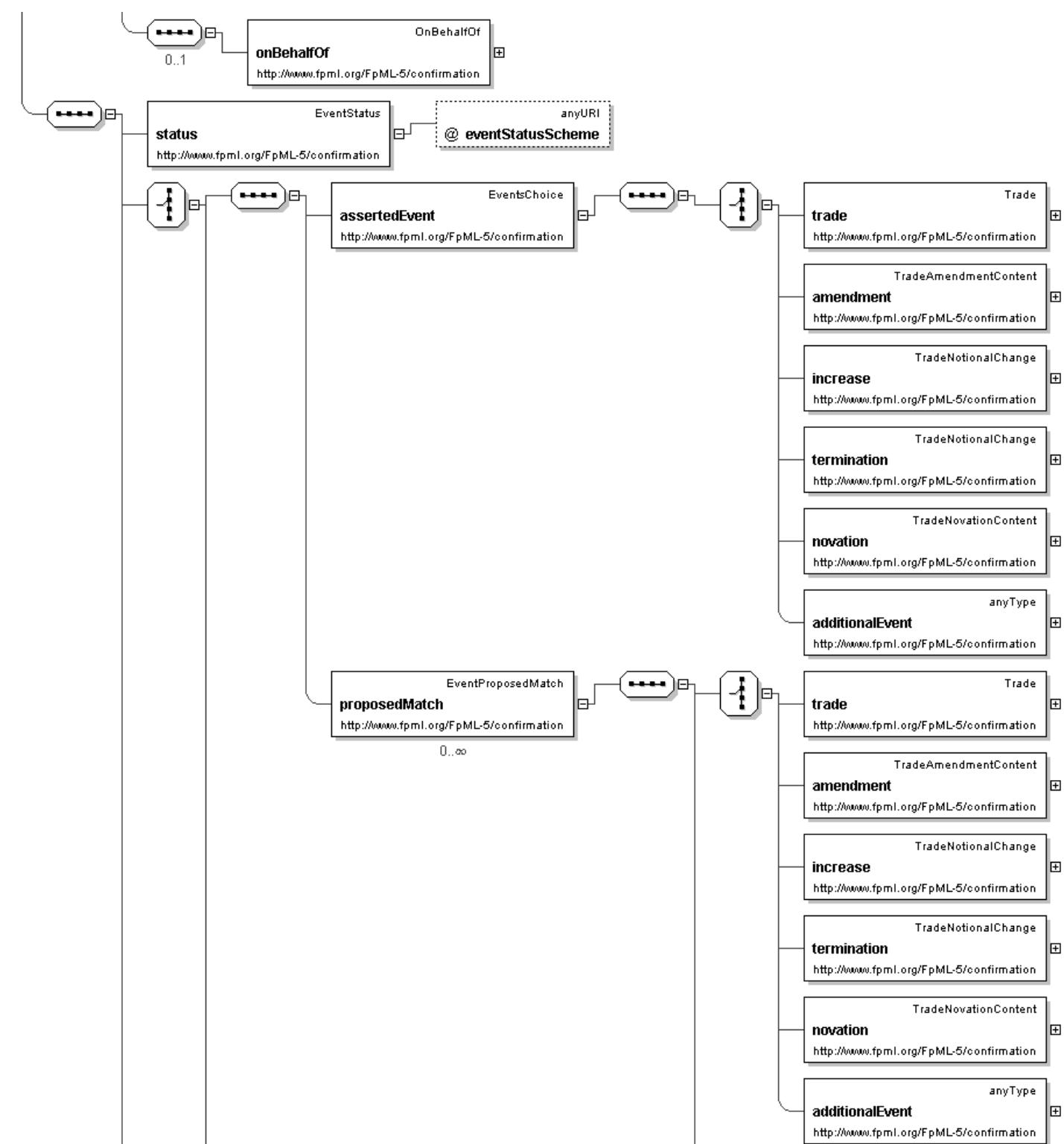
top

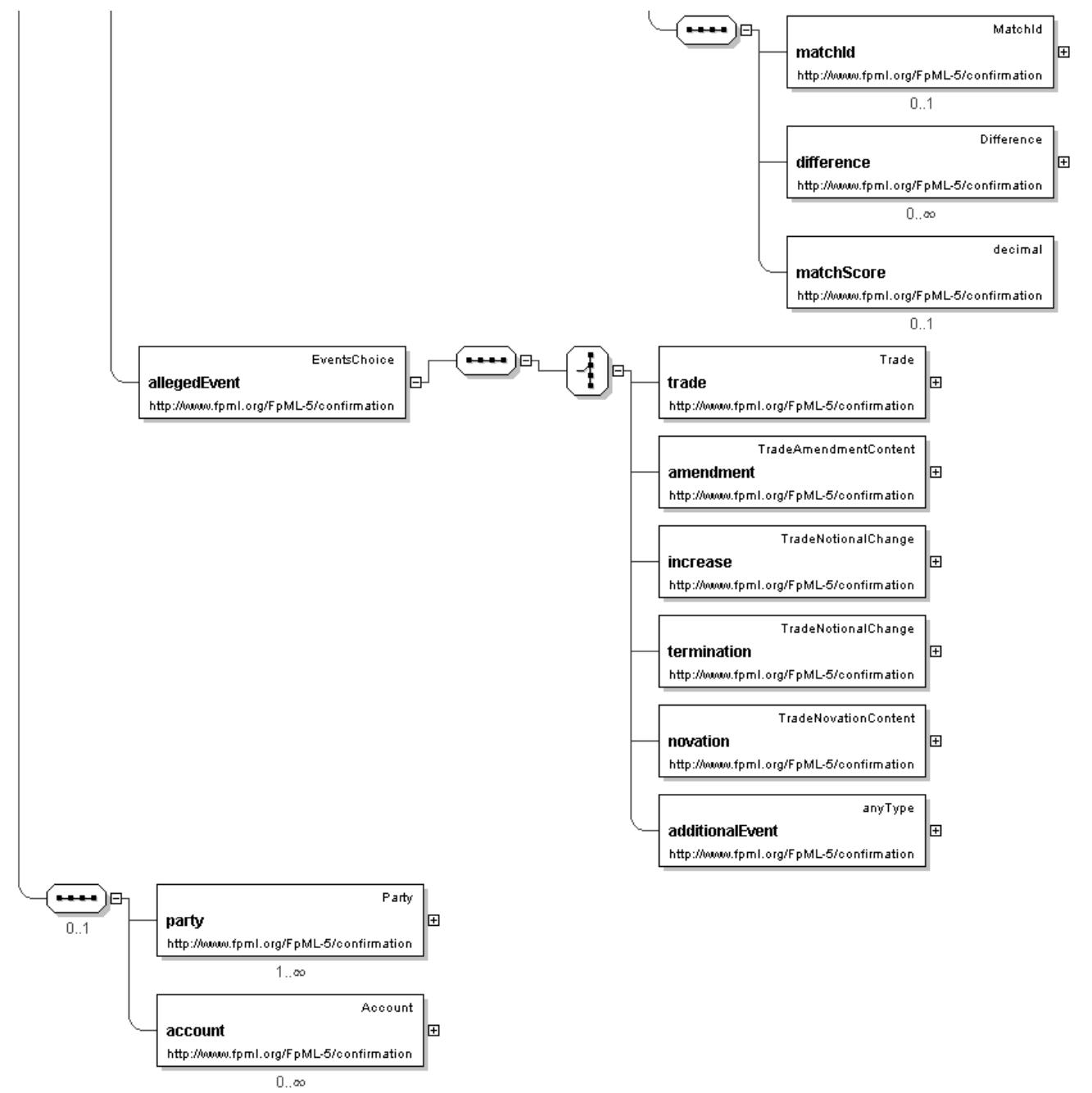
**Element: confirmationStatus**

Name	confirmationStatus
Type	<a href="#">ConfirmationStatus</a>
Nullable	no
Abstract	no

**Logical Diagram**





**XML Instance Representation**

```

<confirmationStatus
  fpmVersion="xsd:token (value comes from list: {'5-0'})" [1]
  'Indicate which version of the FpML Schema an FpML message adheres to.'
  " "
  
```

```

| expectedBuild= "xsd:positiveInteger [0..1]
| 'This optional attribute can be supplied by a message creator in an FpML instance to
| specify which build number of the schema was used to define the message when it was generated.'

"
actualBuild="8 [0..1]
'The specific build number of this schema version. This attribute is not included in
an instance document. Instead, it is supplied by the XML parser when the document is
validated against the FpML schema and indicates the build number of the schema file. Every
time FpML publishes a change to the schema, validation rules, or examples within a version
(e.g., version 4.2) the actual build number is incremented. If no changes have been
made between releases within a version (i.e. from Trial Recommendation to Recommendation)
the actual build number stays the same.'


">
<header> ResponseMessageHeader </header> [1]
<validation> Validation </validation> [0..*]
<correlationId> CorrelationId </correlationId> [1]
'A qualified identifier used to correlate between messages'

Start Group: Sequence.model [0..1]
<sequenceNumber> xsd:positiveInteger </sequenceNumber> [1]
'A numeric value that can be used to order messages with the same correlation identifier
from the same sender.'

End Group: Sequence.model
Start Group: OnBehalfOf.model [0..1]
<onBehalfOf> OnBehalfOf </onBehalfOf> [1]
'Indicates which party (and accounts) a trade is being processed for.'

End Group: OnBehalfOf.model
<status> EventStatus </status> [1]
'Defines the confirmation status of a trade or post-trade event (e.g. Matched,
Mismatched, Unmatched, Alleged).'

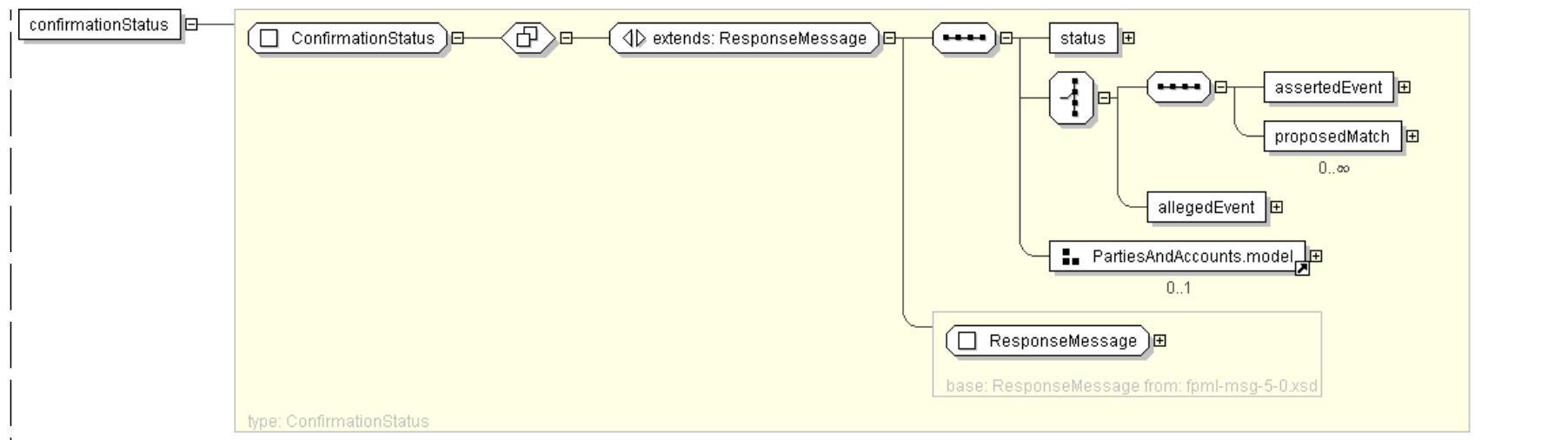
Start Choice [1]
<assertedEvent> EventsChoice </assertedEvent> [1]
<proposedMatch> EventProposedMatch </proposedMatch> [0..*]
<allegedEvent> EventsChoice </allegedEvent> [1]
End Choice
Start Group: PartiesAndAccounts.model [0..1]
<party> Party </party> [1..*]
'A legal entity or a subdivision of a legal entity.', 'Parties can perform multiple roles in
a trade lifecycle. For example, the principal parties obligated to make payments from time
to time during the term of the trade, but may include other parties involved in, or
incidental to, the trade, such as parties acting in the role of novation transferor/
transferee, broker, calculation agent, etc. In FpML roles are defined in multiple places
within a document.'


<account> Account </account> [0..*]
'Optional account information used to precisely define the origination and destination
of financial instruments.'

End Group: PartiesAndAccounts.model
</confirmationStatus>

```

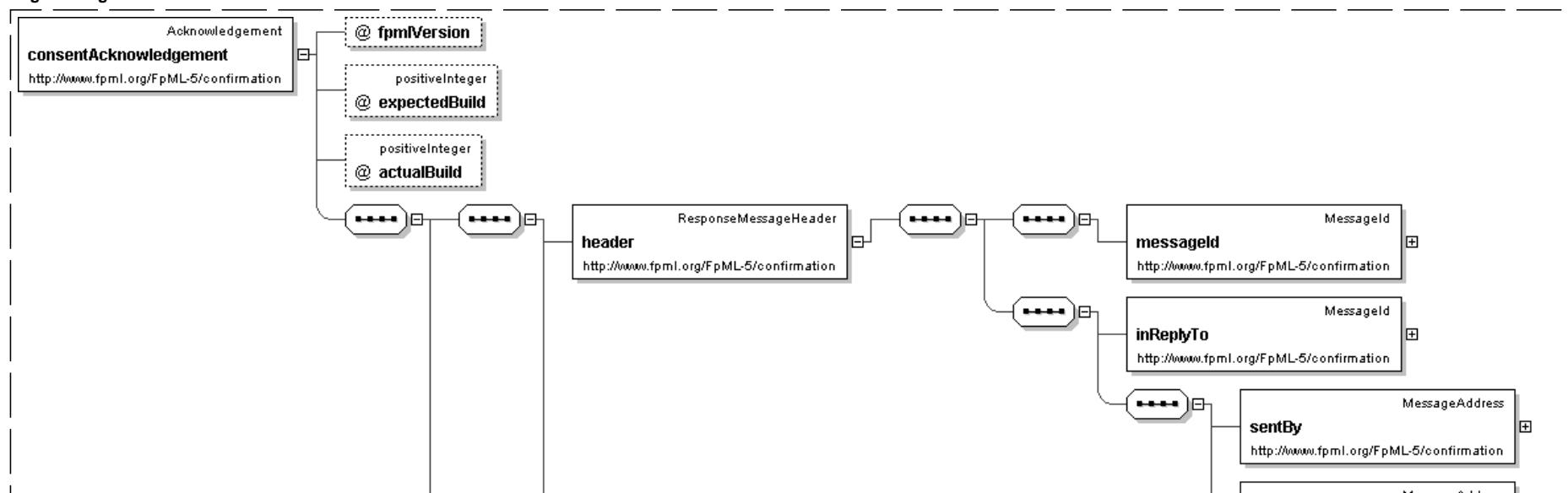
**Diagram**

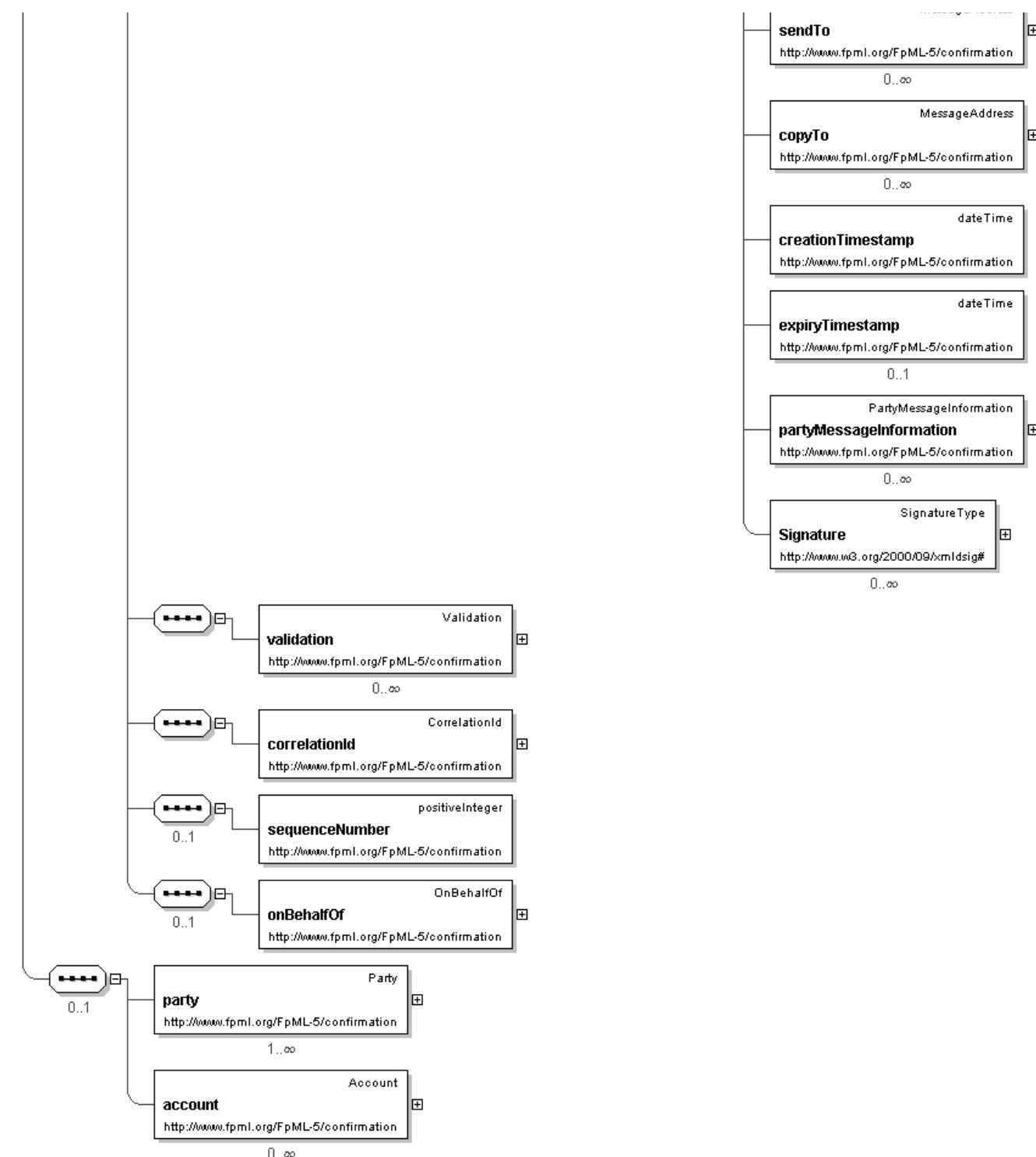
**Schema Component Representation**

```
<xsd:element name="confirmationStatus" type="ConfirmationStatus" />
```

[top](#)**Element: consentAcknowledgement**

Name	consentAcknowledgement
Type	<a href="#">Acknowledgement</a>
Nillable	no
Abstract	no

**Logical Diagram**



**XML Instance Representation**

```

<consentAcknowledgement
  fpmVersion=" xsd:token (value comes from list: {'5-0'}) [1]
  'Indicate which version of the FpML Schema an FpML message adheres to.'


  "
  expectedBuild=" xsd:positiveInteger [0..1]
  'This optional attribute can be supplied by a message creator in an FpML instance to
  specify which build number of the schema was used to define the message when it was generated.'


  "
  actualBuild="8 [0..1]
  'The specific build number of this schema version. This attribute is not included in
  an instance document. Instead, it is supplied by the XML parser when the document is
  validated against the FpML schema and indicates the build number of the schema file. Every
  time FpML publishes a change to the schema, validation rules, or examples within a version
  (e.g., version 4.2) the actual build number is incremented. If no changes have been
  made between releases within a version (i.e. from Trial Recommendation to Recommendation)
  the actual build number stays the same.'


  ">
    <header> ResponseMessageHeader </header> [1]
    <validation> Validation </validation> [0..*]
    <correlationId> CorrelationId </correlationId> [1]
    'A qualified identifier used to correlate between messages'


Start Group: Sequence.model [0..1]
  <sequenceNumber> xsd:positiveInteger </sequenceNumber> [1]
  'A numeric value that can be used to order messages with the same correlation identifier
  from the same sender.'


End Group: Sequence.model
Start Group: OnBehalfOf.model [0..1]
  <onBehalfOf> OnBehalfOf </onBehalfOf> [1]
  'Indicates which party (and accounts) a trade is being processed for.'

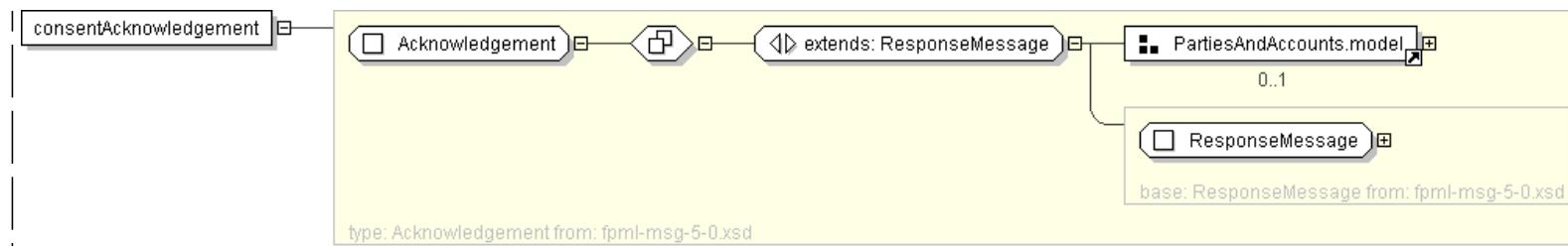

End Group: OnBehalfOf.model
Start Group: PartiesAndAccounts.model [0..1]
  <party> Party </party> [1..*]
  'A legal entity or a subdivision of a legal entity.', 'Parties can perform multiple roles in
  a trade lifecycle. For example, the principal parties obligated to make payments from time
  to time during the term of the trade, but may include other parties involved in, or
  incidental to, the trade, such as parties acting in the role of novation transferor/
  transferee, broker, calculation agent, etc. In FpML roles are defined in multiple places
  within a document.'


  <account> Account </account> [0..*]
  'Optional account information used to precisely define the origination and destination
  of financial instruments.'


End Group: PartiesAndAccounts.model
</consentAcknowledgement>

```

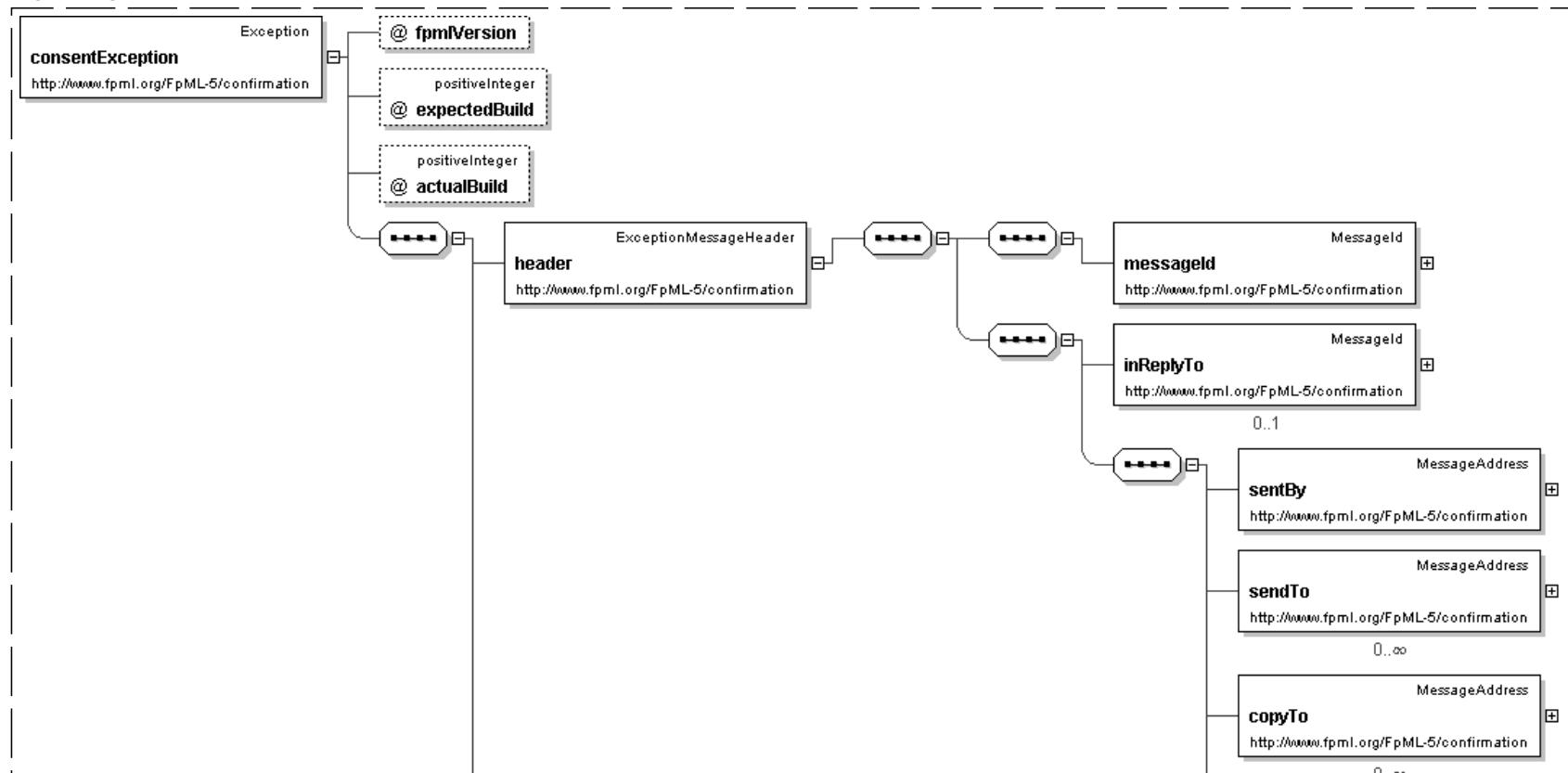
**Diagram**

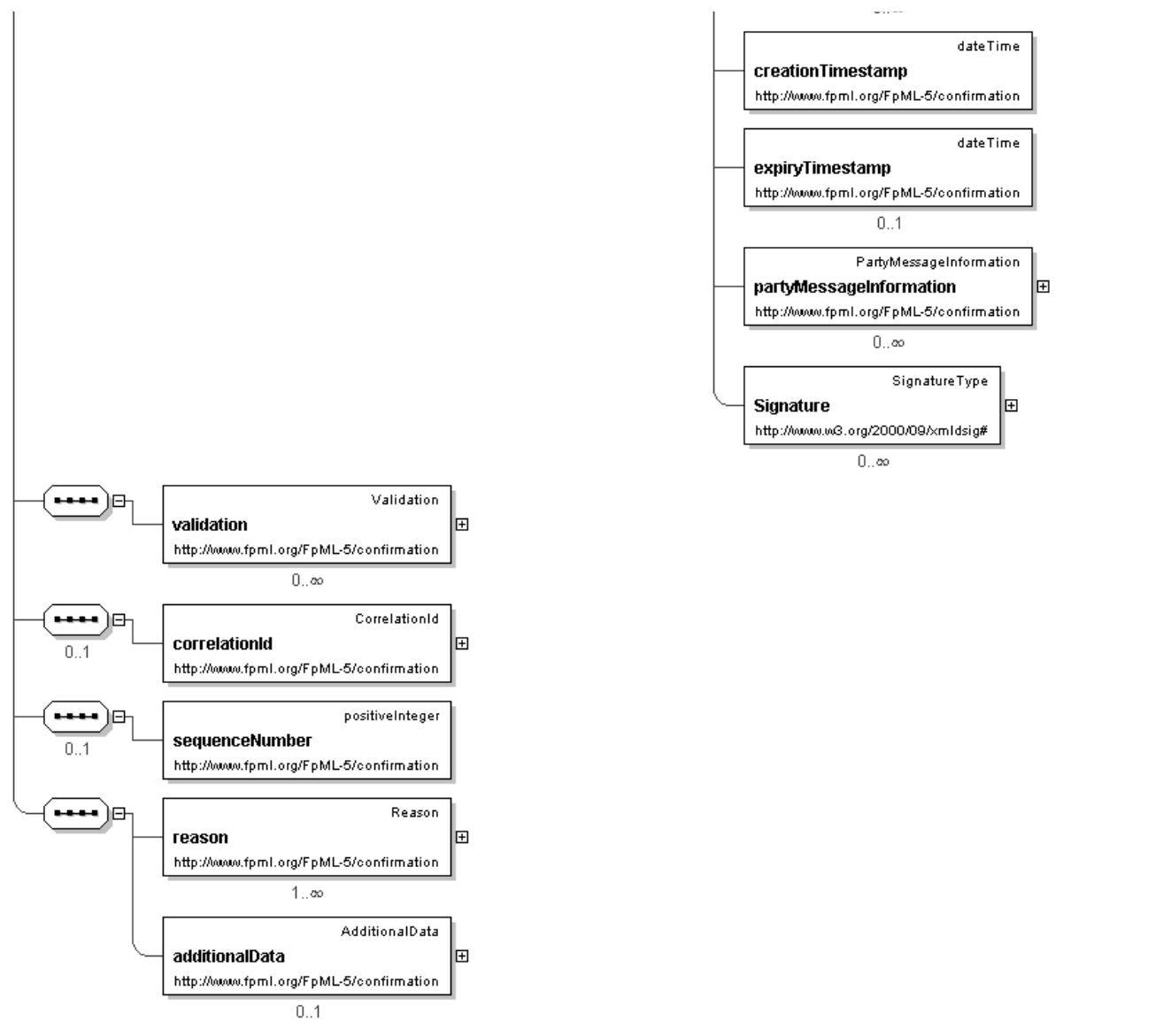
**Schema Component Representation**

```
<xsd:element name="consentAcknowledgement" type=" Acknowledgement " />
```

[top](#)**Element: consentException**

Name	consentException
Type	<a href="#">Exception</a>
Nullable	no
Abstract	no

**Logical Diagram**



#### XML Instance Representation

```

<consentException
fpmlVersion=" xsd:token (value comes from list: {'5-0'}) [1]
'Indicate which version of the FpML Schema an FpML message adheres to.'

"
expectedBuild=" xsd:positiveInteger [0..1]
'This optional attribute can be supplied by a message creator in an FpML instance to
specify which build number of the schema was used to define the message when it was generated.'

"
actualBuild="8 [0..1]
  
```

*The specific build number of this schema version. This attribute is not included in an instance document. Instead, it is supplied by the XML parser when the document is validated against the FpML schema and indicates the build number of the schema file. Every time FpML publishes a change to the schema, validation rules, or examples within a version (e.g., version 4.2) the actual build number is incremented. If no changes have been made between releases within a version (i.e. from Trial Recommendation to Recommendation) the actual build number stays the same.'*

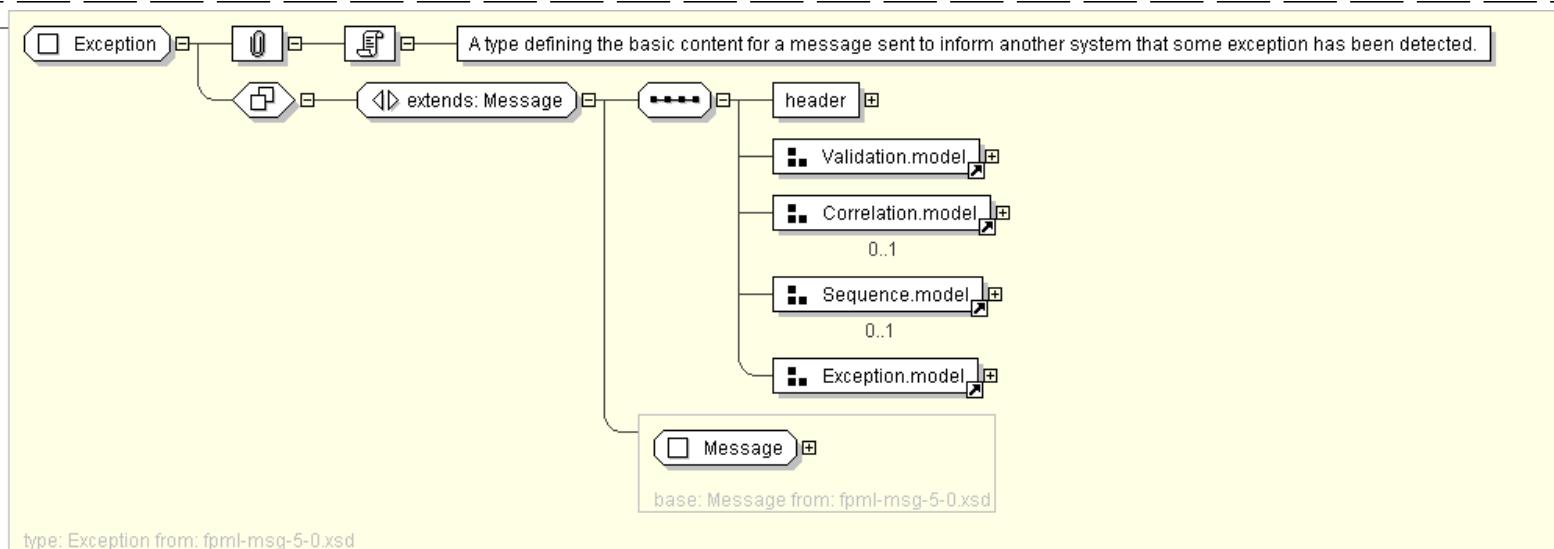
```
">   <header> ExceptionMessageHeader </header> [1]
  <validation> Validation </validation> [0..*]
Start Group: Correlation.model [0..1]
  <correlationId> CorrelationId </correlationId> [1]
    'A qualified identifier used to correlate between messages'

End Group: Correlation.model
Start Group: Sequence.model [0..1]
  <sequenceNumber> xsd:positiveInteger </sequenceNumber> [1]
    'A numeric value that can be used to order messages with the same correlation identifier from the same sender.'

End Group: Sequence.model
  <reason> Reason </reason> [1..*]
    'An instance of the Reason type used to record the nature of any errors associated with a message.'

  <additionalData> AdditionalData </additionalData> [0..1]
    'Any string of additional data that may help the message processor, for example in a rejection message this might contain a code value or the text of the original request (within a CDATA section).'

</consentException>
```

**Diagram****Schema Component Representation**

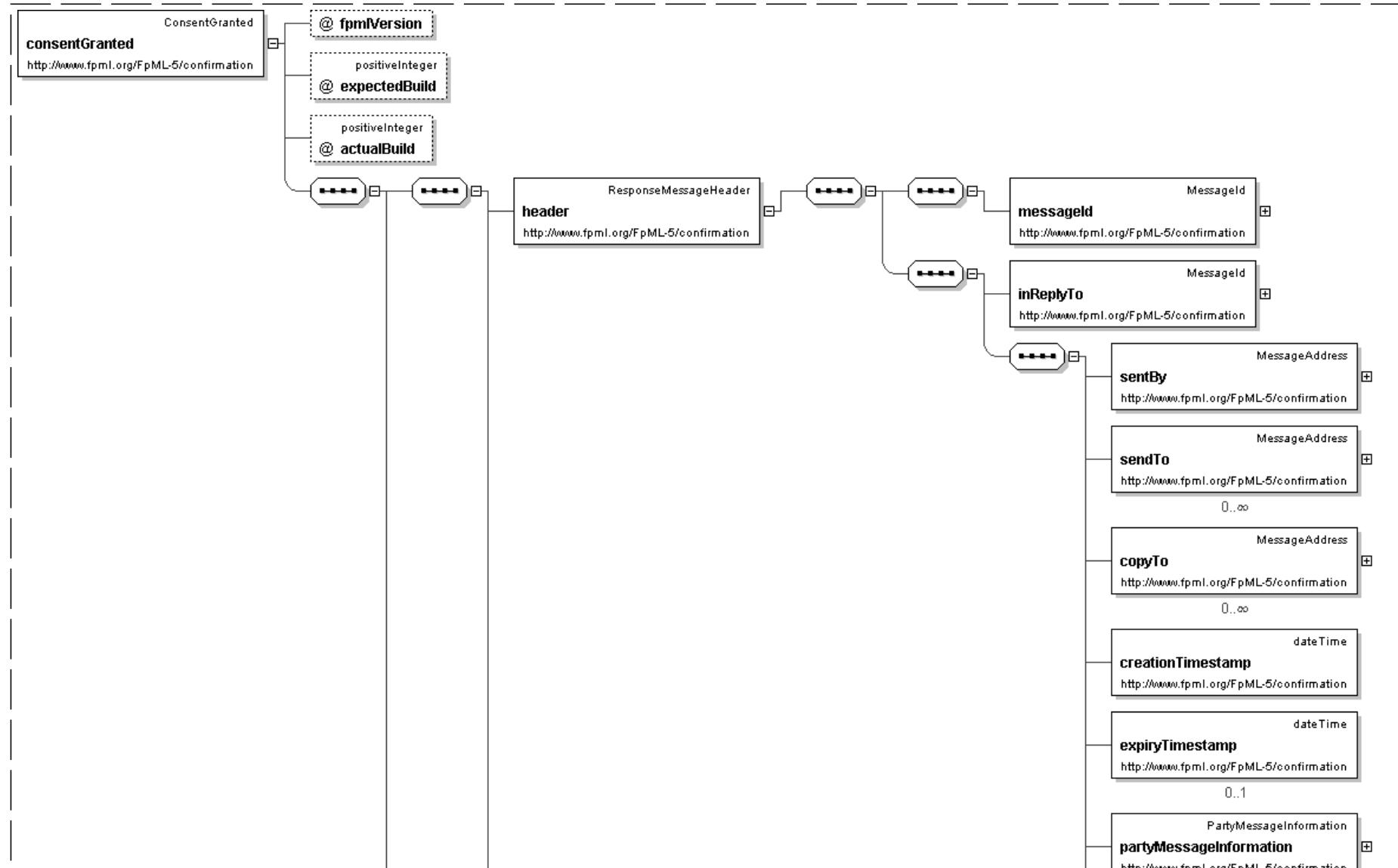
```
<xsd:element name="consentException" type=" Exception " />
```

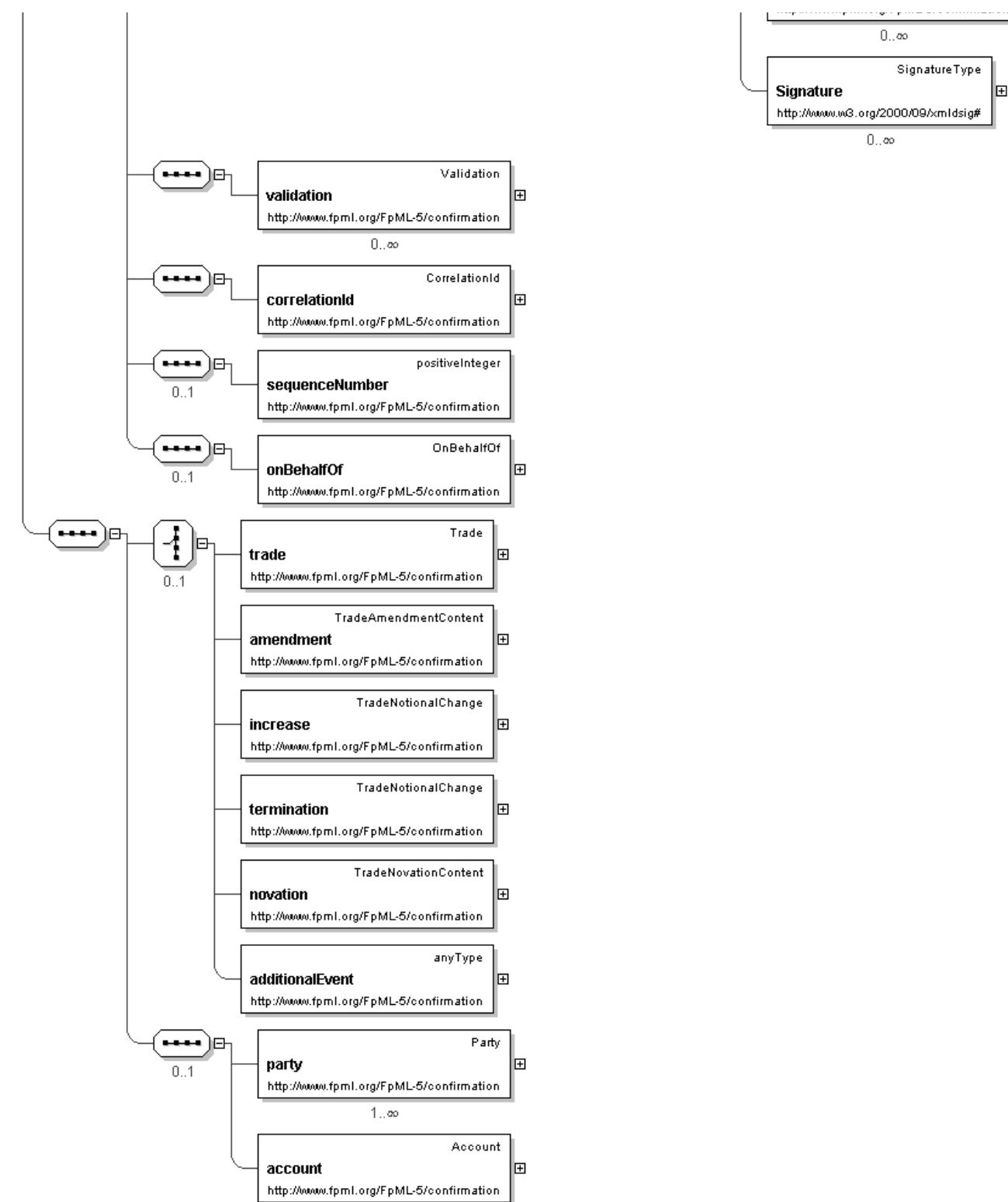
[top](#)

## Element: consentGranted

Name	consentGranted
Type	ConsentGranted
Nillable	no
Abstract	no

### Logical Diagram





0..∞

**XML Instance Representation**

```

<consentGranted
  fpmVersion=" xsd:token (value comes from list: {'5-0'}) [1]
  'Indicate which version of the FpML Schema an FpML message adheres to.'


"
expectedBuild=" xsd:positiveInteger [0..1]
'This optional attribute can be supplied by a message creator in an FpML instance to
specify which build number of the schema was used to define the message when it was generated.'


"
actualBuild="8 [0..1]
'The specific build number of this schema version. This attribute is not included in
an instance document. Instead, it is supplied by the XML parser when the document is
validated against the FpML schema and indicates the build number of the schema file. Every
time FpML publishes a change to the schema, validation rules, or examples within a version
(e.g., version 4.2) the actual build number is incremented. If no changes have been
made between releases within a version (i.e. from Trial Recommendation to Recommendation)
the actual build number stays the same.'


">
<header> ResponseMessageHeader </header> [1]
<validation> Validation </validation> [0..*]
<correlationId> CorrelationId </correlationId> [1]
'A qualified identifier used to correlate between messages'


Start Group: Sequence.model [0..1]
<sequenceNumber> xsd:positiveInteger </sequenceNumber> [1]
'A numeric value that can be used to order messages with the same correlation identifier
from the same sender.'

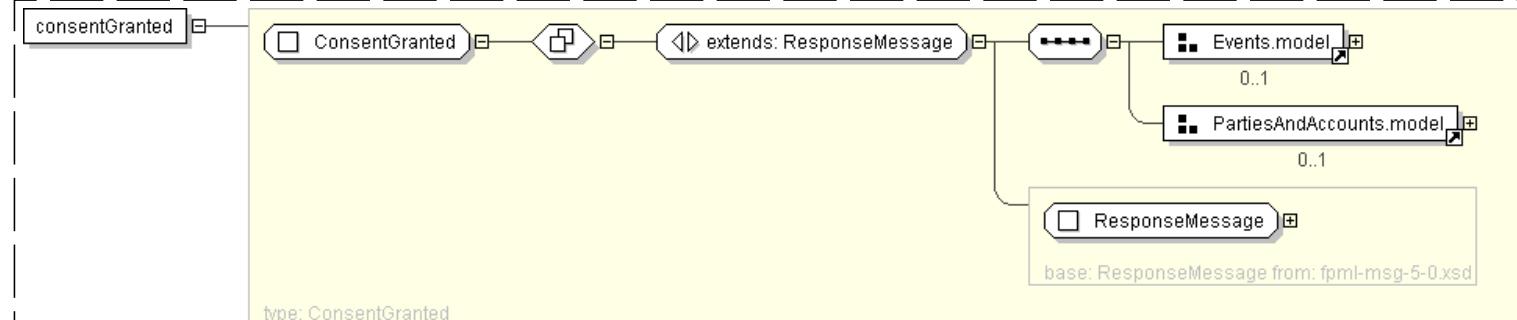

End Group: Sequence.model
Start Group: OnBehalfOf.model [0..1]
<onBehalfOf> OnBehalfOf </onBehalfOf> [1]
'Indicates which party (and accounts) a trade is being processed for.'


End Group: OnBehalfOf.model
Start Group: Events.model [0..1]
Start Choice [1]
  <trade> Trade </trade> [1]
  <amendment> TradeAmendmentContent </amendment> [1]
  <increase> TradeNotionalChange </increase> [1]
  <termination> TradeNotionalChange </termination> [1]
  <novation> TradeNovationContent </novation> [1]
  <additionalEvent> ... </additionalEvent> [1]
End Choice
End Group: Events.model
Start Group: PartiesAndAccounts.model [0..1]
<party> Party </party> [1..*]
'A legal entity or a subdivision of a legal entity.', 'Parties can perform multiple roles in
a trade lifecycle. For example, the principal parties obligated to make payments from time
to time during the term of the trade, but may include other parties involved in, or
incidental to, the trade, such as parties acting in the role of novation transferor/
transferee, broker, calculation agent, etc. In FpML roles are defined in multiple places
within a document.'

```

```
<account> Account </account> [0..*]
'Optional account information used to precisely define the origination and destination
of financial instruments.'
```

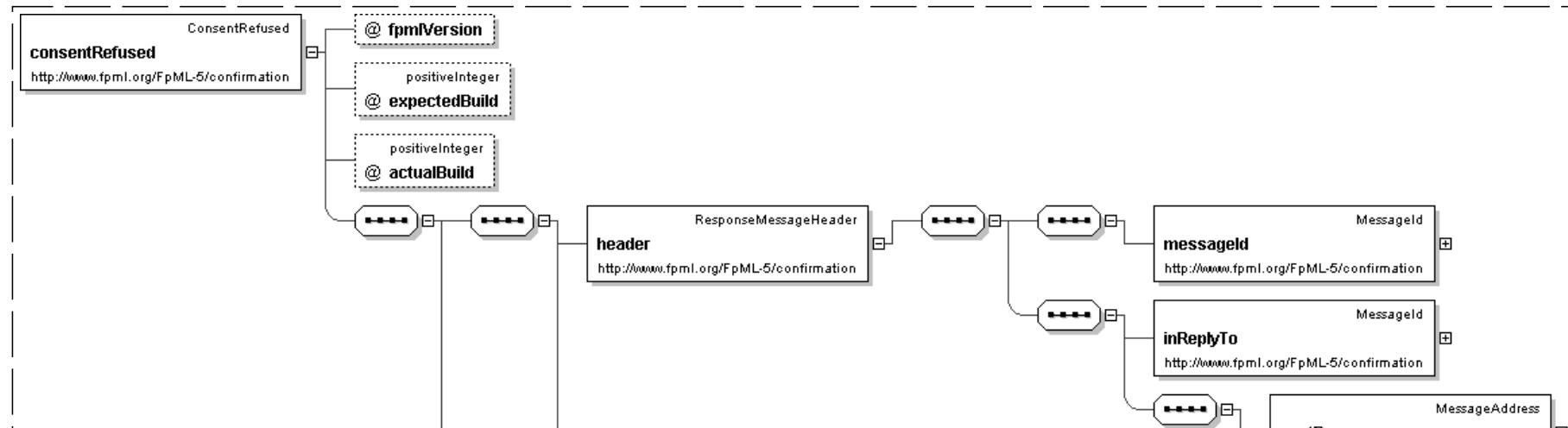
End Group: [PartiesAndAccounts.model](#)  
 </consentGranted>

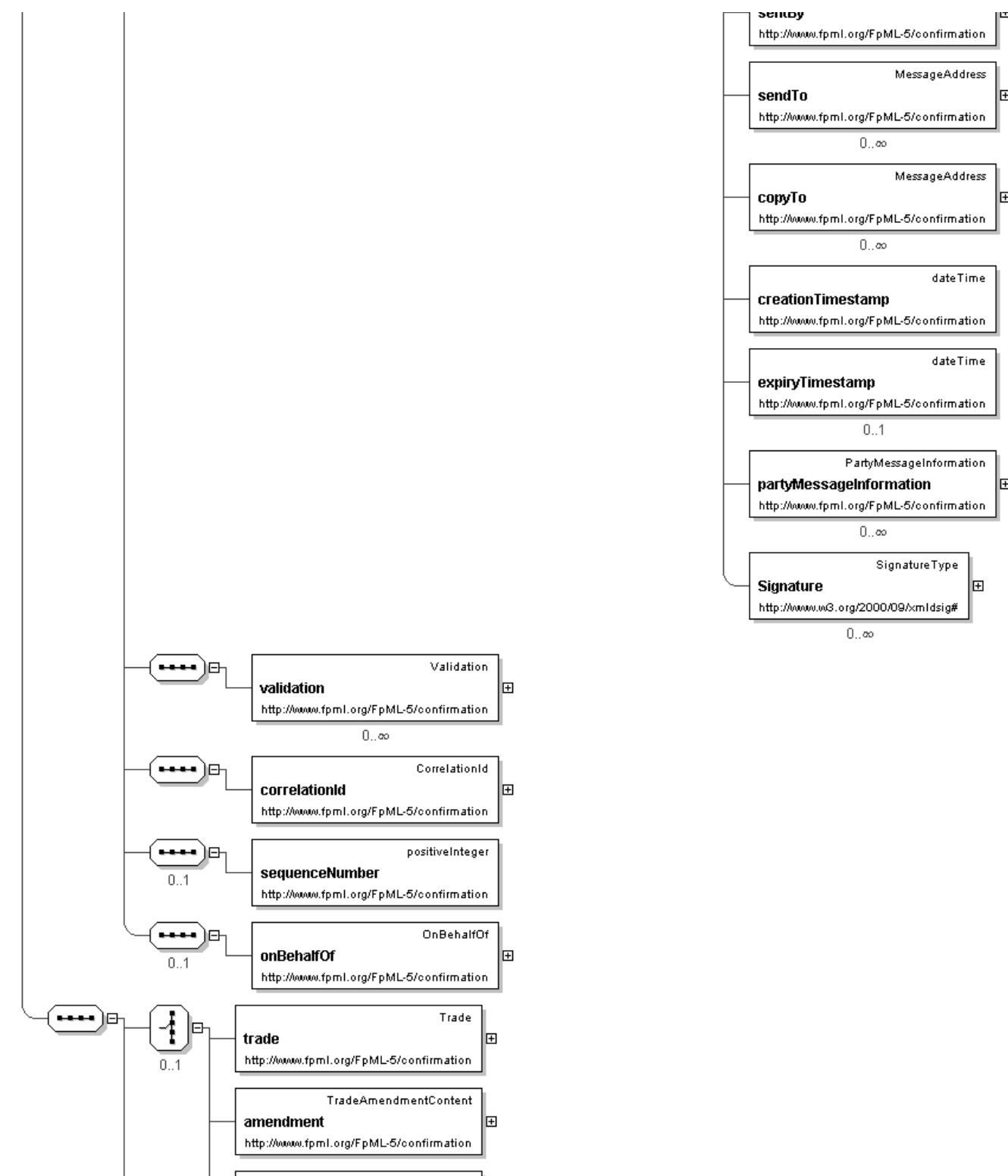
**Diagram****Schema Component Representation**

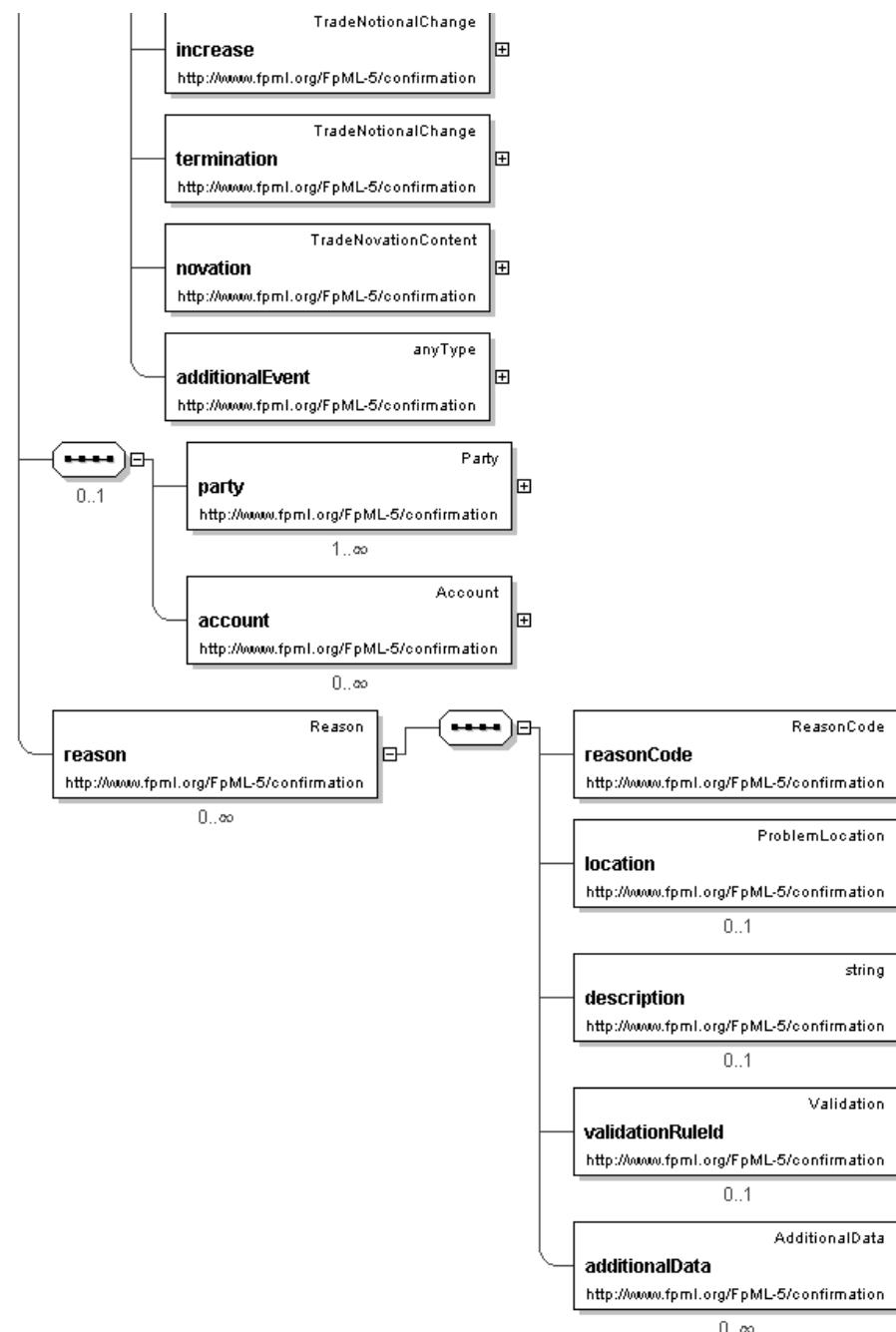
```
<xsd:element name="consentGranted" type=" ConsentGranted " />
```

[top](#)**Element: consentRefused**

Name	consentRefused
Type	ConsentRefused
Nillable	no
Abstract	no

**Logical Diagram**



**XML Instance Representation**

```

<consentRefused
fpmlVersion=" xsd:token (value comes from list: {'5-0'}) [1]
'Indicate which version of the FpML Schema an FpML message adheres to.'
  
```

```

"
expectedBuild= xsd:positiveInteger [0..1]
'This optional attribute can be supplied by a message creator in an FpML instance to
specify which build number of the schema was used to define the message when it was generated.'

"
actualBuild="8 [0..1]
'The specific build number of this schema version. This attribute is not included in
an instance document. Instead, it is supplied by the XML parser when the document is
validated against the FpML schema and indicates the build number of the schema file. Every
time FpML publishes a change to the schema, validation rules, or examples within a version
(e.g., version 4.2) the actual build number is incremented. If no changes have been
made between releases within a version (i.e. from Trial Recommendation to Recommendation)
the actual build number stays the same.'


">
<header> ResponseMessageHeader </header> [1]
<validation> Validation </validation> [0..*]
<correlationId> CorrelationId </correlationId> [1]
'A qualified identifier used to correlate between messages'

Start Group: Sequence.model [0..1]
<sequenceNumber> xsd:positiveInteger </sequenceNumber> [1]
'A numeric value that can be used to order messages with the same correlation identifier
from the same sender.'

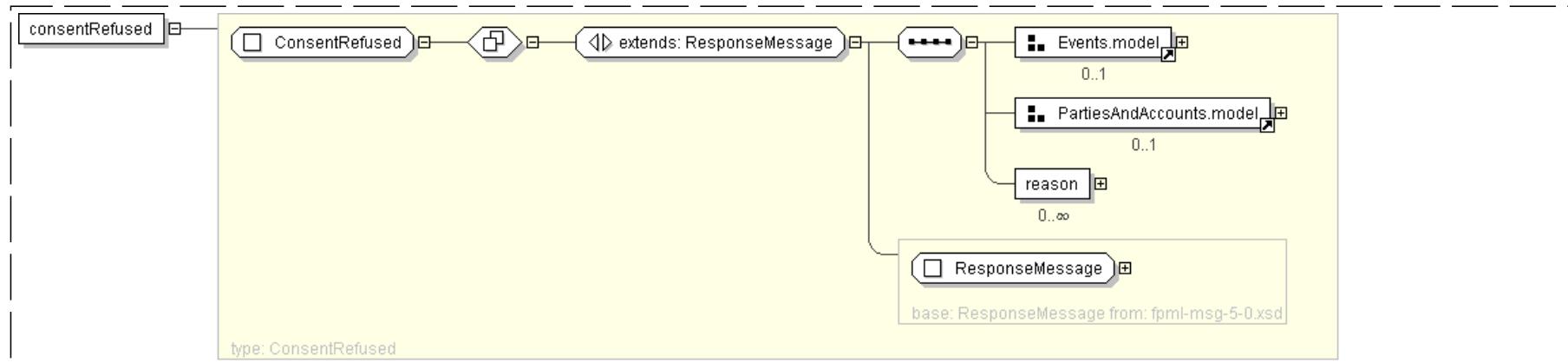

End Group: Sequence.model
Start Group: OnBehalfOf.model [0..1]
<onBehalfOf> OnBehalfOf </onBehalfOf> [1]
'Indicates which party (and accounts) a trade is being processed for.'


End Group: OnBehalfOf.model
Start Group: Events.model [0..1]
Start Choice [1]
<trade> Trade </trade> [1]
<amendment> TradeAmendmentContent </amendment> [1]
<increase> TradeNotionalChange </increase> [1]
<termination> TradeNotionalChange </termination> [1]
<novation> TradeNovationContent </novation> [1]
<additionalEvent> ... </additionalEvent> [1]
End Choice
End Group: Events.model
Start Group: PartiesAndAccounts.model [0..1]
<party> Party </party> [1..*]
'A legal entity or a subdivision of a legal entity.', 'Parties can perform multiple roles in
a trade lifecycle. For example, the principal parties obligated to make payments from time
to time during the term of the trade, but may include other parties involved in, or
incidental to, the trade, such as parties acting in the role of novation transferor/
transferee, broker, calculation agent, etc. In FpML roles are defined in multiple places
within a document.'


<account> Account </account> [0..*]
'Optional account information used to precisely define the origination and destination
of financial instruments.'


End Group: PartiesAndAccounts.model
<reason> Reason </reason> [0..*]
</consentRefused>

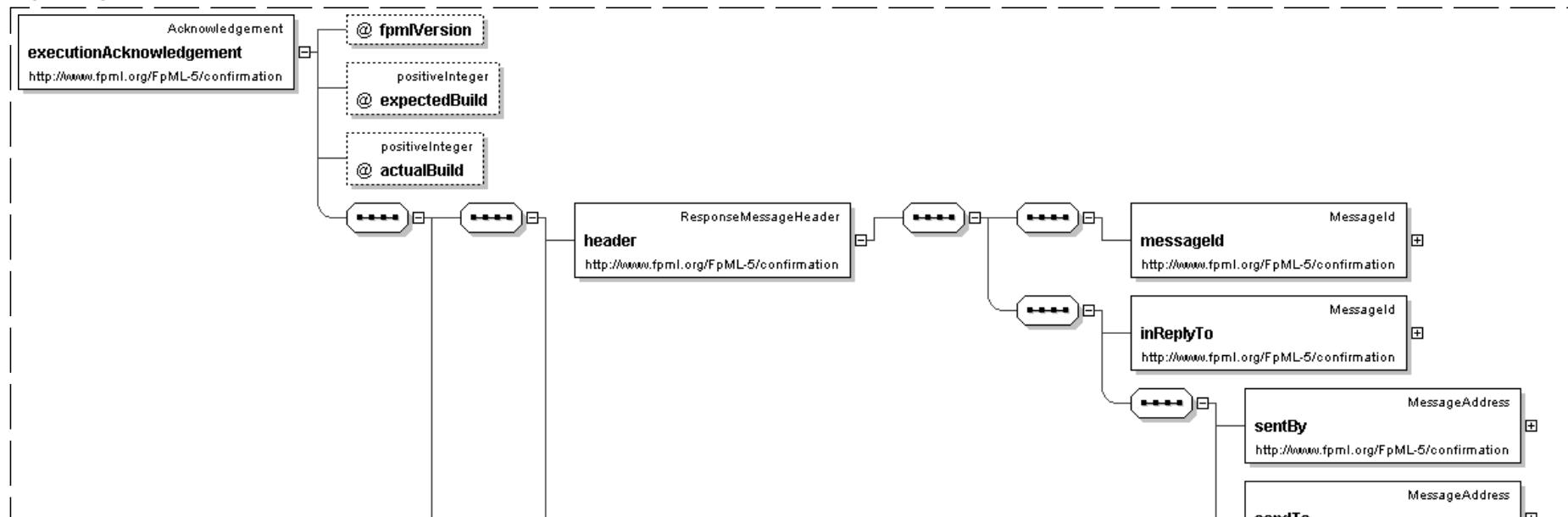
```

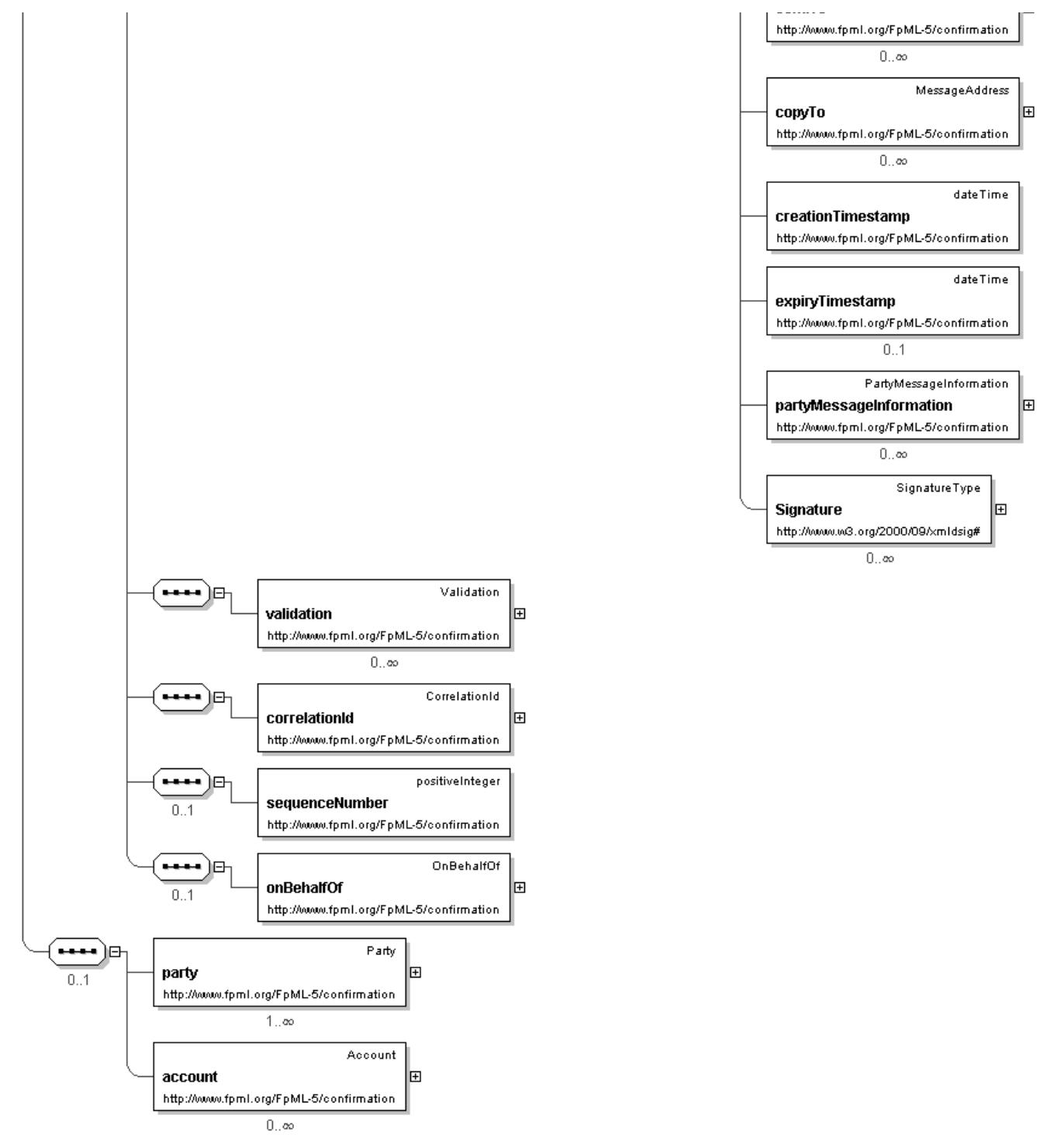
**Diagram****Schema Component Representation**

```
<xsd:element name="consentRefused" type=" ConsentRefused " />
```

[top](#)**Element: executionAcknowledgement**

<b>Name</b>	executionAcknowledgement
<b>Type</b>	<a href="#">Acknowledgement</a>
<b>Nillable</b>	no
<b>Abstract</b>	no

**Logical Diagram**

**XML Instance Representation**

```

<executionAcknowledgement
  fpmVersion=" xsd:token (value comes from list: {'5-0'}) [1]
  'Indicate which version of the FpML Schema an FpML message adheres to.'


"
expectedBuild=" xsd:positiveInteger [0..1]
'This optional attribute can be supplied by a message creator in an FpML instance to
specify which build number of the schema was used to define the message when it was generated.'


"
actualBuild="8 [0..1]
'The specific build number of this schema version. This attribute is not included in
an instance document. Instead, it is supplied by the XML parser when the document is
validated against the FpML schema and indicates the build number of the schema file. Every
time FpML publishes a change to the schema, validation rules, or examples within a version
(e.g., version 4.2) the actual build number is incremented. If no changes have been
made between releases within a version (i.e. from Trial Recommendation to Recommendation)
the actual build number stays the same.'


">
<header> ResponseMessageHeader </header> [1]
<validation> Validation </validation> [0..*]
<correlationId> CorrelationId </correlationId> [1]
'A qualified identifier used to correlate between messages'


Start Group: Sequence.model [0..1]
<sequenceNumber> xsd:positiveInteger </sequenceNumber> [1]
'A numeric value that can be used to order messages with the same correlation identifier
from the same sender.'


End Group: Sequence.model
Start Group: OnBehalfOf.model [0..1]
<onBehalfOf> OnBehalfOf </onBehalfOf> [1]
'Indicates which party (and accounts) a trade is being processed for.'

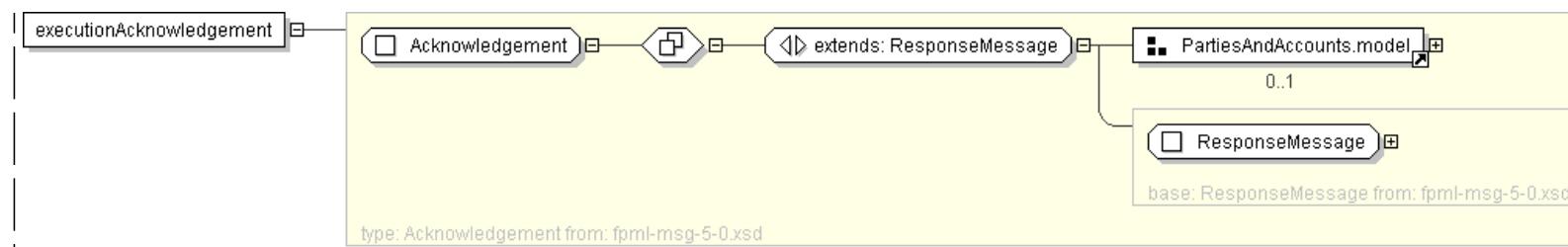

End Group: OnBehalfOf.model
Start Group: PartiesAndAccounts.model [0..1]
<party> Party </party> [1..*]
'A legal entity or a subdivision of a legal entity.', 'Parties can perform multiple roles in
a trade lifecycle. For example, the principal parties obligated to make payments from time
to time during the term of the trade, but may include other parties involved in, or
incidental to, the trade, such as parties acting in the role of novation transferor/
transferee, broker, calculation agent, etc. In FpML roles are defined in multiple places
within a document.'


<account> Account </account> [0..*]
'Optional account information used to precisely define the origination and destination
of financial instruments.'


End Group: PartiesAndAccounts.model
</executionAcknowledgement>

```

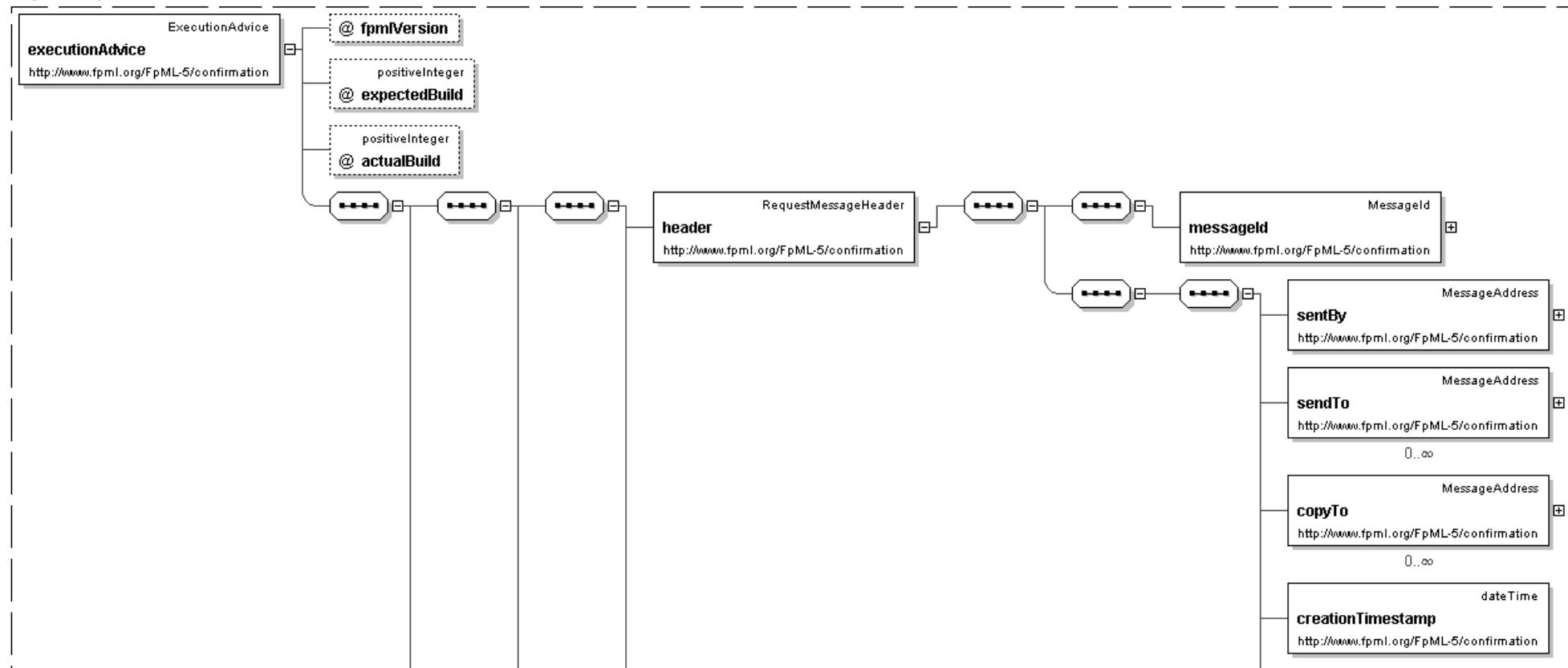
**Diagram**

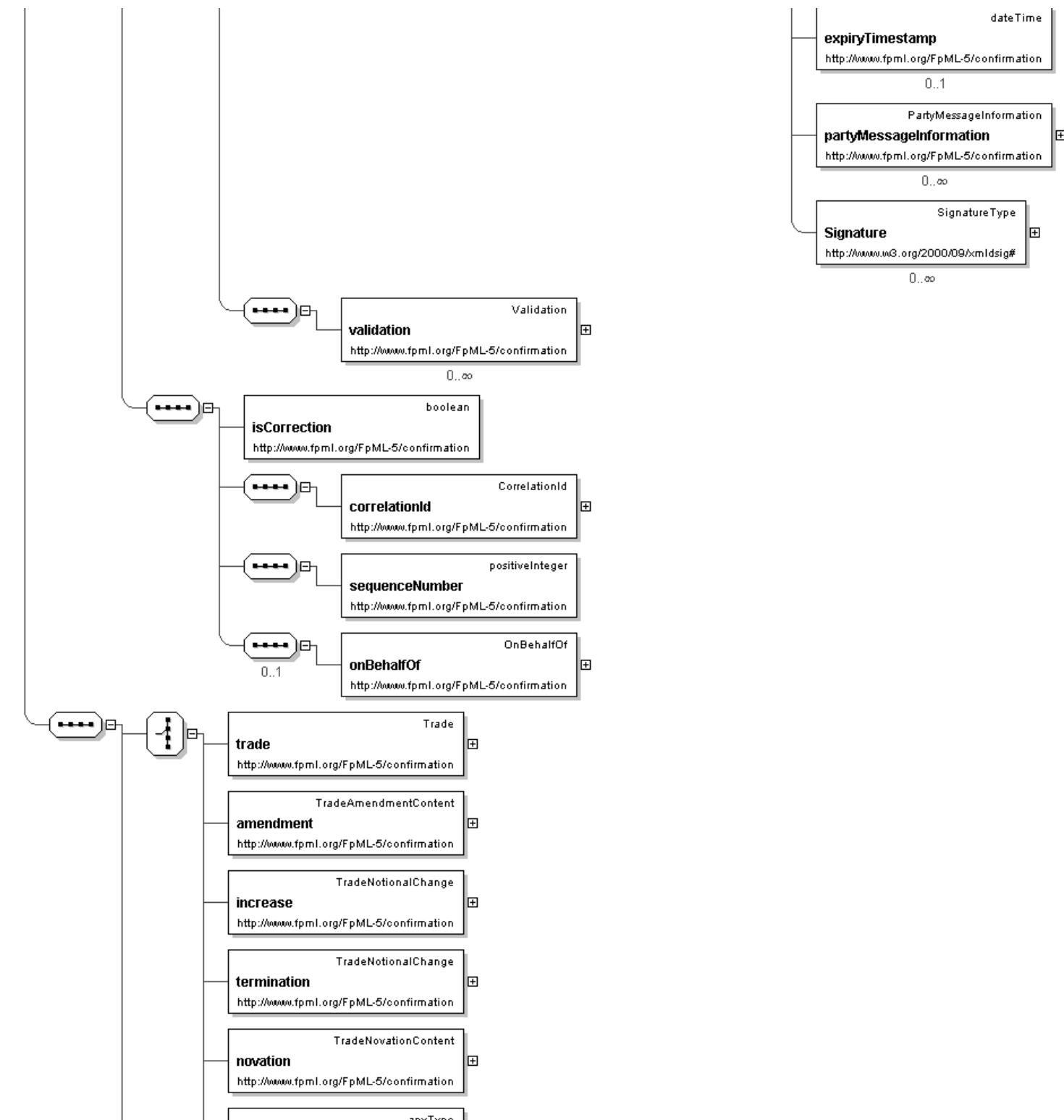
**Schema Component Representation**

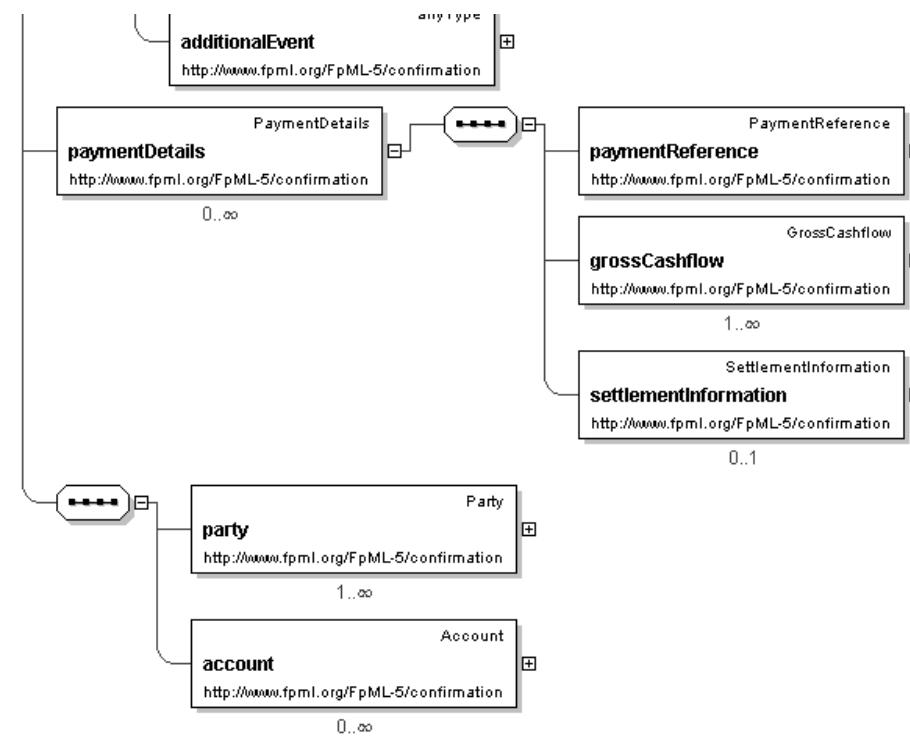
```
<xsd:element name="executionAcknowledgement" type="Acknowledgement" />
```

[top](#)**Element: executionAdvice**

<b>Name</b>	executionAdvice
<b>Type</b>	<a href="#">ExecutionAdvice</a>
<b>Nullable</b>	no
<b>Abstract</b>	no

**Logical Diagram**



**XML Instance Representation**

```

<executionAdvice
  fpmlVersion=" xsd:token (value comes from list: {'5-0'}) [1]
  'Indicate which version of the FpML Schema an FpML message adheres to.'

  "
  expectedBuild=" xsd:positiveInteger [0..1]
  'This optional attribute can be supplied by a message creator in an FpML instance to
  specify which build number of the schema was used to define the message when it was generated.'

  "
  actualBuild="8 [0..1]
  'The specific build number of this schema version. This attribute is not included in
  an instance document. Instead, it is supplied by the XML parser when the document is
  validated against the FpML schema and indicates the build number of the schema file. Every
  time FpML publishes a change to the schema, validation rules, or examples within a version
  (e.g., version 4.2) the actual build number is incremented. If no changes have been
  made between releases within a version (i.e. from Trial Recommendation to Recommendation)
  the actual build number stays the same.'


  ">
  <header> RequestMessageHeader </header> [1]
  <validation> Validation </validation> [0..*]
  <isCorrection> xsd:boolean </isCorrection> [1]
  'Indicates if this message corrects an earlier request.'


  <correlationId> CorrelationId </correlationId> [1]
  'A qualified identifier used to correlate between messages'
  
```

```
<sequenceNumber> xsd:positiveInteger </sequenceNumber> [1]
'A numeric value that can be used to order messages with the same correlation identifier
from the same sender.'
```

Start Group: OnBehalfOf.model [0..1]

```
<onBehalfOf> OnBehalfOf </onBehalfOf> [1]
'Indicates which party (and accounts) a trade is being processed for.'
```

End Group: OnBehalfOf.model

Start Choice [1]

```
<trade> Trade </trade> [1]
<amendment> TradeAmendmentContent </amendment> [1]
<increase> TradeNotionalChange </increase> [1]
<termination> TradeNotionalChange </termination> [1]
<novation> TradeNovationContent </novation> [1]
<additionalEvent> ... </additionalEvent> [1]
```

End Choice

```
<paymentDetails> PaymentDetails </paymentDetails> [0..*]
```

'Details of the payments, like amount breakdowns, settlement information.'

```
<party> Party </party> [1..*]
```

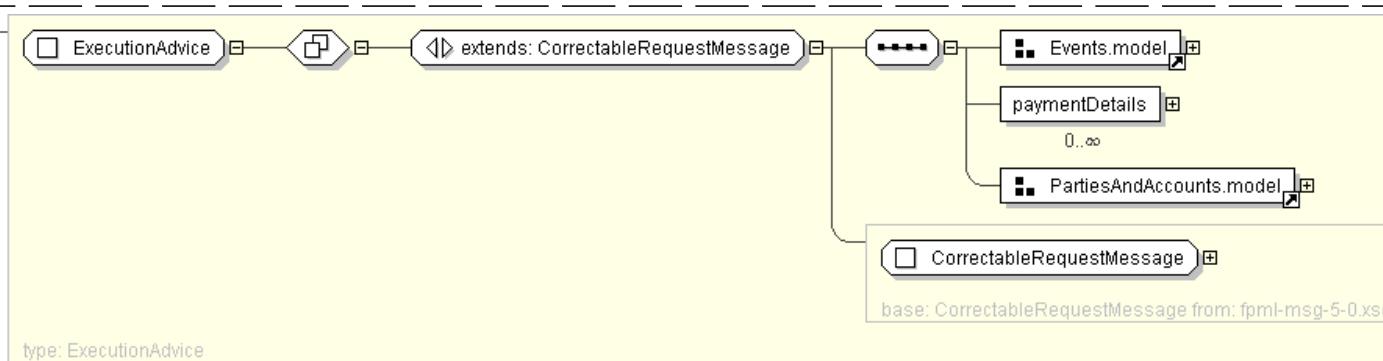
'A legal entity or a subdivision of a legal entity.', 'Parties can perform multiple roles in a trade lifecycle. For example, the principal parties obligated to make payments from time to time during the term of the trade, but may include other parties involved in, or incidental to, the trade, such as parties acting in the role of novation transferor/transferee, broker, calculation agent, etc. In FpML roles are defined in multiple places within a document.'

```
<account> Account </account> [0..*]
```

'Optional account information used to precisely define the origination and destination of financial instruments.'

</executionAdvice>

#### Diagram

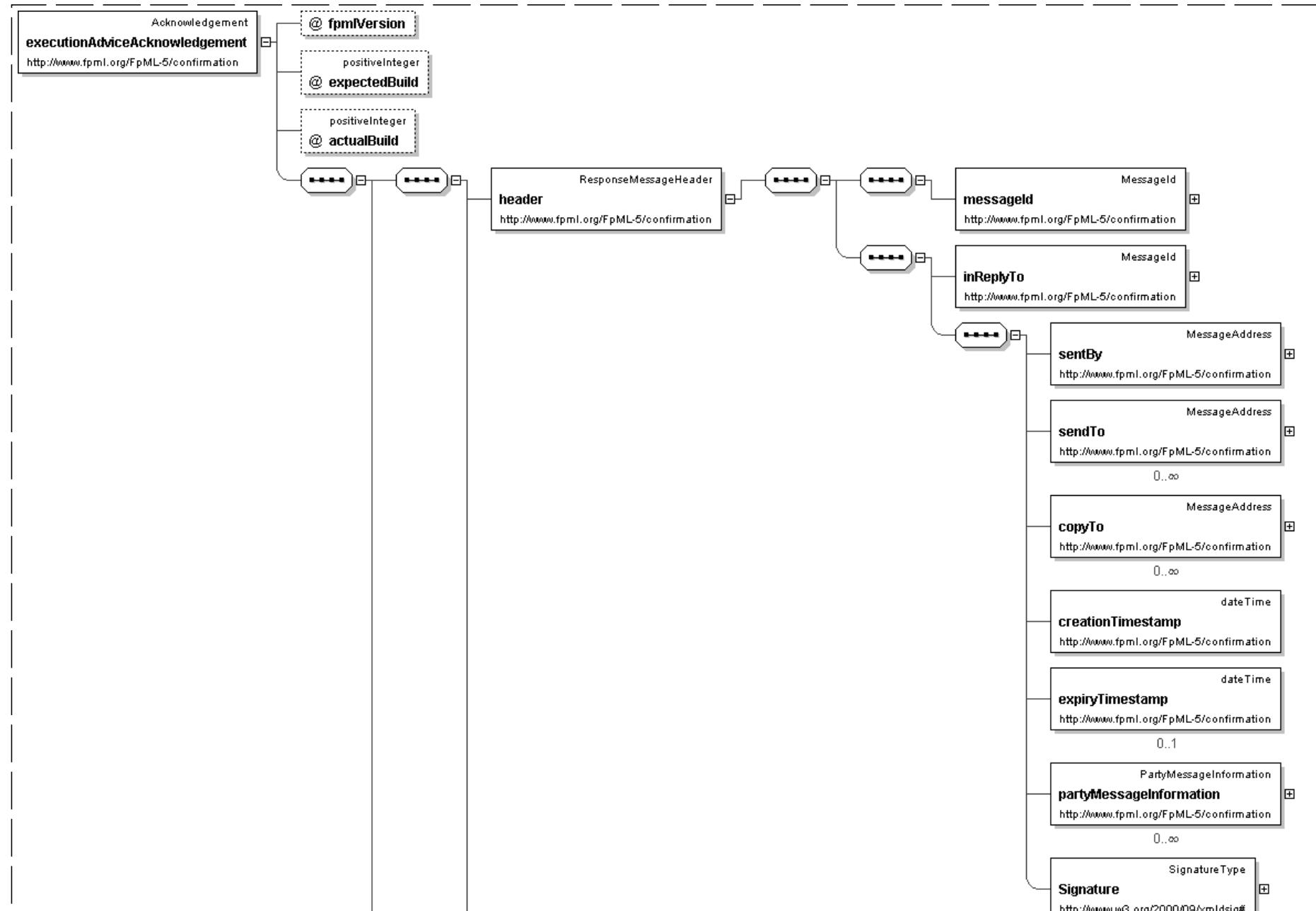


#### Schema Component Representation

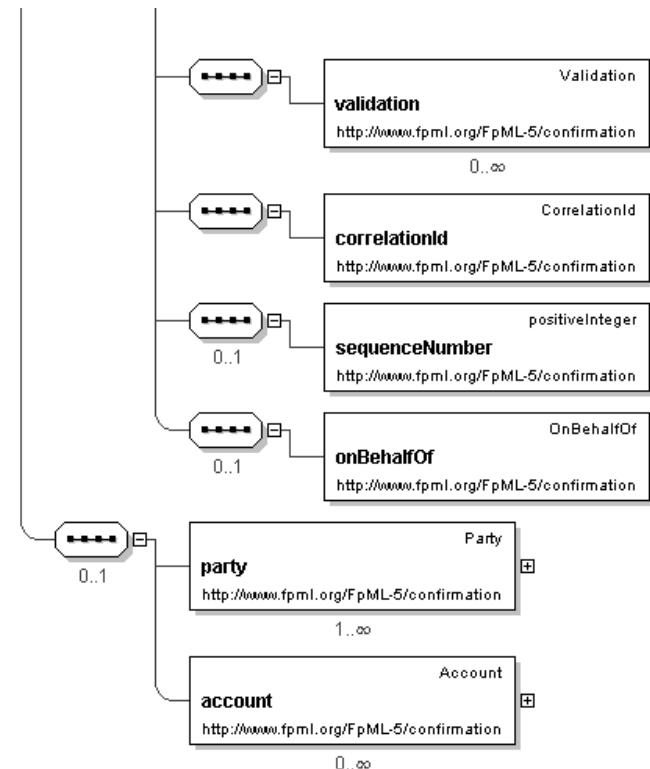
```
<xsd:element name="executionAdvice" type=" ExecutionAdvice " />
```

**Element: executionAdviceAcknowledgement**

<b>Name</b>	executionAdviceAcknowledgement
<b>Type</b>	Acknowledgement
<b>Nillable</b>	no
<b>Abstract</b>	no

**Logical Diagram**

0..∞

**XML Instance Representation**

```

<executionAdviceAcknowledgement
fpmlVersion=" xsd:token (value comes from list: {'5-0'}) [1]
'Indicate which version of the FpML Schema an FpML message adheres to.'

"
expectedBuild=" xsd:positiveInteger [0..1]
'This optional attribute can be supplied by a message creator in an FpML instance to
specify which build number of the schema was used to define the message when it was generated.'

"
actualBuild="8 [0..1]
'The specific build number of this schema version. This attribute is not included in
an instance document. Instead, it is supplied by the XML parser when the document is
validated against the FpML schema and indicates the build number of the schema file. Every
time FpML publishes a change to the schema, validation rules, or examples within a version
(e.g., version 4.2) the actual build number is incremented. If no changes have been
made between releases within a version (i.e. from Trial Recommendation to Recommendation)
the actual build number stays the same.'

">
<header> ResponseMessageHeader </header> [1]
<validation> Validation </validation> [0..*]
<correlationId> CorrelationId </correlationId> [1]
'A qualified identifier used to correlate between messages'

Start Group: Sequence.model [0..1]

```

```
<sequenceNumber> xsd:positiveInteger </sequenceNumber> [1]
'A numeric value that can be used to order messages with the same correlation identifier
from the same sender.'
```

End Group: Sequence.modelStart Group: OnBehalfOf.model [0..1]

```
<onBehalfOf> OnBehalfOf </onBehalfOf> [1]
'Indicates which party (and accounts) a trade is being processed for.'
```

End Group: OnBehalfOf.modelStart Group: PartiesAndAccounts.model [0..1]

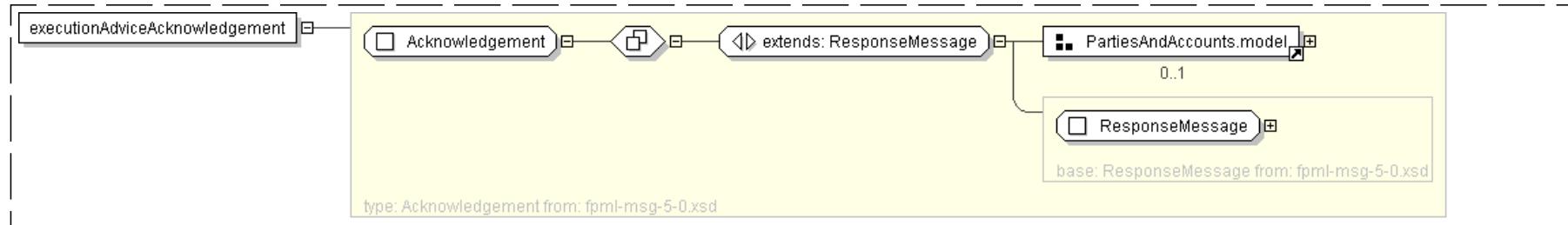
```
<party> Party </party> [1..*]
'A legal entity or a subdivision of a legal entity.', 'Parties can perform multiple roles in
a trade lifecycle. For example, the principal parties obligated to make payments from time
to time during the term of the trade, but may include other parties involved in, or
incidental to, the trade, such as parties acting in the role of novation transferor/
transferee, broker, calculation agent, etc. In FpML roles are defined in multiple places
within a document.'
```

```
<account> Account </account> [0..*]
```

```
'Optional account information used to precisely define the origination and destination
of financial instruments.'
```

End Group: PartiesAndAccounts.model

&lt;/executionAdviceAcknowledgement&gt;

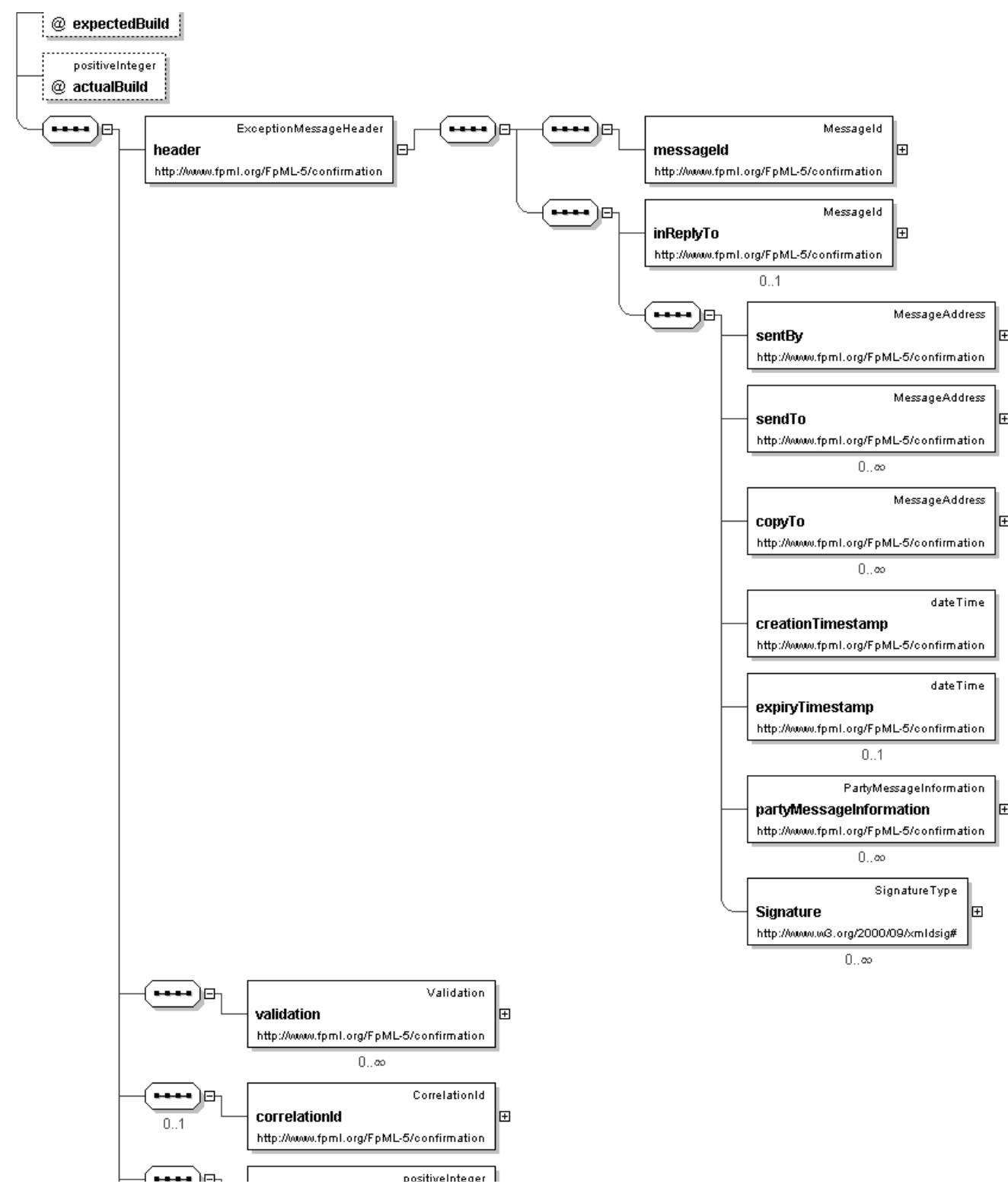
**Diagram****Schema Component Representation**

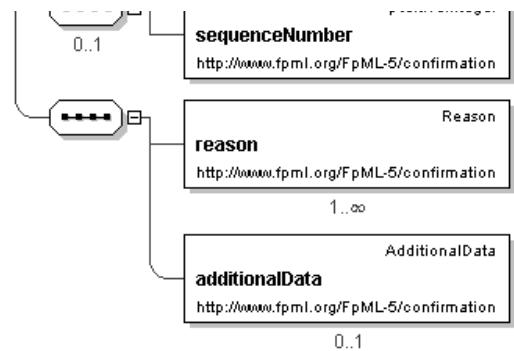
```
<xsd:element name="executionAdviceAcknowledgement" type=" Acknowledgement " />
```

[top](#)**Element: executionAdviceException**

<b>Name</b>	executionAdviceException
<b>Type</b>	<a href="#">Exception</a>
<b>Nillable</b>	no
<b>Abstract</b>	no

**Logical Diagram**



**XML Instance Representation**

```

<executionAdviceException
  fpmlVersion=" xsd:token (value comes from list: {'5-0'}) [1]
  'Indicate which version of the FpML Schema an FpML message adheres to.'

  "
  expectedBuild=" xsd:positiveInteger [0..1]
  'This optional attribute can be supplied by a message creator in an FpML instance to
  specify which build number of the schema was used to define the message when it was generated.'

  "
  actualBuild="8 [0..1]
  'The specific build number of this schema version. This attribute is not included in
  an instance document. Instead, it is supplied by the XML parser when the document is
  validated against the FpML schema and indicates the build number of the schema file. Every
  time FpML publishes a change to the schema, validation rules, or examples within a version
  (e.g., version 4.2) the actual build number is incremented. If no changes have been
  made between releases within a version (i.e. from Trial Recommendation to Recommendation)
  the actual build number stays the same.'

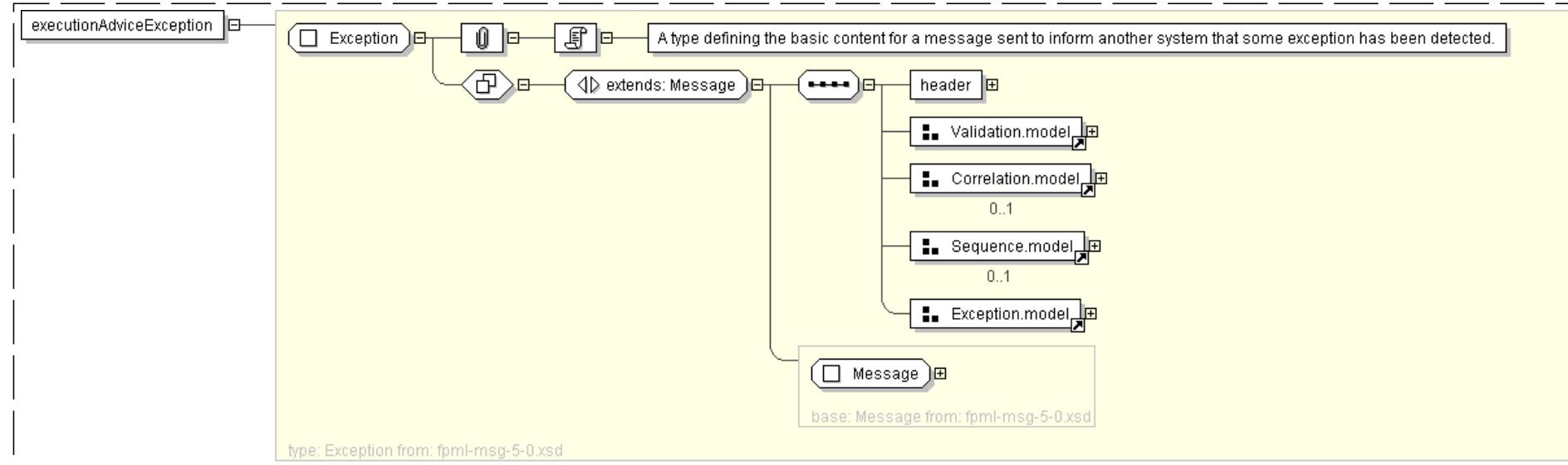

  ">
  <header> ExceptionMessageHeader </header> [1]
  <validation> Validation </validation> [0..*]
  Start Group: Correlation.model [0..1]
    <correlationId> CorrelationId </correlationId> [1]
    'A qualified identifier used to correlate between messages'

  End Group: Correlation.model
  Start Group: Sequence.model [0..1]
    <sequenceNumber> xsd:positiveInteger </sequenceNumber> [1]
    'A numeric value that can be used to order messages with the same correlation identifier
    from the same sender.'

  End Group: Sequence.model
  <reason> Reason </reason> [1..*]
  'An instance of the Reason type used to record the nature of any errors associated with
  a message.'

  <additionalData> AdditionalData </additionalData> [0..1]
  'Any string of additional data that may help the message processor, for example in a
  rejection message this might contain a code value or the text of the original request (within
  a CDATA section).'
  
```

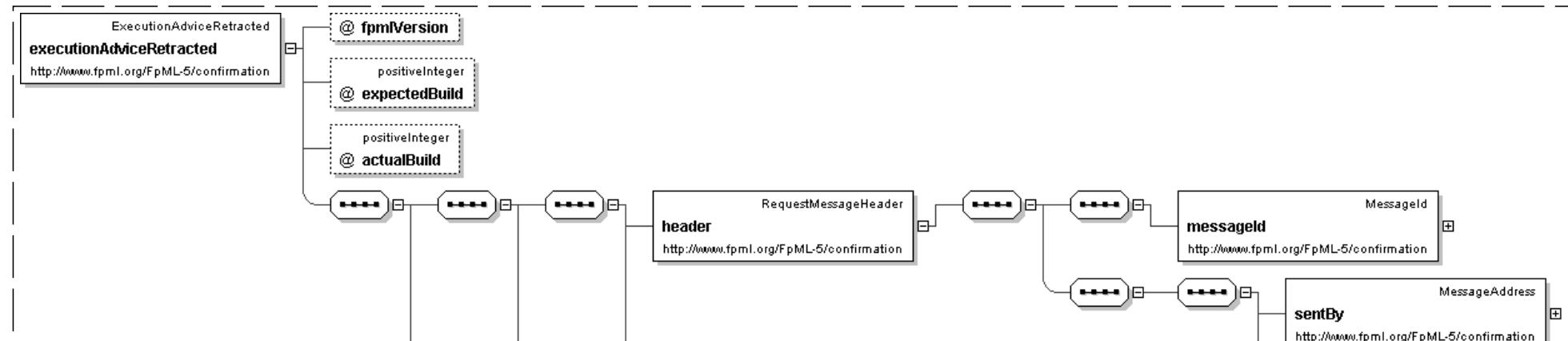
```
</executionAdviceException>
```

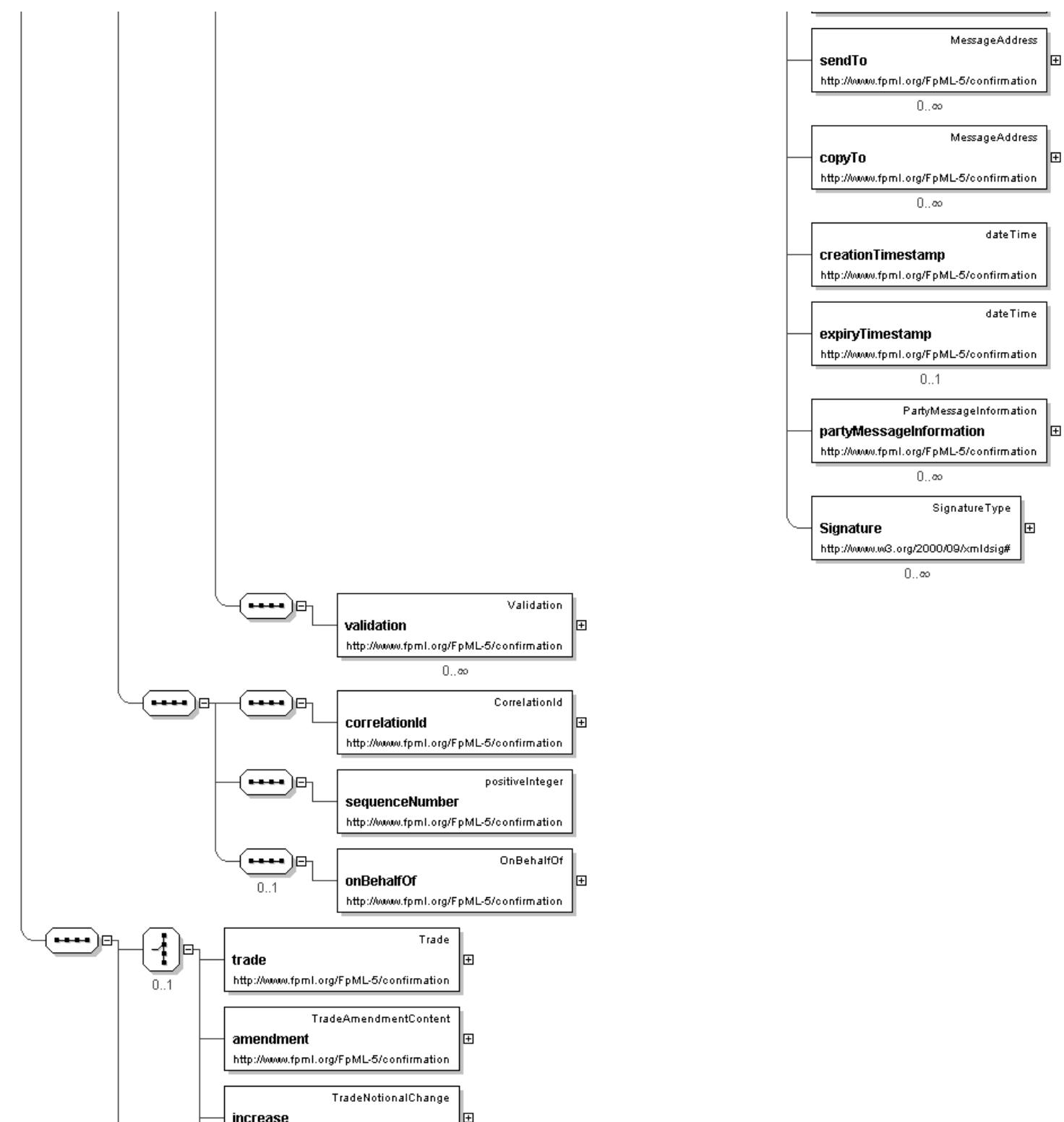
**Diagram****Schema Component Representation**

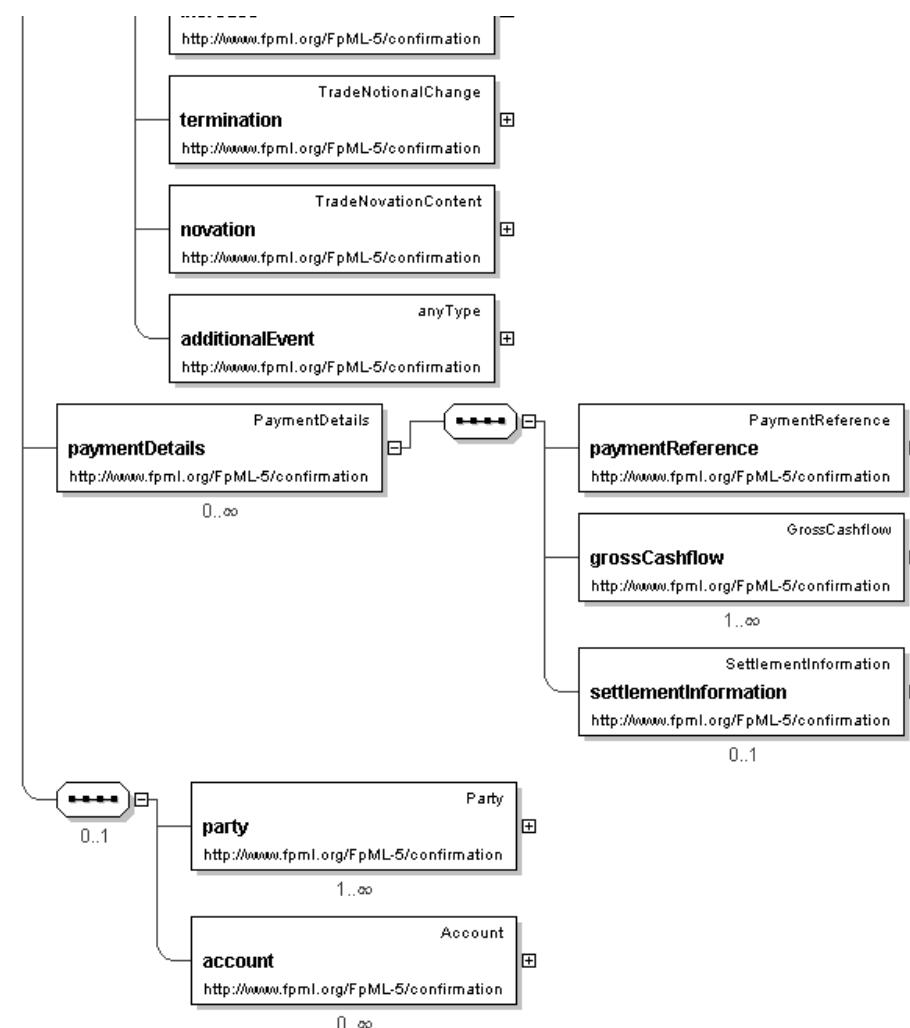
```
<xsd:element name="executionAdviceException" type=" Exception " />
```

[top](#)**Element: executionAdviceRetracted**

Name	executionAdviceRetracted
Type	ExecutionAdviceRetracted
Nillable	no
Abstract	no

**Logical Diagram**





#### XML Instance Representation

```

<executionAdviceRetracted
fpmlVersion=" xsd:token (value comes from list: {'5-0'}) [1]
'Indicate which version of the FpML Schema an FpML message adheres to.'

"
expectedBuild=" xsd:positiveInteger [0..1]
'This optional attribute can be supplied by a message creator in an FpML instance to
specify which build number of the schema was used to define the message when it was generated.'

"
actualBuild="8 [0..1]
'The specific build number of this schema version. This attribute is not included in
an instance document. Instead, it is supplied by the XML parser when the document is
validated against the FpML schema and indicates the build number of the schema file. Every
time FpML publishes a change to the schema, validation rules, or examples within a version
(e.g., version 4.2) the actual build number is incremented. If no changes have been
'
  
```

This section provides the XML instance representation for the FpML Confirmation Processes. It includes attributes like executionAdviceRetracted, fpmlVersion, expectedBuild, and actualBuild, along with their descriptions and notes.

*'made between releases within a version (i.e. from Trial Recommendation to Recommendation)  
the actual build number stays the same.'*

">

<header> RequestMessageHeader </header> [1]  
<validation> Validation </validation> [0..\*]  
<correlationId> CorrelationId </correlationId> [1]

*'A qualified identifier used to correlate between messages'*

<sequenceNumber> xsd:positiveInteger </sequenceNumber> [1]

*'A numeric value that can be used to order messages with the same correlation identifier  
from the same sender.'*

Start Group: OnBehalfOf.model [0..1]  
<onBehalfOf> OnBehalfOf </onBehalfOf> [1]

*'Indicates which party (and accounts) a trade is being processed for.'*

End Group: OnBehalfOf.model

Start Group: Events.model [0..1]  
Start Choice [1]

<trade> Trade </trade> [1]  
<amendment> TradeAmendmentContent </amendment> [1]  
<increase> TradeNotionalChange </increase> [1]  
<termination> TradeNotionalChange </termination> [1]  
<novation> TradeNovationContent </novation> [1]  
<additionalEvent> ... </additionalEvent> [1]

End Choice

End Group: Events.model

<paymentDetails> PaymentDetails </paymentDetails> [0..\*]

*'Details of the payments, like amount breakdowns, settlement information.'*

Start Group: PartiesAndAccounts.model [0..1]  
<party> Party </party> [1..\*]

*'A legal entity or a subdivision of a legal entity.', 'Parties can perform multiple roles in  
a trade lifecycle. For example, the principal parties obligated to make payments from time  
to time during the term of the trade, but may include other parties involved in, or  
incidental to, the trade, such as parties acting in the role of novation transferor/  
transferee, broker, calculation agent, etc. In FpML roles are defined in multiple places  
within a document.'*

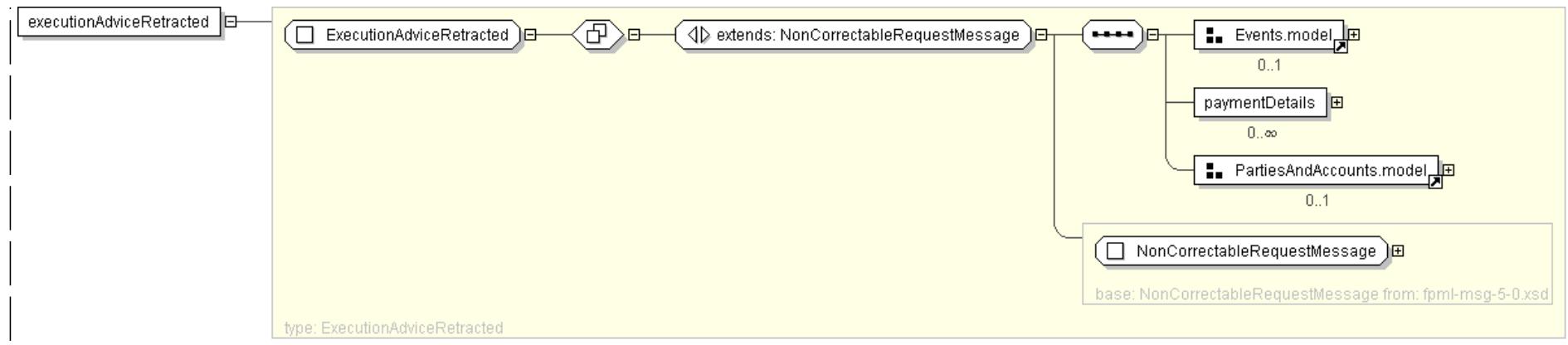
<account> Account </account> [0..\*]

*'Optional account information used to precisely define the origination and destination  
of financial instruments.'*

End Group: PartiesAndAccounts.model

</executionAdviceRetracted>

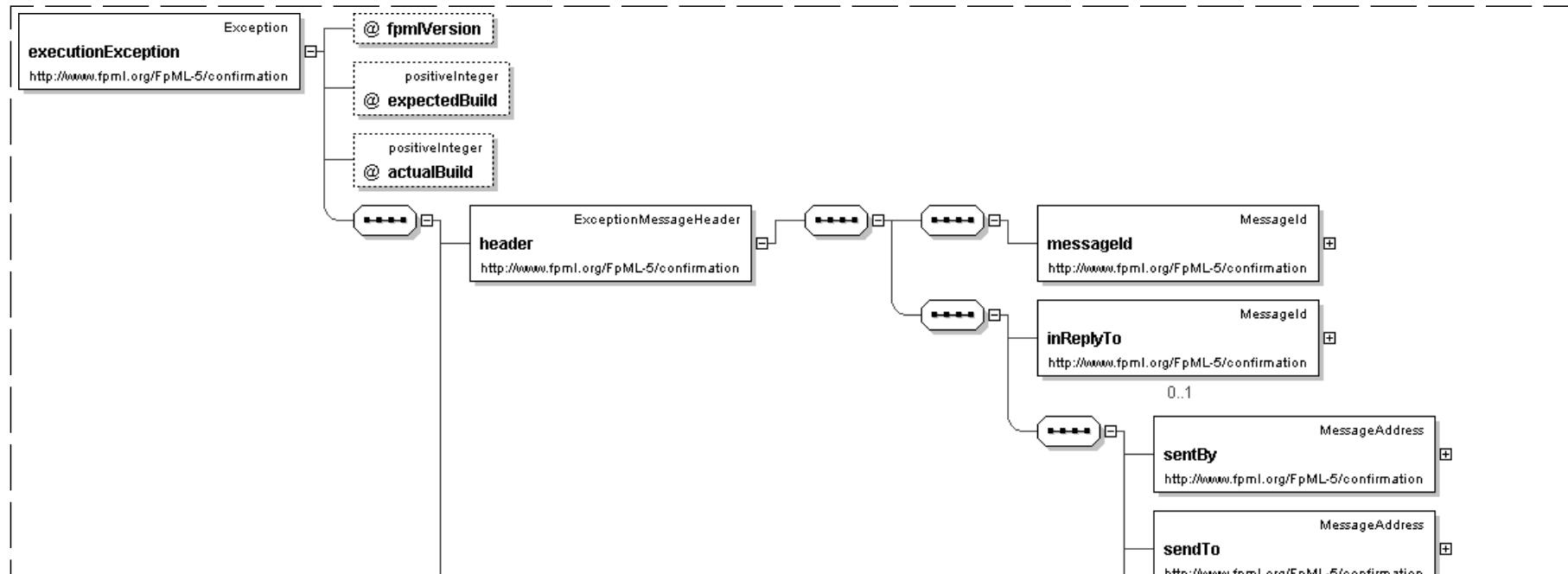
**Diagram**

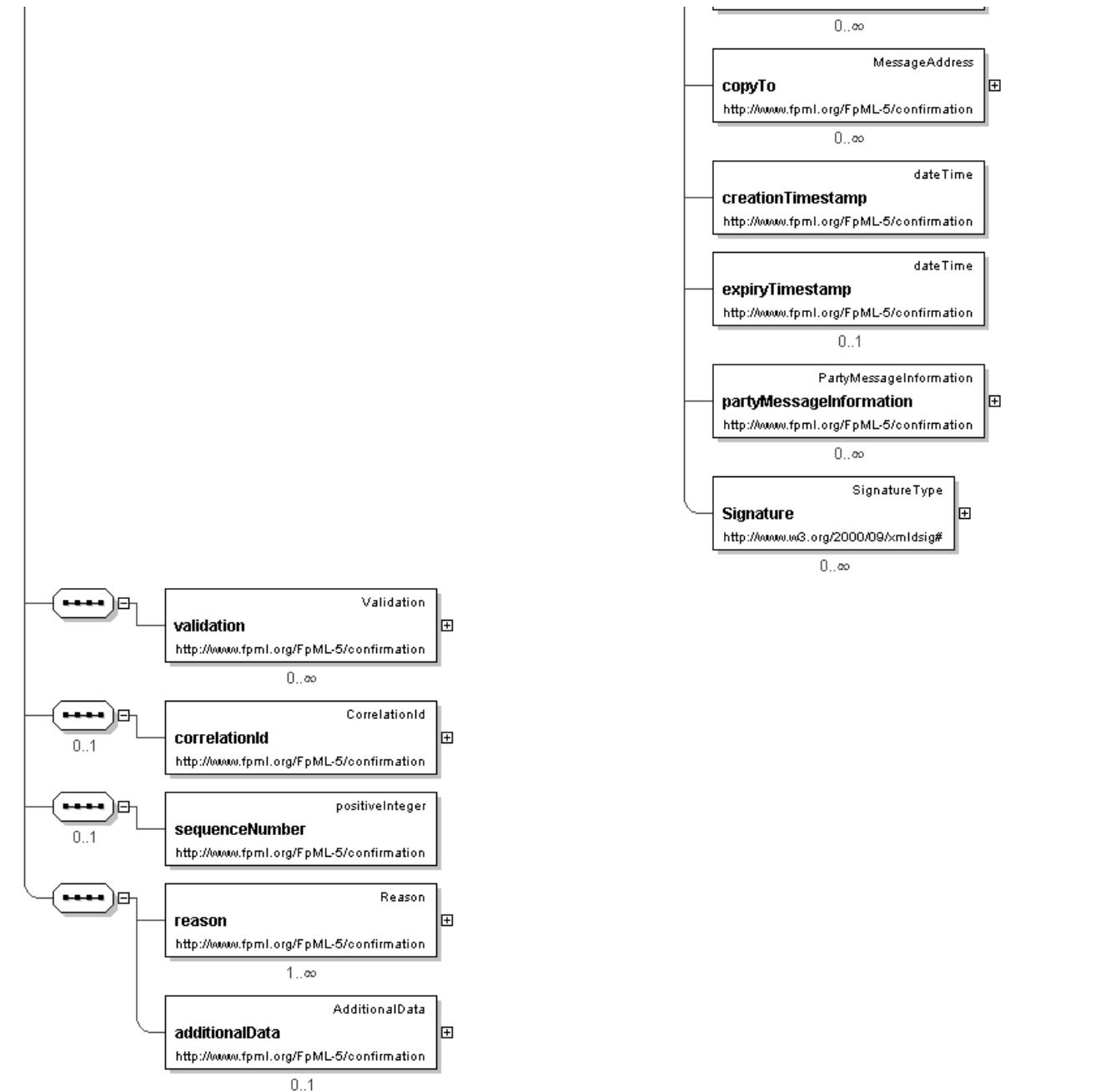
**Schema Component Representation**

```
<xsd:element name="executionAdviceRetracted" type="ExecutionAdviceRetracted" />
```

[top](#)**Element: executionException**

<b>Name</b>	executionException
<b>Type</b>	Exception
<b>Nillable</b>	no
<b>Abstract</b>	no

**Logical Diagram**

**XML Instance Representation**

```

<executionException
fpmVersion=" xsd:token (value comes from list: {'5-0'}) [1]
  
```

*Indicate which version of the FpML Schema an FpML message adheres to.'*

"

```

expectedBuild= "xsd:positiveInteger [0..1]
'This optional attribute can be supplied by a message creator in an FpML instance to
specify which build number of the schema was used to define the message when it was generated.'

"
actualBuild= "8 [0..1]
'The specific build number of this schema version. This attribute is not included in
an instance document. Instead, it is supplied by the XML parser when the document is
validated against the FpML schema and indicates the build number of the schema file. Every
time FpML publishes a change to the schema, validation rules, or examples within a version
(e.g., version 4.2) the actual build number is incremented. If no changes have been
made between releases within a version (i.e. from Trial Recommendation to Recommendation)
the actual build number stays the same.'


">
<header> ExceptionMessageHeader </header> [1]
<validation> Validation </validation> [0..*]
Start Group: Correlation.model [0..1]
  <correlationId> CorrelationId </correlationId> [1]
    'A qualified identifier used to correlate between messages'

End Group: Correlation.model
Start Group: Sequence.model [0..1]
  <sequenceNumber> xsd:positiveInteger </sequenceNumber> [1]
    'A numeric value that can be used to order messages with the same correlation identifier
from the same sender.'

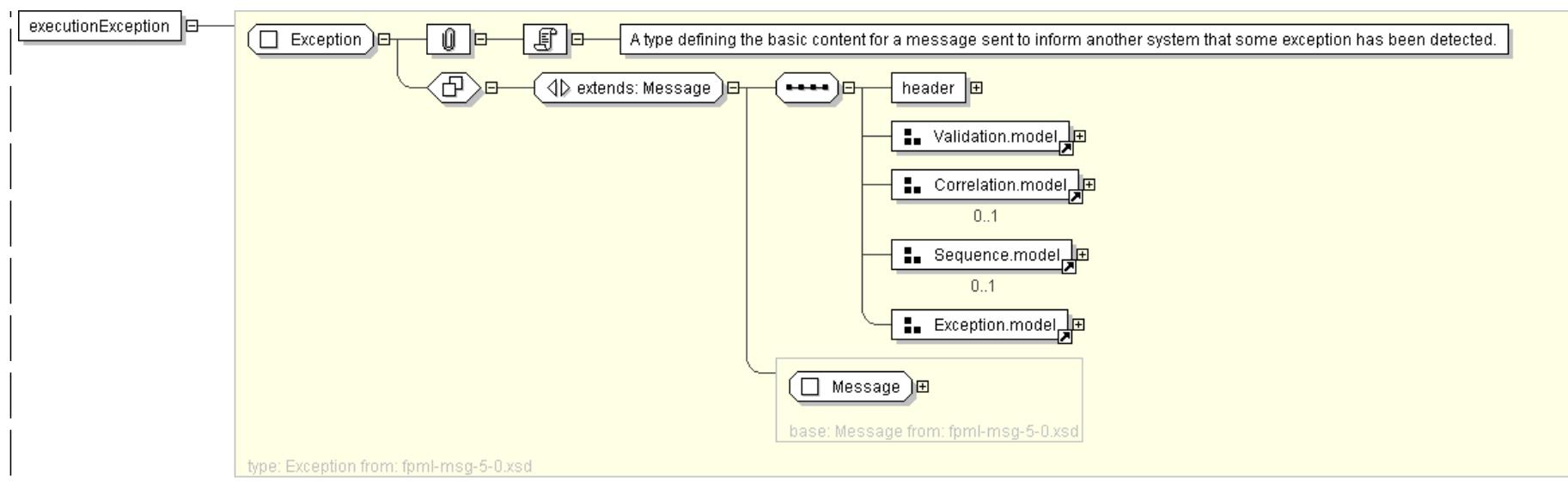

End Group: Sequence.model
  <reason> Reason </reason> [1..*]
    'An instance of the Reason type used to record the nature of any errors associated with
a message.'


<additionalData> AdditionalData </additionalData> [0..1]
  'Any string of additional data that may help the message processor, for example in a
rejection message this might contain a code value or the text of the original request (within
a CDATA section).'


</executionException>

```

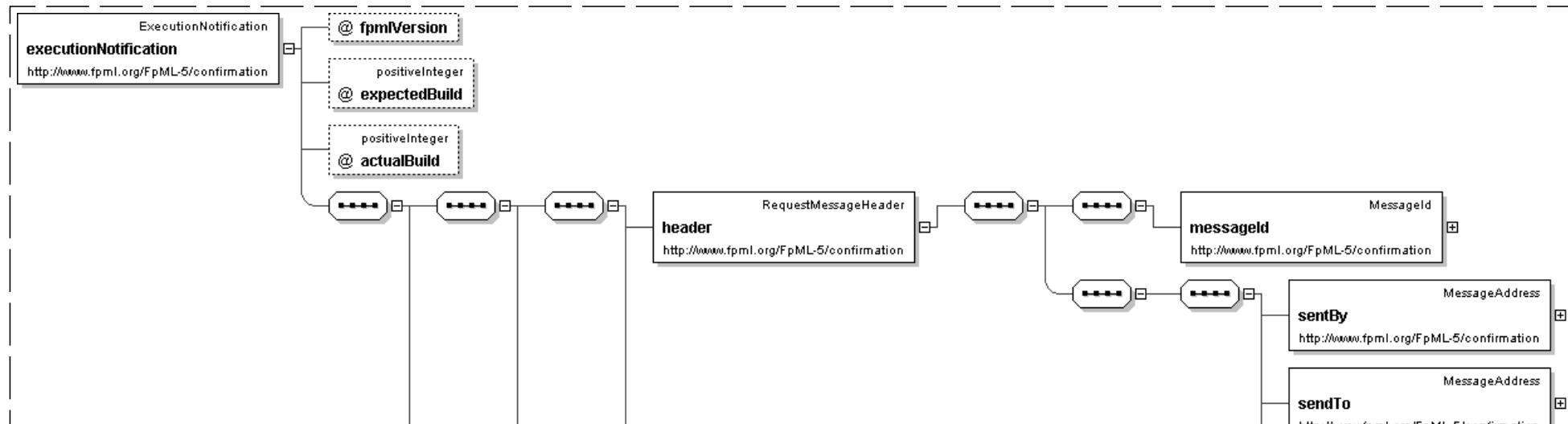
**Diagram**

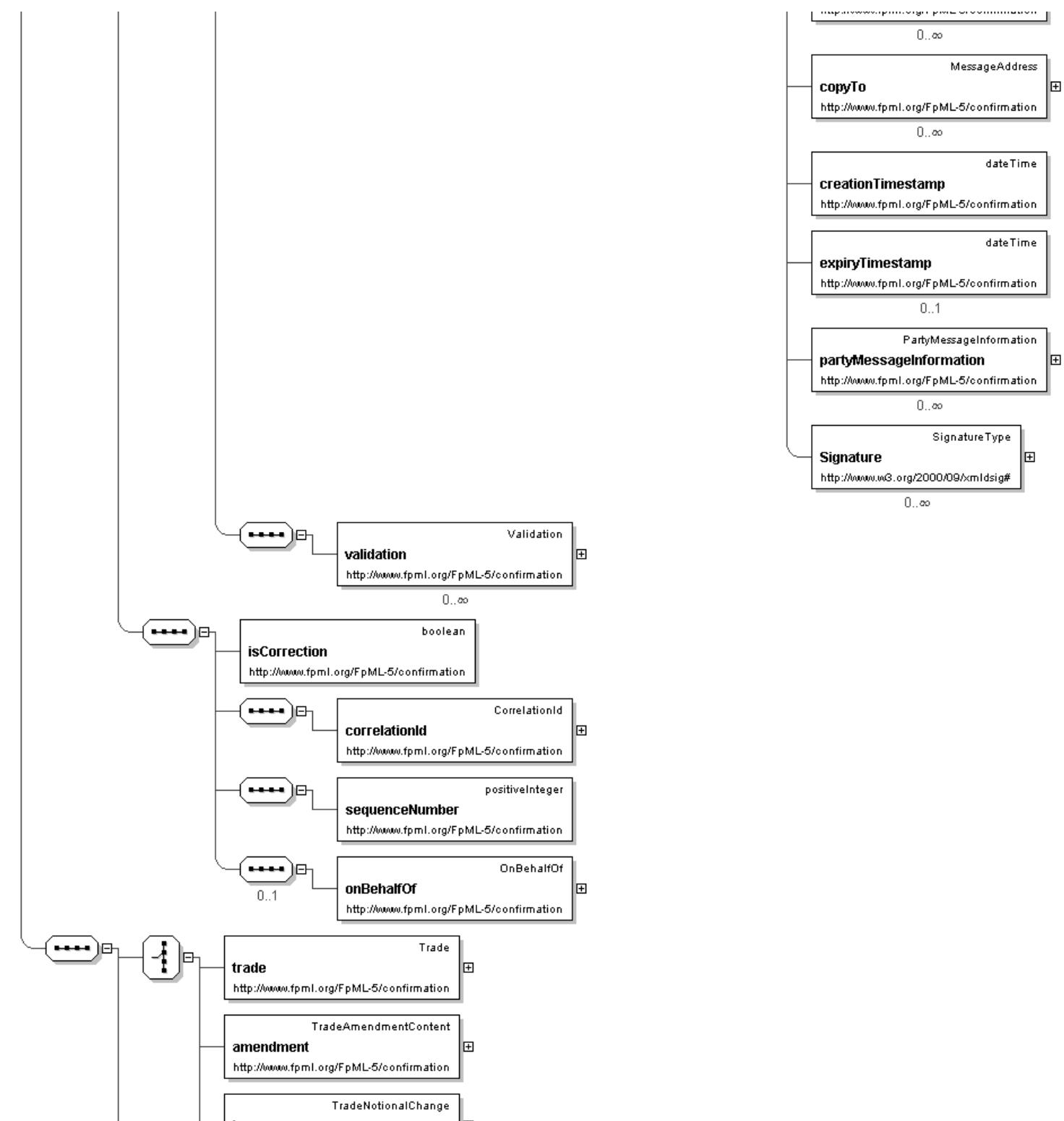
**Schema Component Representation**

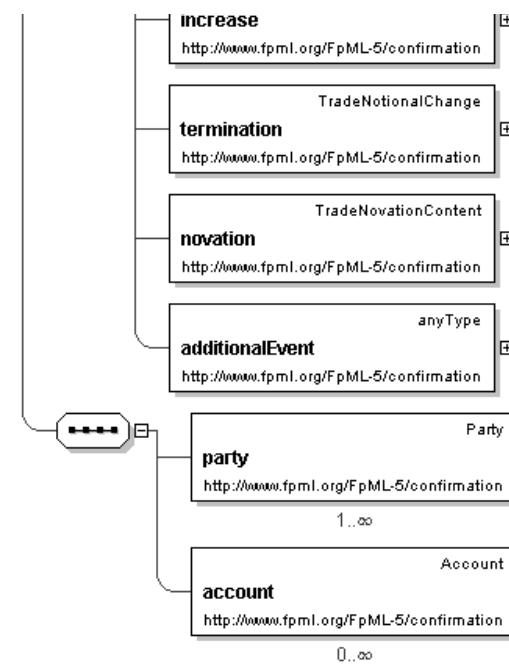
```
<xsd:element name="executionException" type="Exception" />
```

[top](#)**Element: executionNotification**

Name	executionNotification
Type	<a href="#">ExecutionNotification</a>
Nillable	no
Abstract	no

**Logical Diagram**



**XML Instance Representation**

```

<executionNotification
fpmlVersion=" xsd:token (value comes from list: {'5-0'}) [1]
'Indicate which version of the FpML Schema an FpML message adheres to.'

"
expectedBuild=" xsd:positiveInteger [0..1]
'This optional attribute can be supplied by a message creator in an FpML instance to
specify which build number of the schema was used to define the message when it was generated.'

"
actualBuild="8 [0..1]
'The specific build number of this schema version. This attribute is not included in
an instance document. Instead, it is supplied by the XML parser when the document is
validated against the FpML schema and indicates the build number of the schema file. Every
time FpML publishes a change to the schema, validation rules, or examples within a version
(e.g., version 4.2) the actual build number is incremented. If no changes have been
made between releases within a version (i.e. from Trial Recommendation to Recommendation)
the actual build number stays the same.'

">
<header> RequestMessageHeader </header> [1]
<validation> Validation </validation> [0..*]
<isCorrection> xsd:boolean </isCorrection> [1]
'Indicates if this message corrects an earlier request.'

<correlationId> CorrelationId </correlationId> [1]
'A qualified identifier used to correlate between messages'

<sequenceNumber> xsd:positiveInteger </sequenceNumber> [1]
'A numeric value that can be used to order messages with the same correlation identifier
  
```

*from the same sender.'*

Start Group: OnBehalfOf.model [0..1]

<onBehalfOf> OnBehalfOf </onBehalfOf> [1]

*'Indicates which party (and accounts) a trade is being processed for.'*

End Group: OnBehalfOf.model

| Start Choice [1]

```
<trade> Trade </trade> [1]
```

<amendment> TradeAmendmentContent </amendment> [ 1 ]

<increase> TradeNotionalChange </increase> [1]

```
<termination> TradeNotionalChange </termination>
```

```
<novation> TradeNovationContent </novation> [1]
```

```
<additionalEvent> ... </additionalEvent> [1]
```

End Choice

<party> Party </party> [1..\*]

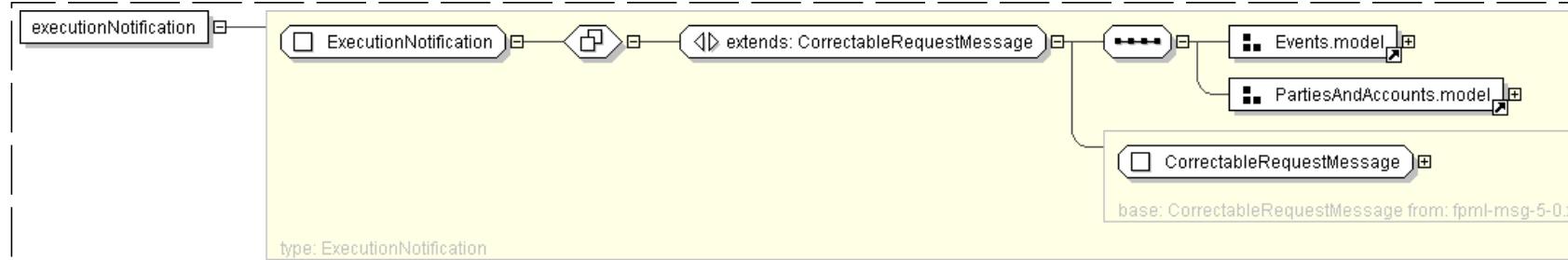
*'A legal entity or a subdivision of a legal entity.', 'Parties can perform multiple roles in a trade lifecycle. For example, the principal parties obligated to make payments from time to time during the term of the trade, but may include other parties involved in, or incidental to, the trade, such as parties acting in the role of novation transferor/transferee, broker, calculation agent, etc. In FpML roles are defined in multiple places within a document.'*

<account> Account </account> [0..\*]

*'Optional account information used to precisely define the origination and destination of financial instruments.'*

```
</executionNotification>
```

## Diagram



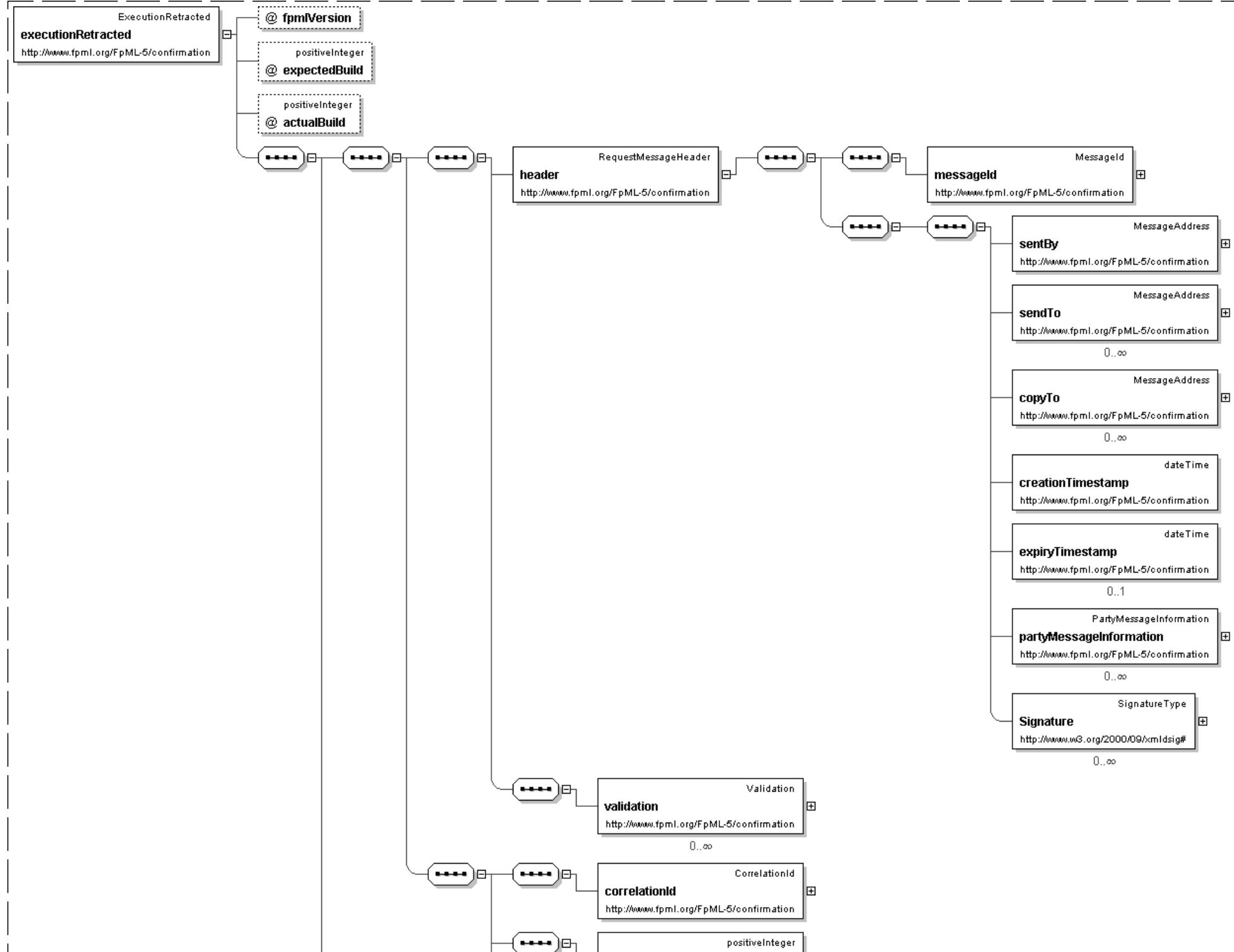
## Schema Component Representation

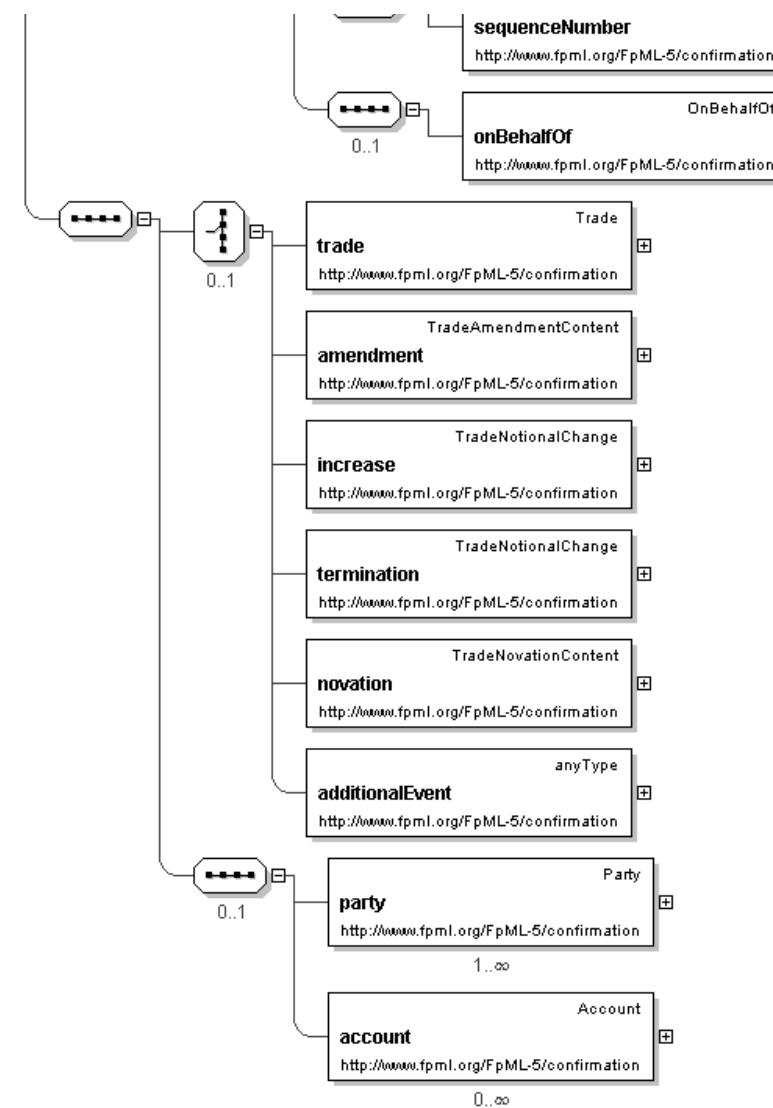
```
<xsd:element name="executionNotification" type="ExecutionNotification" />
```

top

## Element: executionRetracted

<b>Name</b>	executionRetracted
<b>Type</b>	<a href="#">ExecutionRetracted</a>
<b>Nillable</b>	no
<b>Abstract</b>	no

**Logical Diagram**

**XML Instance Representation**

```

<executionRetracted
fpmlVersion=" xsd:token (value comes from list: {'5-0'}) [1]
'Indicate which version of the FpML Schema an FpML message adheres to.'
"
expectedBuild=" xsd:positiveInteger [0..1]
'This optional attribute can be supplied by a message creator in an FpML instance to
specify which build number of the schema was used to define the message when it was generated.'
"
actualBuild="8 [0..1]
'The specific build number of this schema version. This attribute is not included in
an instance document. Instead, it is supplied by the XML parser when the document is
'
  
```

'validated against the FpML schema and indicates the build number of the schema file. Every time FpML publishes a change to the schema, validation rules, or examples within a version (e.g., version 4.2) the actual build number is incremented. If no changes have been made between releases within a version (i.e. from Trial Recommendation to Recommendation) the actual build number stays the same.'

```
">
<header> RequestMessageHeader </header> [1]
<validation> Validation </validation> [0..*]
<correlationId> CorrelationId </correlationId> [1]
'A qualified identifier used to correlate between messages'

<sequenceNumber> xsd:positiveInteger </sequenceNumber> [1]
'A numeric value that can be used to order messages with the same correlation identifier
from the same sender.'
```

```
Start Group: OnBehalfOf.model [0..1]
<onBehalfOf> OnBehalfOf </onBehalfOf> [1]
'Indicates which party (and accounts) a trade is being processed for.'
```

End Group: OnBehalfOf.model

Start Group: Events.model [0..1]

```
Start Choice [1]
  <trade> Trade </trade> [1]
  <amendment> TradeAmendmentContent </amendment> [1]
  <increase> TradeNotionalChange </increase> [1]
  <termination> TradeNotionalChange </termination> [1]
  <novation> TradeNovationContent </novation> [1]
  <additionalEvent> ... </additionalEvent> [1]
```

End Choice

End Group: Events.model

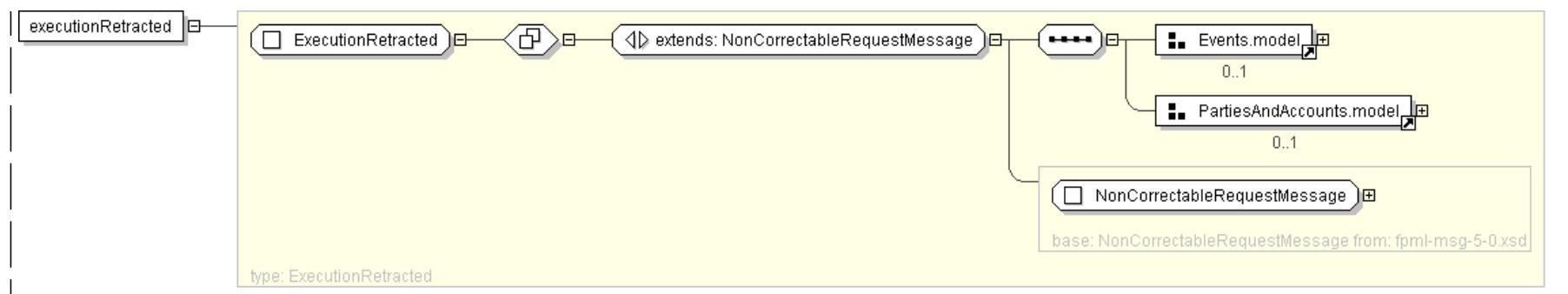
Start Group: PartiesAndAccounts.model [0..1]
<party> Party </party> [1..\*]

'A legal entity or a subdivision of a legal entity.', 'Parties can perform multiple roles in a trade lifecycle. For example, the principal parties obligated to make payments from time to time during the term of the trade, but may include other parties involved in, or incidental to, the trade, such as parties acting in the role of novation transferor/ transferee, broker, calculation agent, etc. In FpML roles are defined in multiple places within a document.'

```
<account> Account </account> [0..*]
'Optional account information used to precisely define the origination and destination
of financial instruments.'
```

End Group: PartiesAndAccounts.model
</executionRetracted>

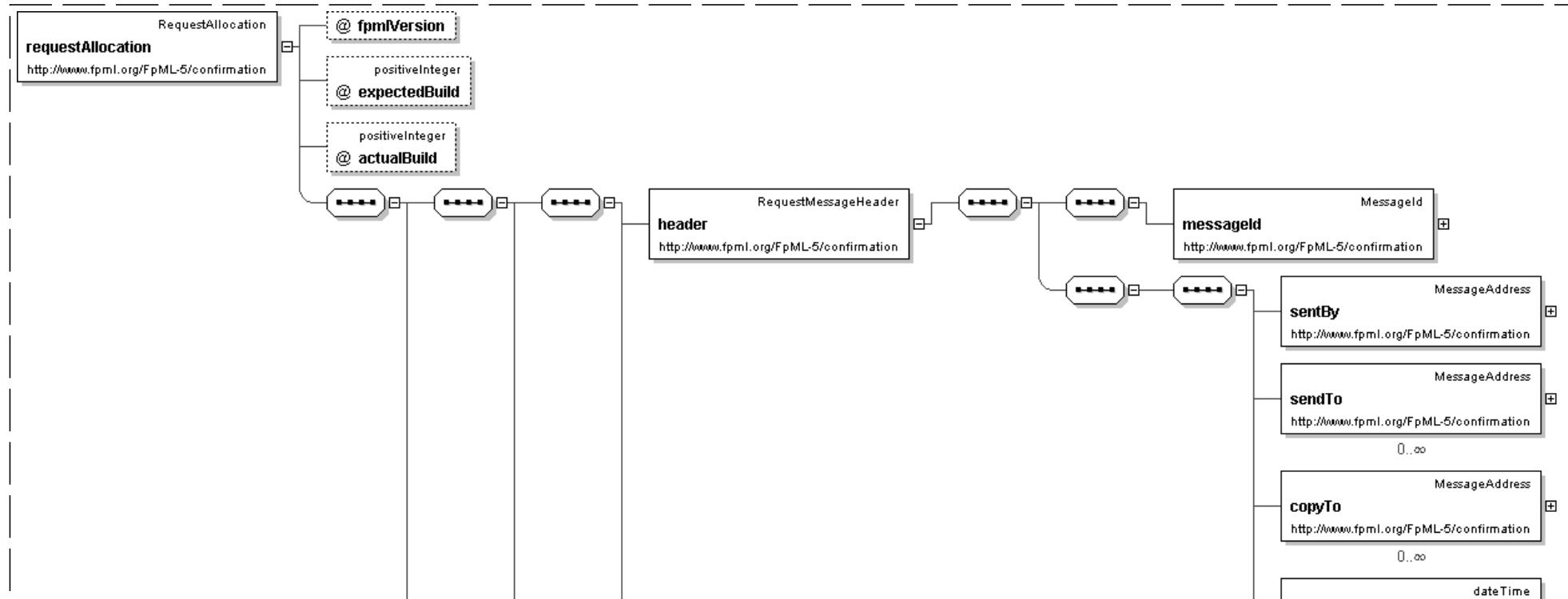
## Diagram

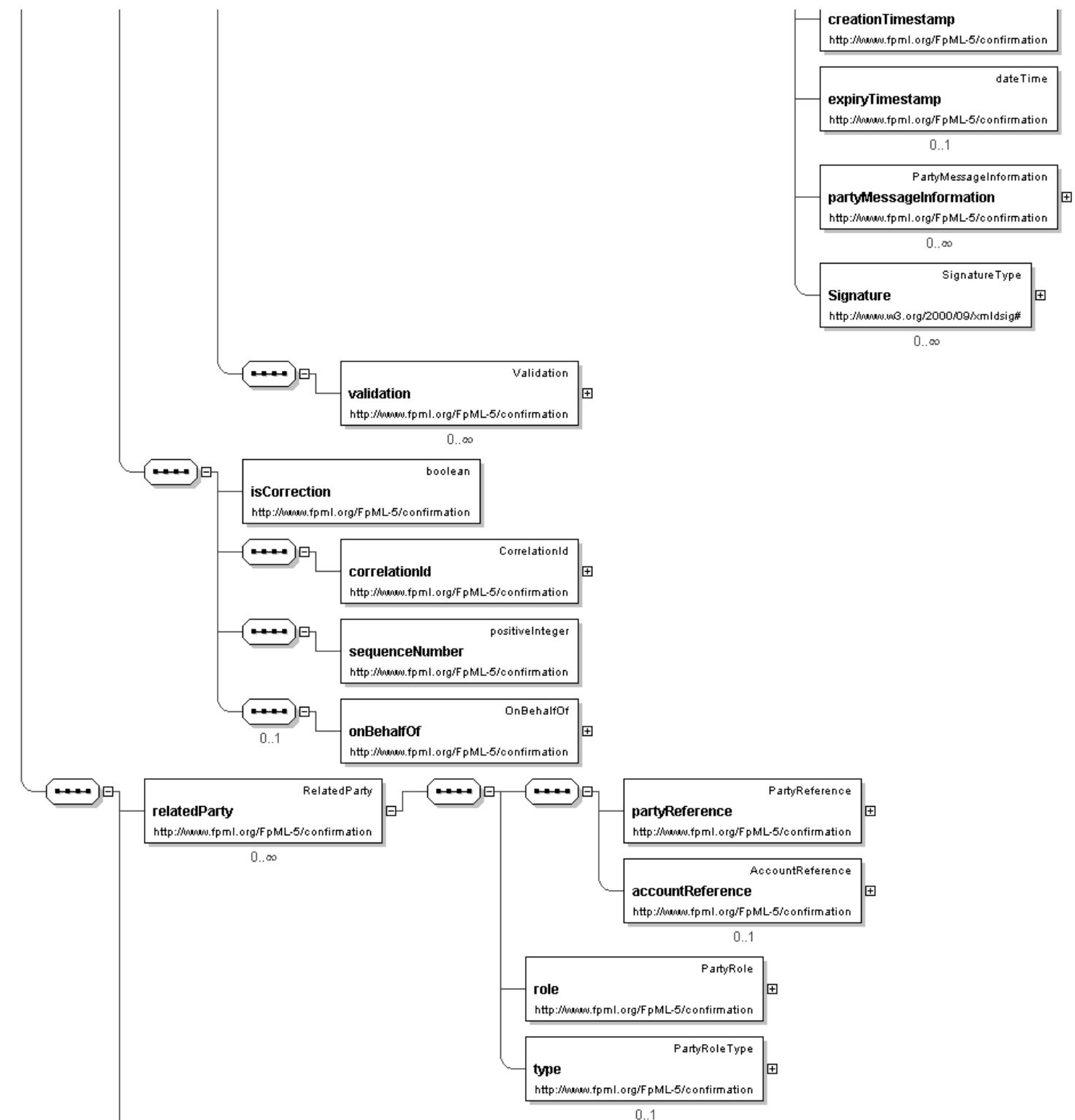
**Schema Component Representation**

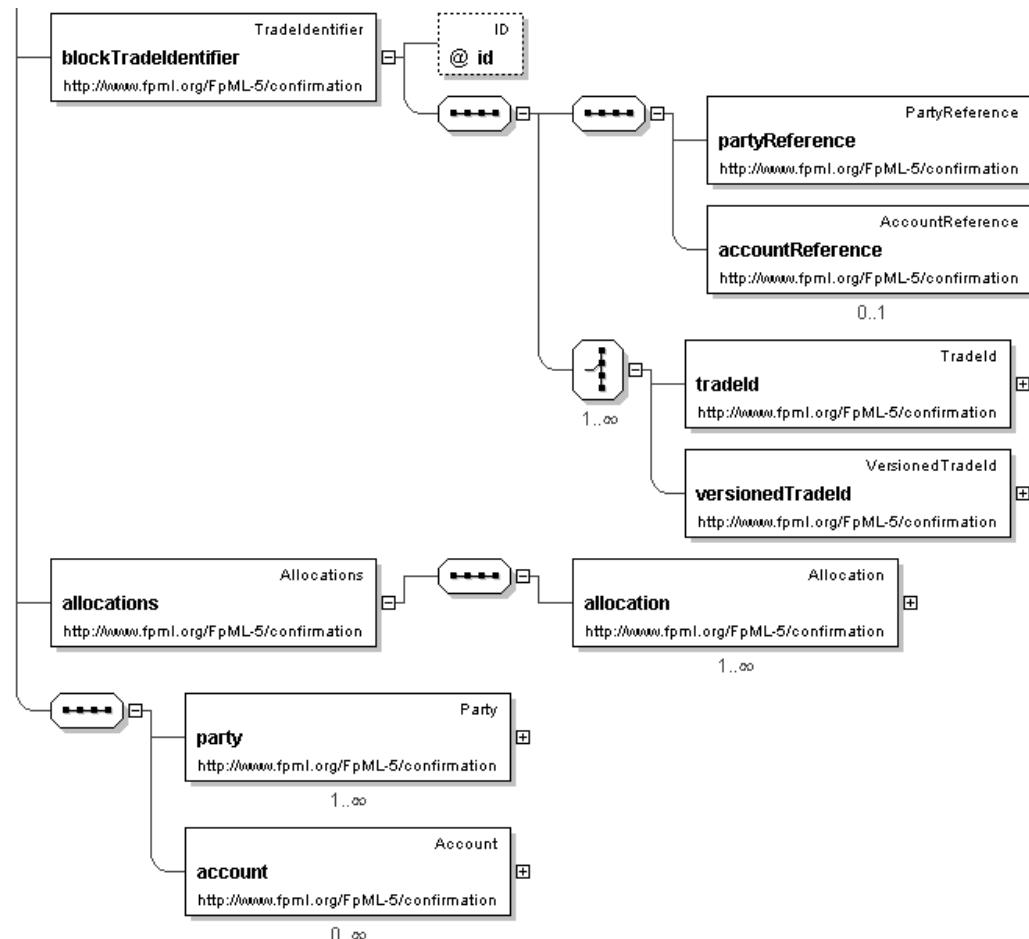
```
<xsd:element name="executionRetracted" type="ExecutionRetracted" />
```

[top](#)**Element: requestAllocation**

<b>Name</b>	requestAllocation
<b>Type</b>	<a href="#">RequestAllocation</a>
<b>Nillable</b>	no
<b>Abstract</b>	no

**Logical Diagram**





#### XML Instance Representation

```

<requestAllocation
  fpmlVersion=" xsd:token (value comes from list: {'5-0'}) [1]
  'Indicate which version of the FpML Schema an FpML message adheres to.'

  "
  expectedBuild=" xsd:positiveInteger [0..1]
  'This optional attribute can be supplied by a message creator in an FpML instance to
  specify which build number of the schema was used to define the message when it was generated.'

  "
  actualBuild="8 [0..1]
  'The specific build number of this schema version. This attribute is not included in
  an instance document. Instead, it is supplied by the XML parser when the document is
  validated against the FpML schema and indicates the build number of the schema file. Every
  time FpML publishes a change to the schema, validation rules, or examples within a version
  (e.g., version 4.2) the actual build number is incremented. If no changes have been
  made between releases within a version (i.e. from Trial Recommendation to Recommendation)
  the actual build number stays the same.'
  ">
  
```

```

<header> RequestMessageHeader </header> [1]
<validation> Validation </validation> [0..*]
<isCorrection> xsd:boolean </isCorrection> [1]
'Indicates if this message corrects an earlier request.'

<correlationId> CorrelationId </correlationId> [1]
'A qualified identifier used to correlate between messages'

<sequenceNumber> xsd:positiveInteger </sequenceNumber> [1]
'A numeric value that can be used to order messages with the same correlation identifier
from the same sender.'

Start Group: OnBehalfOf.model [0..1]
<onBehalfOf> OnBehalfOf </onBehalfOf> [1]
'Indicates which party (and accounts) a trade is being processed for.'

End Group: OnBehalfOf.model
<relatedParty> RelatedParty </relatedParty> [0..*]
'Identifies a related party performing a role within the transaction.'

<blockTradeIdentifier> TradeIdentifier </blockTradeIdentifier> [1]
<allocations> Allocations </allocations> [1]
<party> Party </party> [1..*]

'A legal entity or a subdivision of a legal entity.', 'Parties can perform multiple roles in
a trade lifecycle. For example, the principal parties obligated to make payments from time
to time during the term of the trade, but may include other parties involved in, or
incidental to, the trade, such as parties acting in the role of novation transferor/
transferee, broker, calculation agent, etc. In FpML roles are defined in multiple places
within a document..'

<account> Account </account> [0..*]
'Optional account information used to precisely define the origination and destination
of financial instruments.'

</requestAllocation>

```

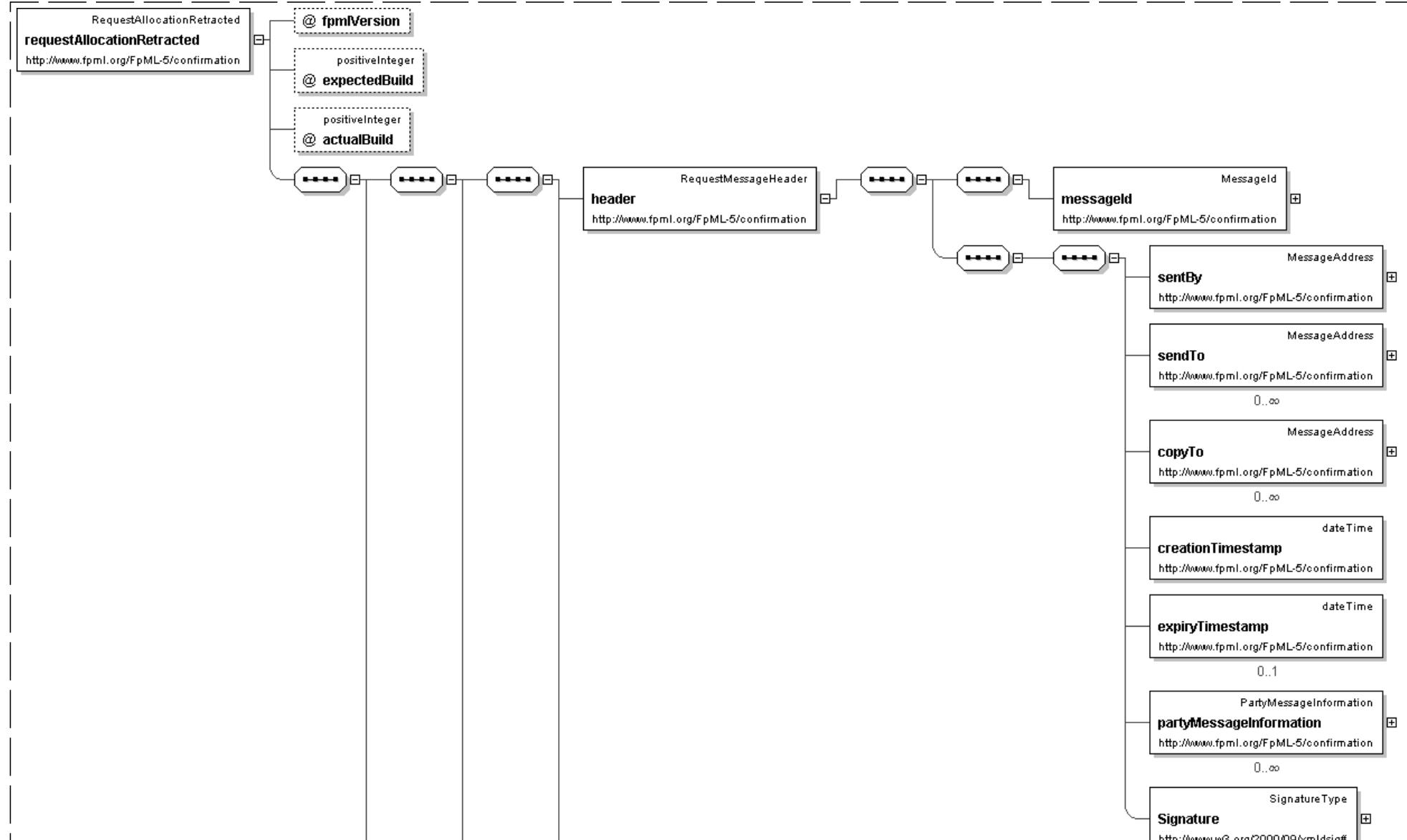
**Diagram****Schema Component Representation**

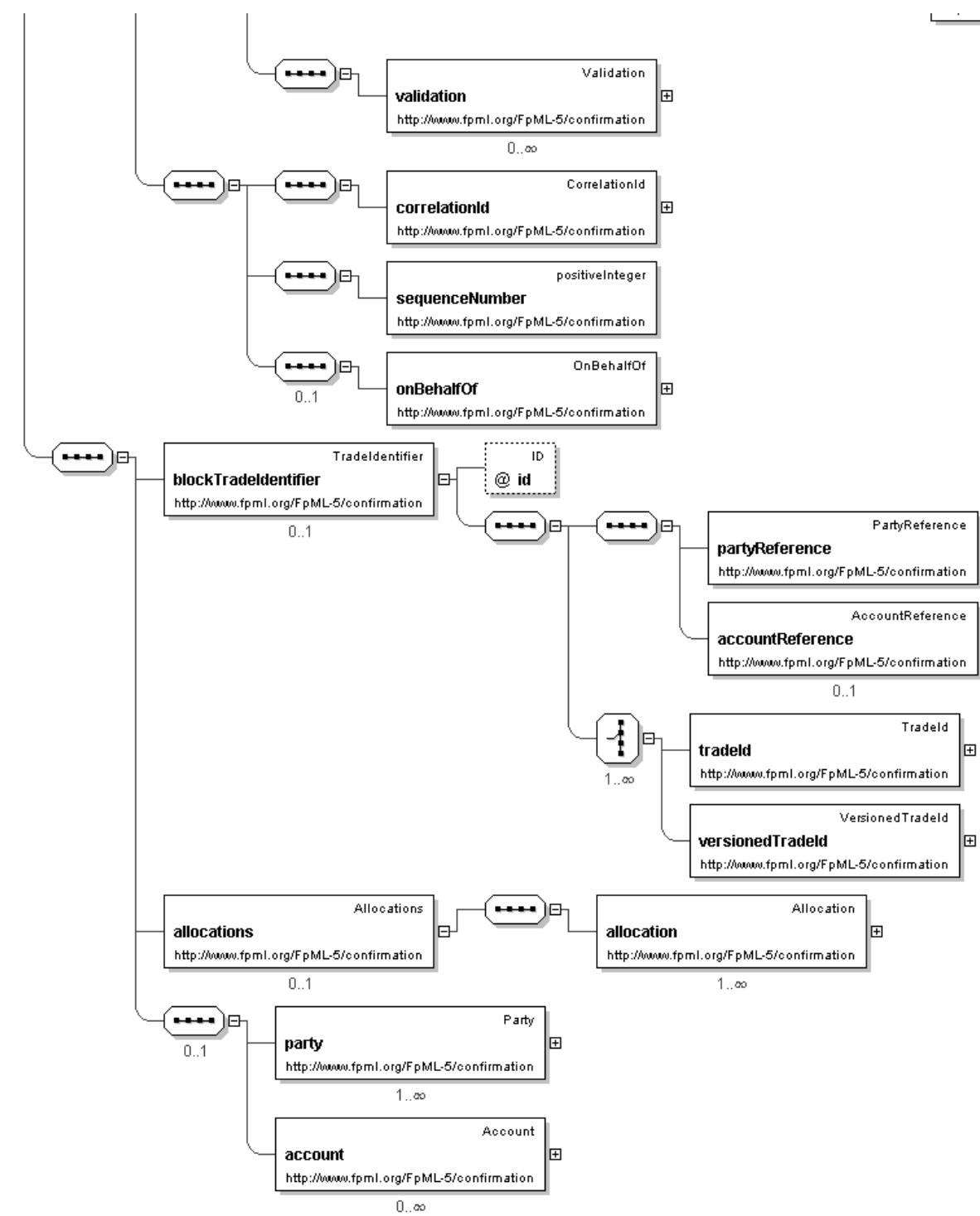
```
<xsd:element name="requestAllocation" type=" RequestAllocation " />
```

## Element: requestAllocationRetracted

Name	requestAllocationRetracted
Type	<a href="#">RequestAllocationRetracted</a>
Nillable	no
Abstract	no

### Logical Diagram



**XML Instance Representation**

```

'<requestAllocationRetracted
fpmVersion=" xsd:token (value comes from list: {'5-0'}) [1]
'Indicate which version of the FpML Schema an FpML message adheres to.'

"
expectedBuild=" xsd:positiveInteger [0..1]
'This optional attribute can be supplied by a message creator in an FpML instance to
specify which build number of the schema was used to define the message when it was generated.

"
actualBuild="8 [0..1]
'The specific build number of this schema version. This attribute is not included in
an instance document. Instead, it is supplied by the XML parser when the document is
validated against the FpML schema and indicates the build number of the schema file. Every
time FpML publishes a change to the schema, validation rules, or examples within a version
(e.g., version 4.2) the actual build number is incremented. If no changes have been
made between releases within a version (i.e. from Trial Recommendation to Recommendation)
the actual build number stays the same.'


">
<header> RequestMessageHeader </header> [1]
<validation> Validation </validation> [0..*]
<correlationId> CorrelationId </correlationId> [1]
'A qualified identifier used to correlate between messages'

<sequenceNumber> xsd:positiveInteger </sequenceNumber> [1]
'A numeric value that can be used to order messages with the same correlation identifier
from the same sender.'


Start Group: OnBehalfOf.model [0..1]
<onBehalfOf> OnBehalfOf </onBehalfOf> [1]
'Indicates which party (and accounts) a trade is being processed for.'

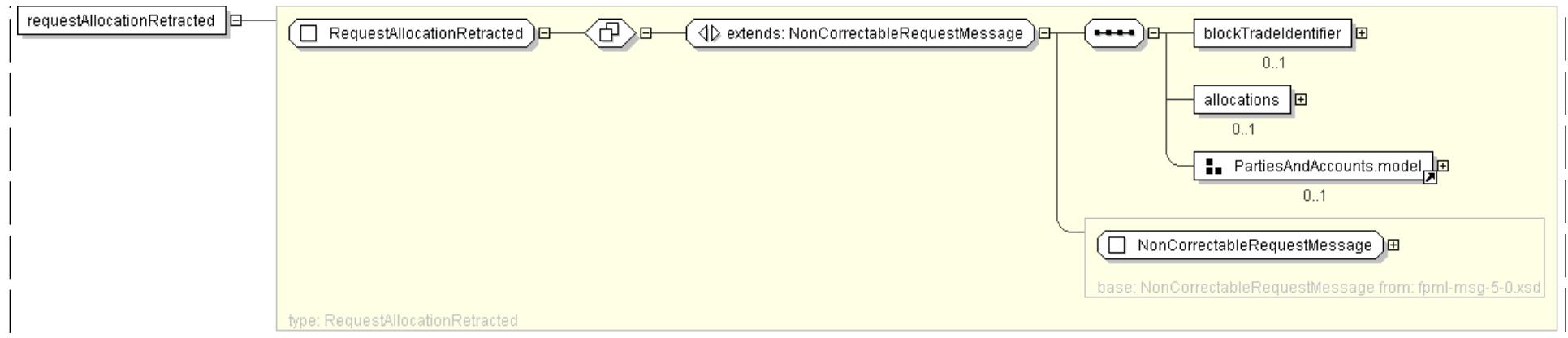

End Group: OnBehalfOf.model
<blockTradeIdentifier> TradeIdentifier </blockTradeIdentifier> [0..1]
<allocations> Allocations </allocations> [0..1]
Start Group: PartiesAndAccounts.model [0..1]
<party> Party </party> [1..*]
'A legal entity or a subdivision of a legal entity.', 'Parties can perform multiple roles in
a trade lifecycle. For example, the principal parties obligated to make payments from time
to time during the term of the trade, but may include other parties involved in, or
incidental to, the trade, such as parties acting in the role of novation transferor/
transferee, broker, calculation agent, etc. In FpML roles are defined in multiple places
within a document.'


<account> Account </account> [0..*]
'Optional account information used to precisely define the origination and destination
of financial instruments.'


End Group: PartiesAndAccounts.model
</requestAllocationRetracted>

```

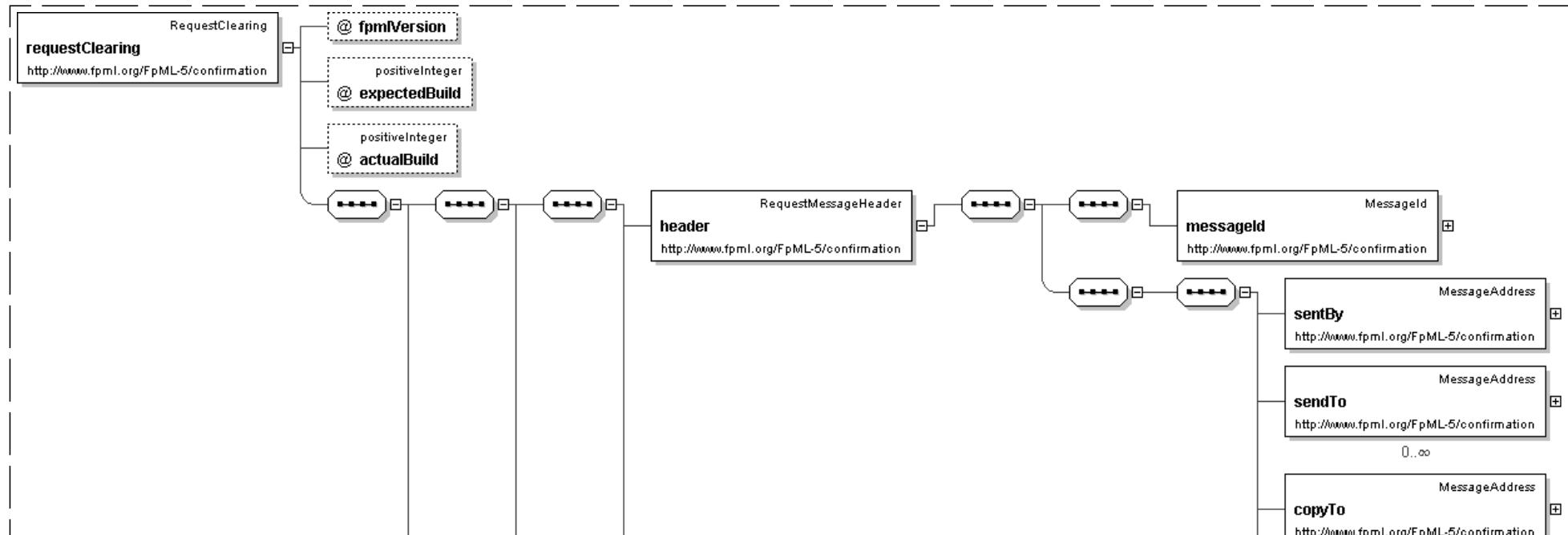
**Diagram**

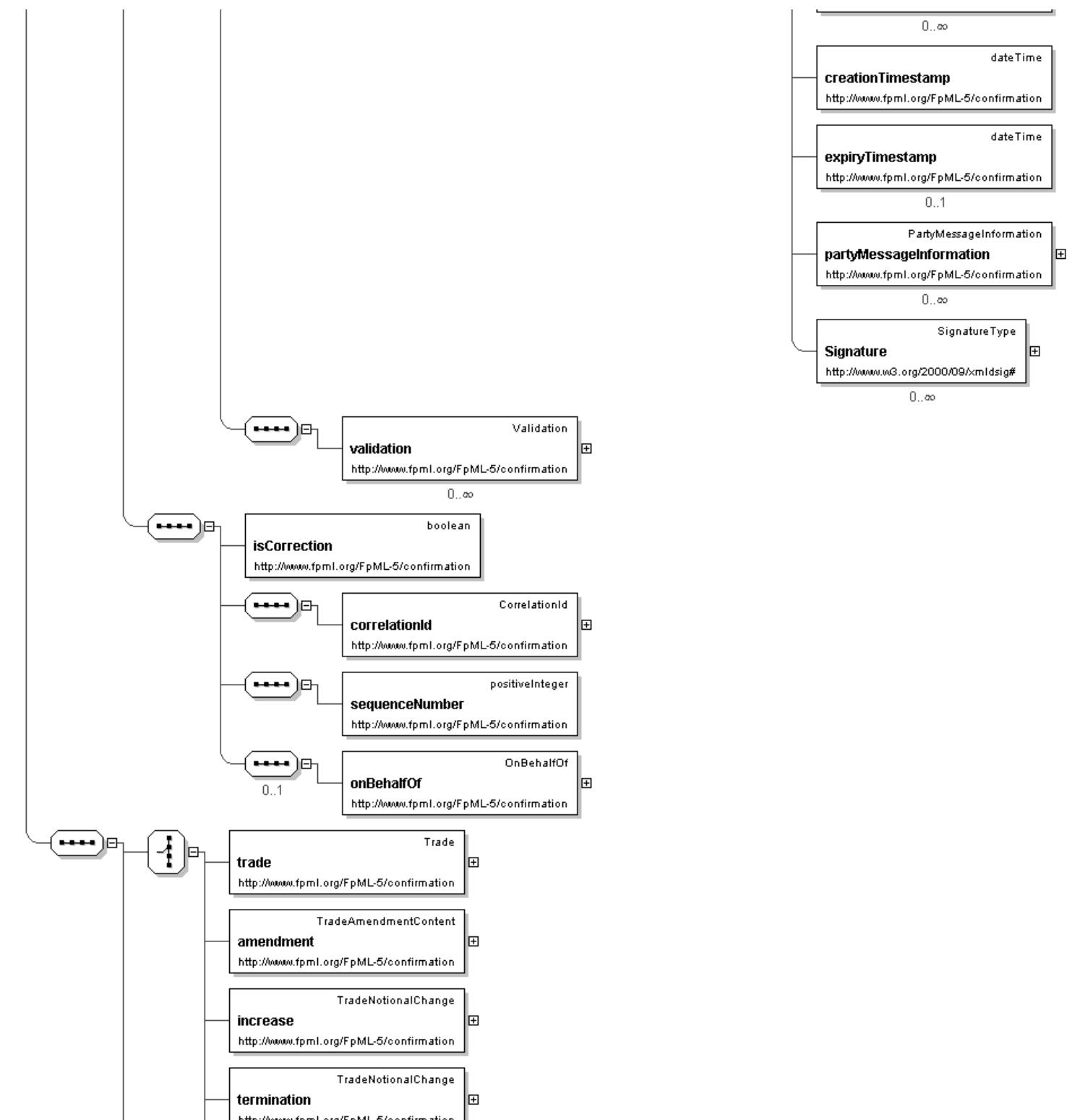
**Schema Component Representation**

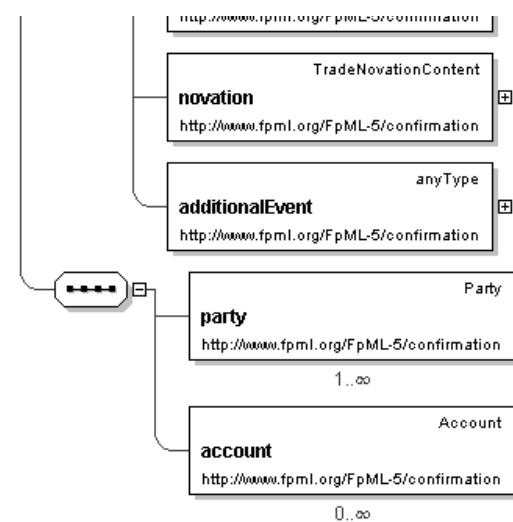
```
<xsd:element name="requestAllocationRetracted" type=" RequestAllocationRetracted " />
```

[top](#)**Element: requestClearing**

<b>Name</b>	requestClearing
<b>Type</b>	<code>RequestClearing</code>
<b>Nillable</b>	no
<b>Abstract</b>	no

**Logical Diagram**



**XML Instance Representation**

```

<requestClearing
fpmlVersion=" xsd:token (value comes from list: {'5-0'}) [1]
'Indicate which version of the FpML Schema an FpML message adheres to.'

"
expectedBuild=" xsd:positiveInteger [0..1]
'This optional attribute can be supplied by a message creator in an FpML instance to
specify which build number of the schema was used to define the message when it was generated.'

"
actualBuild="8 [0..1]
'The specific build number of this schema version. This attribute is not included in
an instance document. Instead, it is supplied by the XML parser when the document is
validated against the FpML schema and indicates the build number of the schema file. Every
time FpML publishes a change to the schema, validation rules, or examples within a version
(e.g., version 4.2) the actual build number is incremented. If no changes have been
made between releases within a version (i.e. from Trial Recommendation to Recommendation)
the actual build number stays the same.'

">
<header> RequestMessageHeader </header> [1]
<validation> Validation </validation> [0..*]
<isCorrection> xsd:boolean </isCorrection> [1]
'Indicates if this message corrects an earlier request.'

<correlationId> CorrelationId </correlationId> [1]
'A qualified identifier used to correlate between messages'

<sequenceNumber> xsd:positiveInteger </sequenceNumber> [1]
'A numeric value that can be used to order messages with the same correlation identifier
from the same sender.'

Start Group: OnBehalfOf.model [0..1]
<onBehalfOf> OnBehalfOf </onBehalfOf> [1]
'Indicates which party (and accounts) a trade is being processed for.'

```

```

End Group: OnBehalfOf.model
Start Choice [1]
  <trade> Trade </trade> [1]
  <amendment> TradeAmendmentContent </amendment> [1]
  <increase> TradeNotionalChange </increase> [1]
  <termination> TradeNotionalChange </termination> [1]
  <novation> TradeNovationContent </novation> [1]
  <additionalEvent> ... </additionalEvent> [1]
End Choice
<party> Party </party> [1..*]
'A legal entity or a subdivision of a legal entity.' , 'Parties can perform multiple roles in a trade lifecycle. For example, the principal parties obligated to make payments from time to time during the term of the trade, but may include other parties involved in, or incidental to, the trade, such as parties acting in the role of novation transferor/ transferee, broker, calculation agent, etc. In FpML roles are defined in multiple places within a document.'
<account> Account </account> [0..*]
'Optional account information used to precisely define the origination and destination of financial instruments.'

```

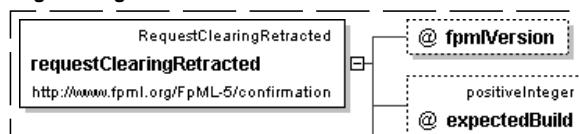
&lt;/requestClearing&gt;

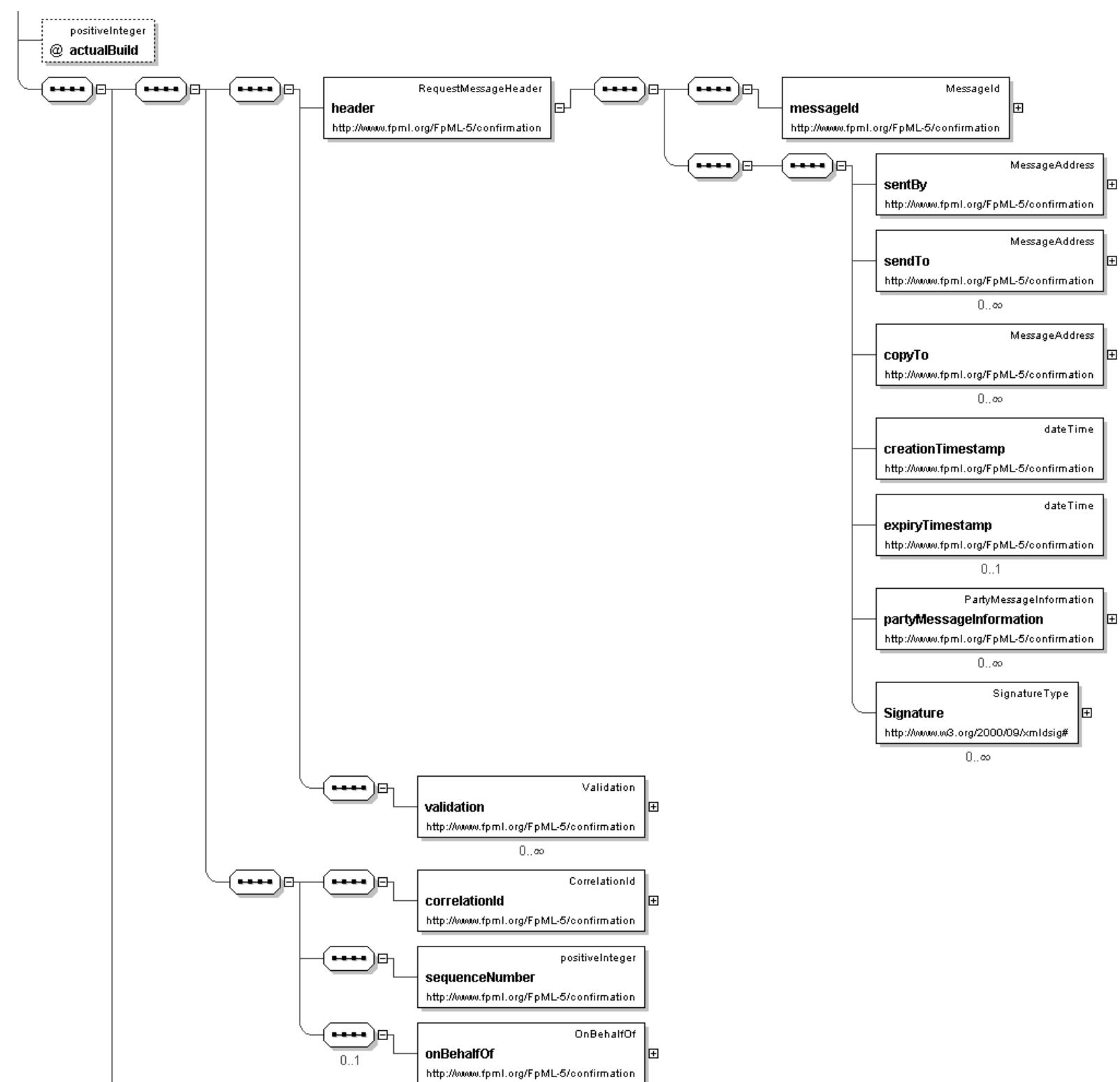
**Diagram****Schema Component Representation**

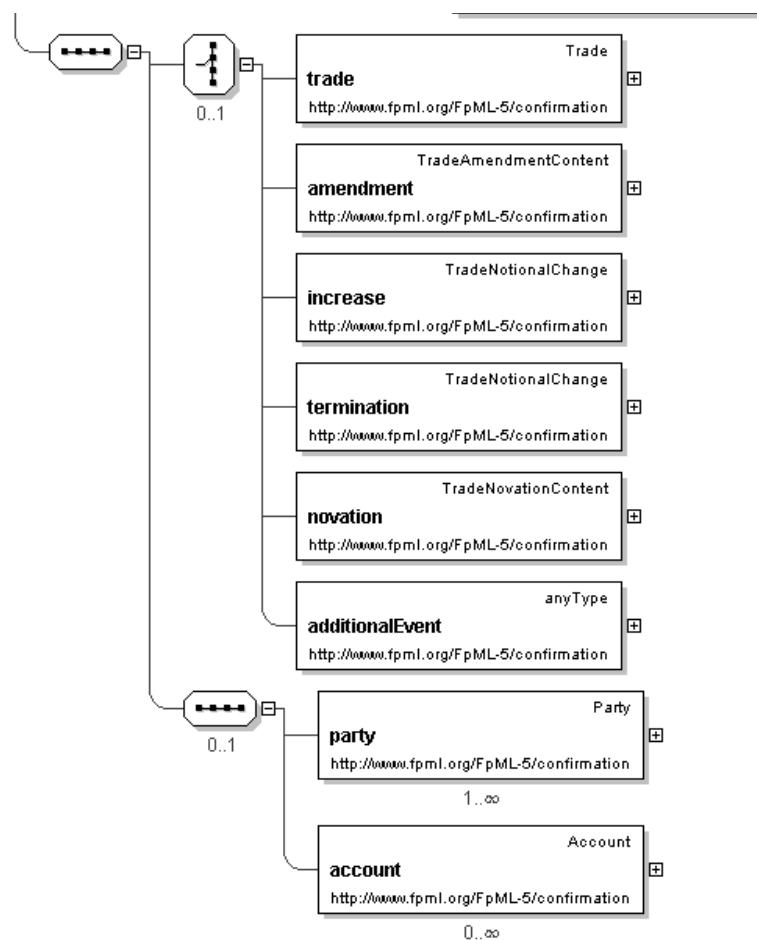
```
<xsd:element name="requestClearing" type=" RequestClearing " />
```

[top](#)**Element: requestClearingRetracted**

<b>Name</b>	requestClearingRetracted
<b>Type</b>	<a href="#">RequestClearingRetracted</a>
<b>Nillable</b>	no
<b>Abstract</b>	no

**Logical Diagram**





#### XML Instance Representation

```

<requestClearingRetracted
fpmlVersion=" xsd:token (value comes from list: {'5-0'}) [1]
'Indicate which version of the FpML Schema an FpML message adheres to.'

"
expectedBuild=" xsd:positiveInteger [0..1]
'This optional attribute can be supplied by a message creator in an FpML instance to
specify which build number of the schema was used to define the message when it was generated.'

"
actualBuild="8 [0..1]
'The specific build number of this schema version. This attribute is not included in
an instance document. Instead, it is supplied by the XML parser when the document is
validated against the FpML schema and indicates the build number of the schema file. Every
time FpML publishes a change to the schema, validation rules, or examples within a version
(e.g., version 4.2) the actual build number is incremented. If no changes have been
made between releases within a version (i.e. from Trial Recommendation to Recommendation)
the actual build number stays the same.'


">
  
```

```

<header> RequestMessageHeader </header> [1]
<validation> Validation </validation> [0..*]
<correlationId> CorrelationId </correlationId> [1]
'A qualified identifier used to correlate between messages'

<sequenceNumber> xsd:positiveInteger </sequenceNumber> [1]
'A numeric value that can be used to order messages with the same correlation identifier
from the same sender.'

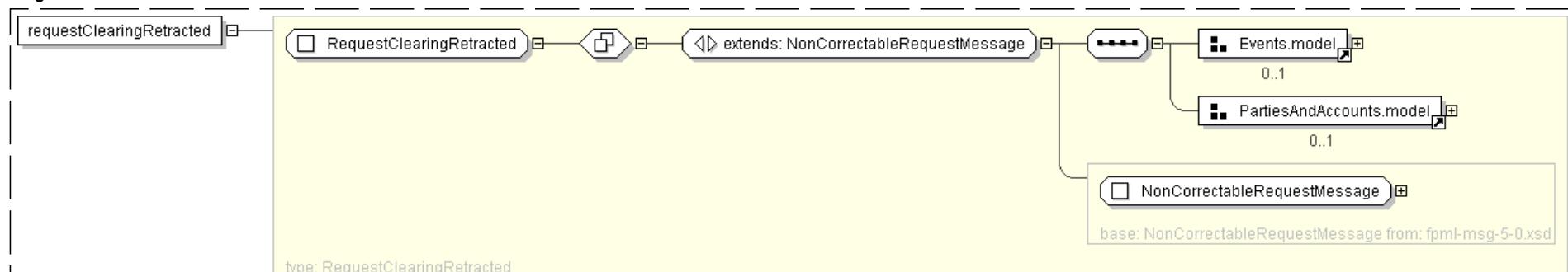
Start Group: OnBehalfOf.model [0..1]
<onBehalfOf> OnBehalfOf </onBehalfOf> [1]
'Indicates which party (and accounts) a trade is being processed for.'

End Group: OnBehalfOf.model
Start Group: Events.model [0..1]
Start Choice [1]
    <trade> Trade </trade> [1]
    <amendment> TradeAmendmentContent </amendment> [1]
    <increase> TradeNotionalChange </increase> [1]
    <termination> TradeNotionalChange </termination> [1]
    <novation> TradeNovationContent </novation> [1]
    <additionalEvent> ... </additionalEvent> [1]
End Choice
End Group: Events.model
Start Group: PartiesAndAccounts.model [0..1]
<party> Party </party> [1..*]
'A legal entity or a subdivision of a legal entity.', 'Parties can perform multiple roles in
a trade lifecycle. For example, the principal parties obligated to make payments from time
to time during the term of the trade, but may include other parties involved in, or
incidental to, the trade, such as parties acting in the role of novation transferor/
transferee, broker, calculation agent, etc. In FpML roles are defined in multiple places
within a document.'

<account> Account </account> [0..*]
'Optional account information used to precisely define the origination and destination
of financial instruments.'

End Group: PartiesAndAccounts.model
</requestClearingRetracted>

```

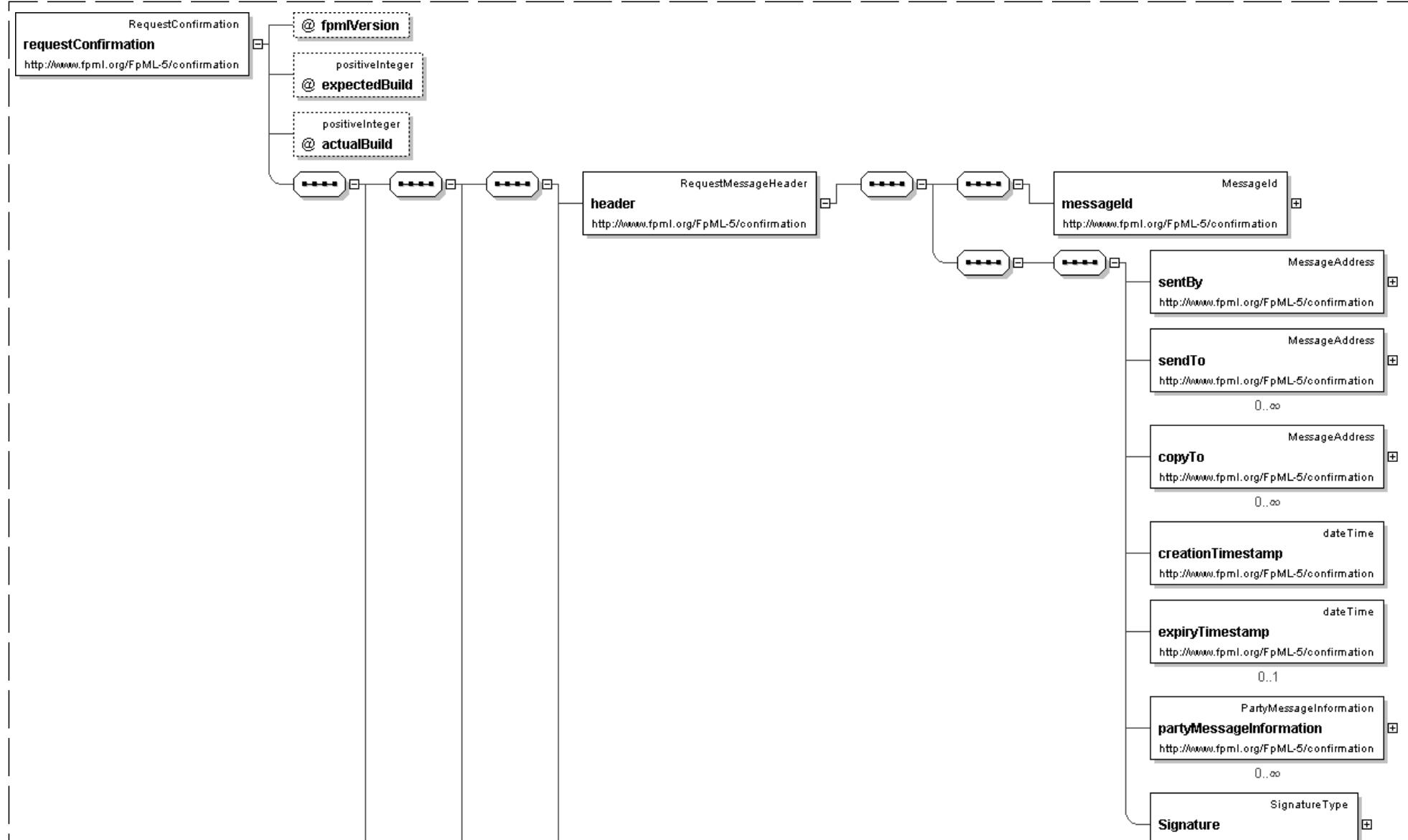
**Diagram****Schema Component Representation**

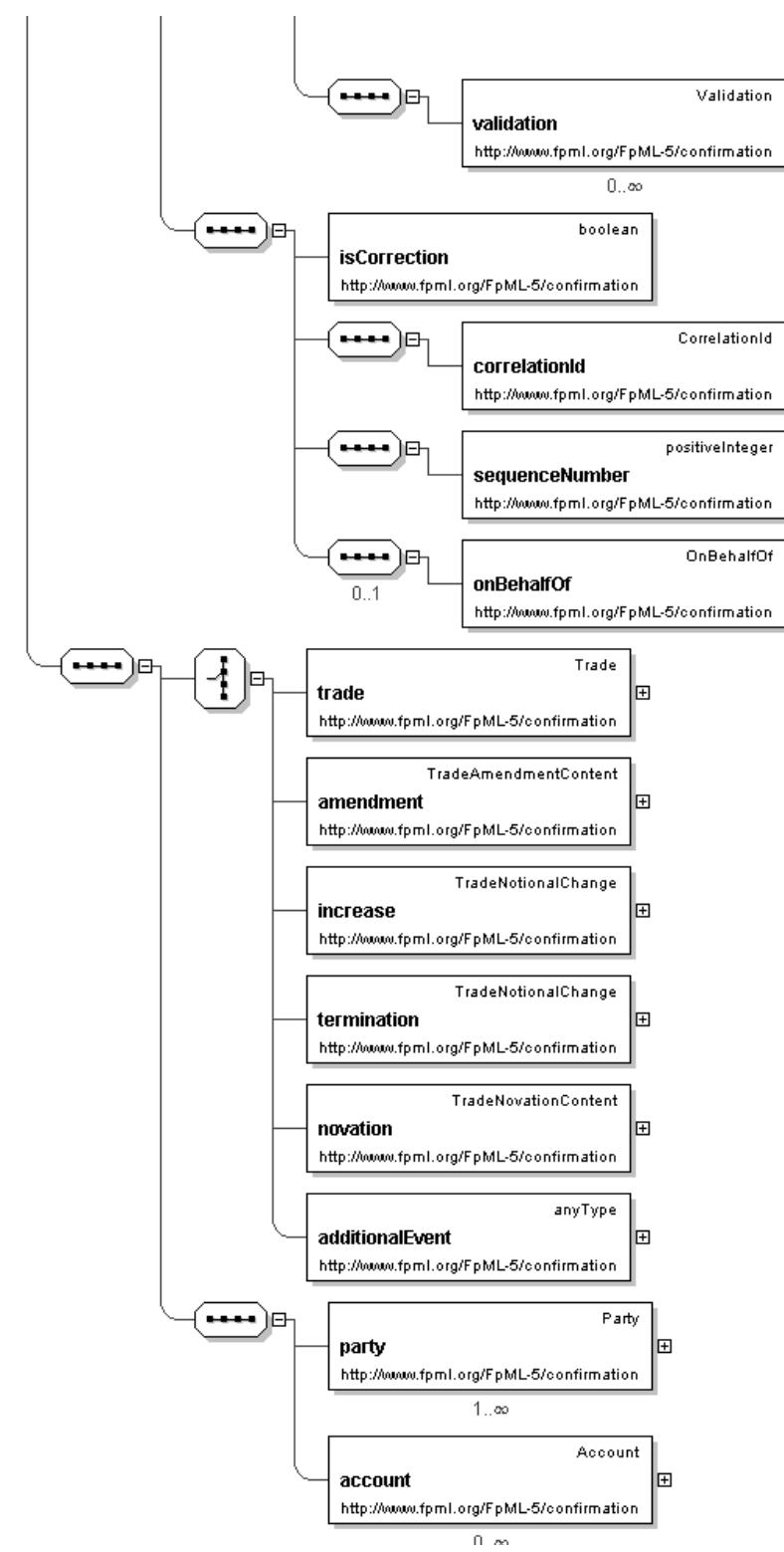
```
<xsd:element name="requestClearingRetracted" type=" RequestClearingRetracted " />
```

## Element: `requestConfirmation`

<b>Name</b>	requestConfirmation
<b>Type</b>	<a href="#">RequestConfirmation</a>
<b>Nillable</b>	no
<b>Abstract</b>	no

## Logical Diagram





**XML Instance Representation**

```

<requestConfirmation
  fpmlVersion=" xsd:token (value comes from list: {'5-0'}) [1]
  'Indicate which version of the FpML Schema an FpML message adheres to.'

  "
  expectedBuild=" xsd:positiveInteger [0..1]
  'This optional attribute can be supplied by a message creator in an FpML instance to
  specify which build number of the schema was used to define the message when it was generated.'

  "
  actualBuild="8 [0..1]
  'The specific build number of this schema version. This attribute is not included in
  an instance document. Instead, it is supplied by the XML parser when the document is
  validated against the FpML schema and indicates the build number of the schema file. Every
  time FpML publishes a change to the schema, validation rules, or examples within a version
  (e.g., version 4.2) the actual build number is incremented. If no changes have been
  made between releases within a version (i.e. from Trial Recommendation to Recommendation)
  the actual build number stays the same.'


  ">
    <header> RequestMessageHeader </header> [1]
    <validation> Validation </validation> [0..*]
    <isCorrection> xsd:boolean </isCorrection> [1]
    'Indicates if this message corrects an earlier request.'

    <correlationId> CorrelationId </correlationId> [1]
    'A qualified identifier used to correlate between messages'

    <sequenceNumber> xsd:positiveInteger </sequenceNumber> [1]
    'A numeric value that can be used to order messages with the same correlation identifier
    from the same sender.'


Start Group: OnBehalfOf.model [0..1]
  <onBehalfOf> OnBehalfOf </onBehalfOf> [1]
  'Indicates which party (and accounts) a trade is being processed for.'

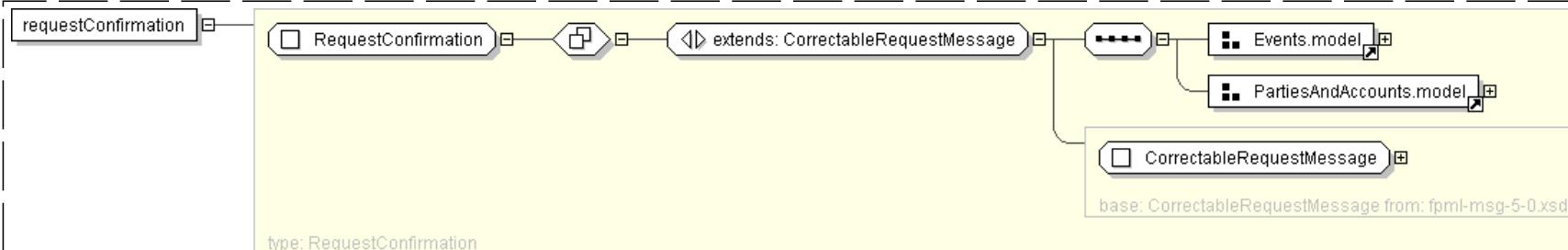

End Group: OnBehalfOf.model
Start Choice [1]
  <trade> Trade </trade> [1]
  <amendment> TradeAmendmentContent </amendment> [1]
  <increase> TradeNotionalChange </increase> [1]
  <termination> TradeNotionalChange </termination> [1]
  <novation> TradeNovationContent </novation> [1]
  <additionalEvent> ... </additionalEvent> [1]
End Choice
  <party> Party </party> [1..*]
  'A legal entity or a subdivision of a legal entity.', 'Parties can perform multiple roles in
  a trade lifecycle. For example, the principal parties obligated to make payments from time
  to time during the term of the trade, but may include other parties involved in, or
  incidental to, the trade, such as parties acting in the role of novation transferor/
  transferee, broker, calculation agent, etc. In FpML roles are defined in multiple places
  within a document.'


  <account> Account </account> [0..*]

```

'Optional account information used to precisely define the origination and destination of financial instruments.'

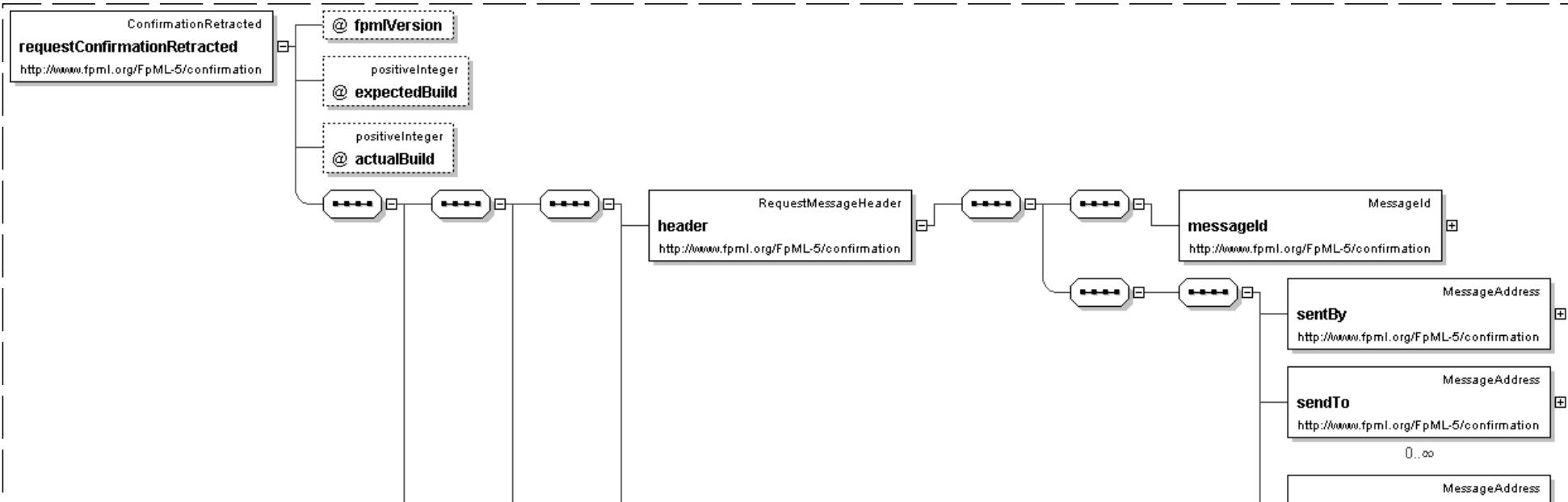
</requestConfirmation>

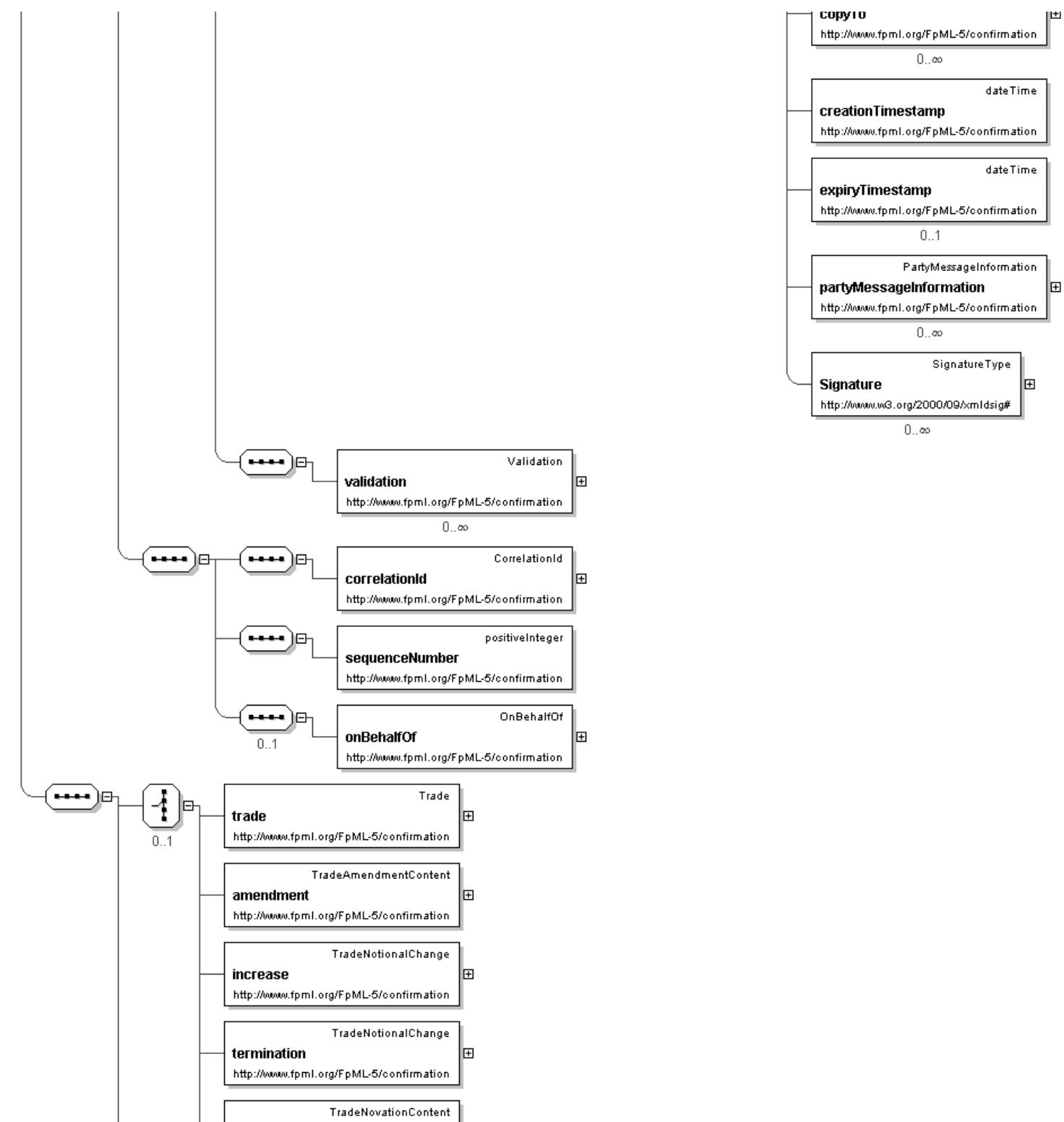
**Diagram****Schema Component Representation**

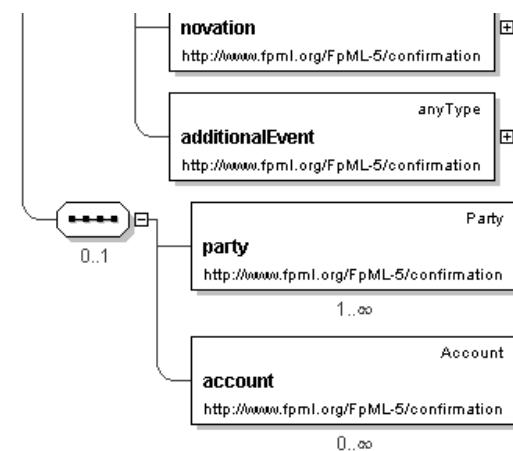
```
<xsd:element name="requestConfirmation" type=" RequestConfirmation " />
```

[top](#)
**Element: requestConfirmationRetracted**

<b>Name</b>	requestConfirmationRetracted
<b>Type</b>	<a href="#">ConfirmationRetracted</a>
<b>Nillable</b>	no
<b>Abstract</b>	no

**Logical Diagram**



**XML Instance Representation**

```

<requestConfirmationRetracted
fpmlVersion=" xsd:token (value comes from list: {'5-0'}) [1]
'Indicate which version of the FpML Schema an FpML message adheres to.'

"
expectedBuild=" xsd:positiveInteger [0..1]
'This optional attribute can be supplied by a message creator in an FpML instance to
specify which build number of the schema was used to define the message when it was generated.'

"
actualBuild="8 [0..1]
'The specific build number of this schema version. This attribute is not included in
an instance document. Instead, it is supplied by the XML parser when the document is
validated against the FpML schema and indicates the build number of the schema file. Every
time FpML publishes a change to the schema, validation rules, or examples within a version
(e.g., version 4.2) the actual build number is incremented. If no changes have been
made between releases within a version (i.e. from Trial Recommendation to Recommendation)
the actual build number stays the same.'

">
<header> RequestMessageHeader </header> [1]
<validation> Validation </validation> [0..*]
<correlationId> CorrelationId </correlationId> [1]
'A qualified identifier used to correlate between messages'

<sequenceNumber> xsd:positiveInteger </sequenceNumber> [1]
'A numeric value that can be used to order messages with the same correlation identifier
from the same sender.'

Start Group: OnBehalfOf.model [0..1]
<onBehalfOf> OnBehalfOf </onBehalfOf> [1]
'Indicates which party (and accounts) a trade is being processed for.'

End Group: OnBehalfOf.model
Start Group: Events.model [0..1]
Start Choice [1]
    <trade> Trade </trade> [1]
    <amendment> TradeAmendmentContent </amendment> [1]

```

```

<increase> TradeNotionalChange </increase> [1]
<termination> TradeNotionalChange </termination> [1]
<novation> TradeNovationContent </novation> [1]
<additionalEvent> ... </additionalEvent> [1]

End Choice
End Group: Events.model
Start Group: PartiesAndAccounts.model [0..1]
<party> Party </party> [1..*]

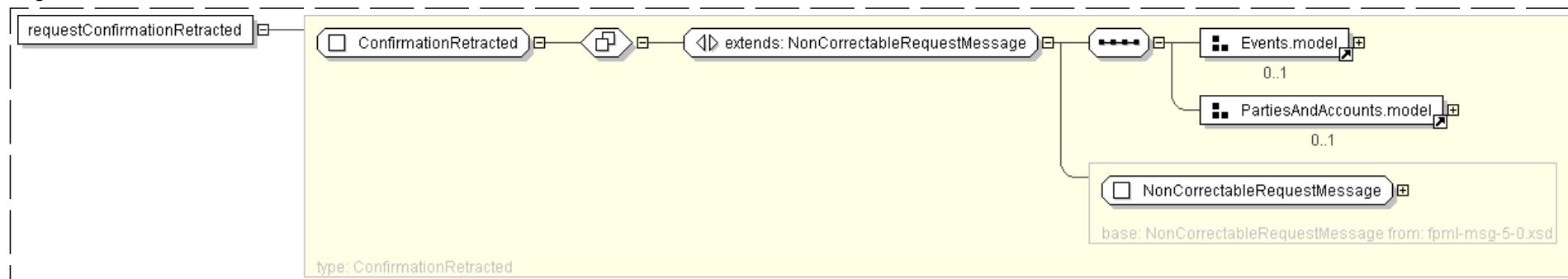
'A legal entity or a subdivision of a legal entity.', 'Parties can perform multiple roles in
a trade lifecycle. For example, the principal parties obligated to make payments from time
to time during the term of the trade, but may include other parties involved in, or
incidental to, the trade, such as parties acting in the role of novation transferor/
transferee, broker, calculation agent, etc. In FpML roles are defined in multiple places
within a document.'

<account> Account </account> [0..*]

'Optional account information used to precisely define the origination and destination
of financial instruments.'

End Group: PartiesAndAccounts.model
</requestConfirmationRetracted>

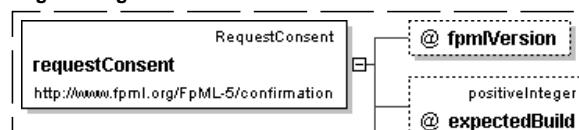
```

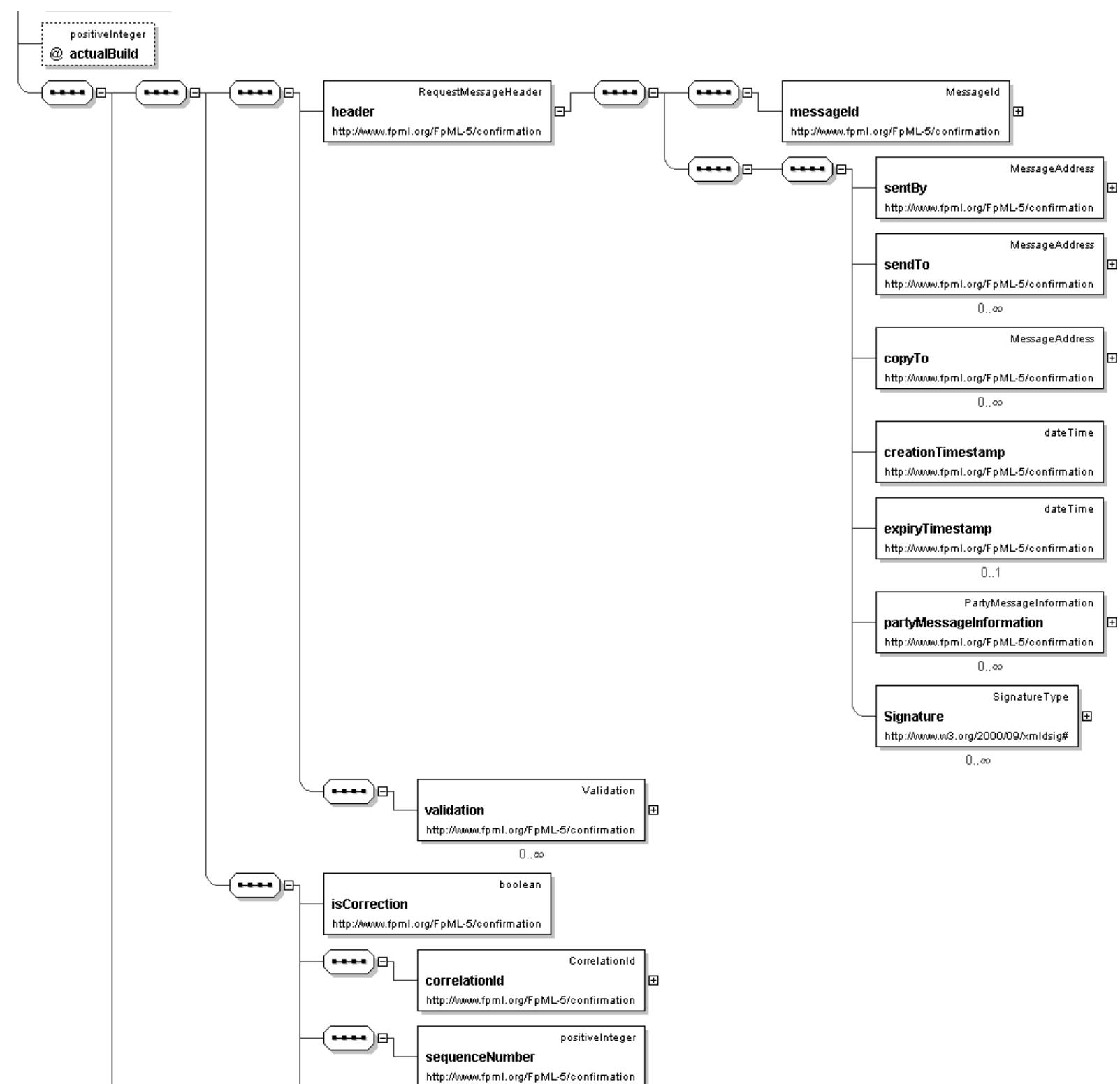
**Diagram****Schema Component Representation**

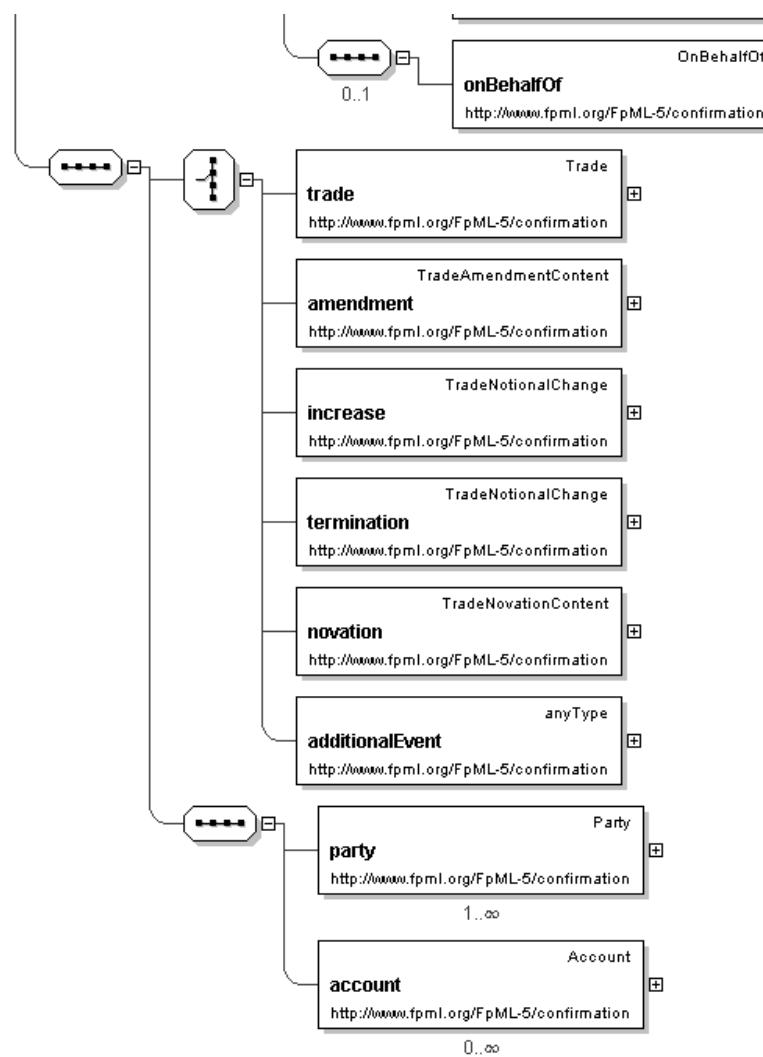
```
<xsd:element name="requestConfirmationRetracted" type="ConfirmationRetracted" />
```

[top](#)**Element: requestConsent**

Name	requestConsent
Type	<a href="#">RequestConsent</a>
Nillable	no
Abstract	no

**Logical Diagram**



**XML Instance Representation**

```

<requestConsent
fpmlVersion=" xsd:token (value comes from list: {'5-0'}) [1]
'Indicate which version of the FpML Schema an FpML message adheres to.'

"
expectedBuild=" xsd:positiveInteger [0..1]
'This optional attribute can be supplied by a message creator in an FpML instance to
specify which build number of the schema was used to define the message when it was generated.'

"
actualBuild="8 [0..1]
'The specific build number of this schema version. This attribute is not included in
an instance document. Instead, it is supplied by the XML parser when the document is
validated against the FpML schema and indicates the build number of the schema file. Every
time FpML publishes a change to the schema, validation rules, or examples within a version
'
```

(e.g., version 4.2) the actual build number is incremented. If no changes have been made between releases within a version (i.e. from Trial Recommendation to Recommendation) the actual build number stays the same.'

```
">
<header> RequestMessageHeader </header> [1]
<validation> Validation </validation> [0..*]
<isCorrection> xsd:boolean </isCorrection> [1]
'Indicates if this message corrects an earlier request.'

<correlationId> CorrelationId </correlationId> [1]
'A qualified identifier used to correlate between messages'

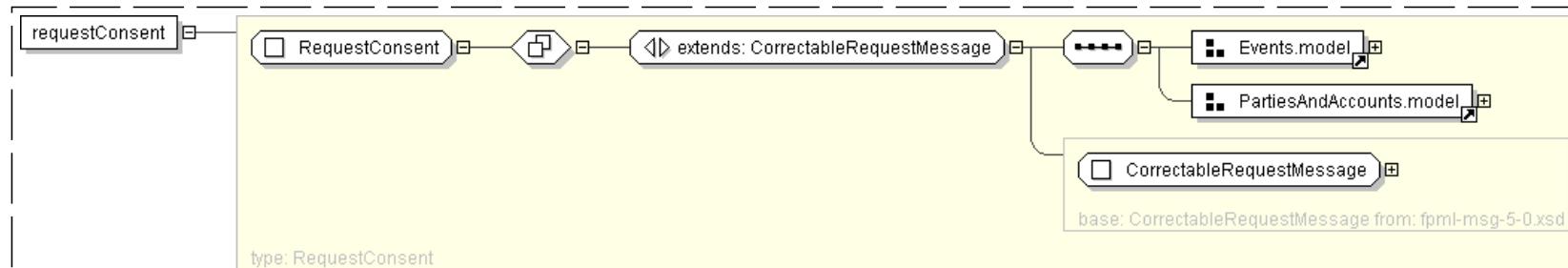
<sequenceNumber> xsd:positiveInteger </sequenceNumber> [1]
'A numeric value that can be used to order messages with the same correlation identifier from the same sender.'

Start Group: OnBehalfOf.model [0..1]
<onBehalfOf> OnBehalfOf </onBehalfOf> [1]
'Indicates which party (and accounts) a trade is being processed for.'

End Group: OnBehalfOf.model
Start Choice [1]
<trade> Trade </trade> [1]
<amendment> TradeAmendmentContent </amendment> [1]
<increase> TradeNotionalChange </increase> [1]
<termination> TradeNotionalChange </termination> [1]
<novation> TradeNovationContent </novation> [1]
<additionalEvent> ... </additionalEvent> [1]
End Choice
<party> Party </party> [1..*]
'A legal entity or a subdivision of a legal entity.' , 'Parties can perform multiple roles in a trade lifecycle. For example, the principal parties obligated to make payments from time to time during the term of the trade, but may include other parties involved in, or incidental to, the trade, such as parties acting in the role of novation transferor/transferee, broker, calculation agent, etc. In FpML roles are defined in multiple places within a document.'

<account> Account </account> [0..*]
'Optional account information used to precisely define the origination and destination of financial instruments.'

</requestConsent>
```

**Diagram**

## Schema Component Representation

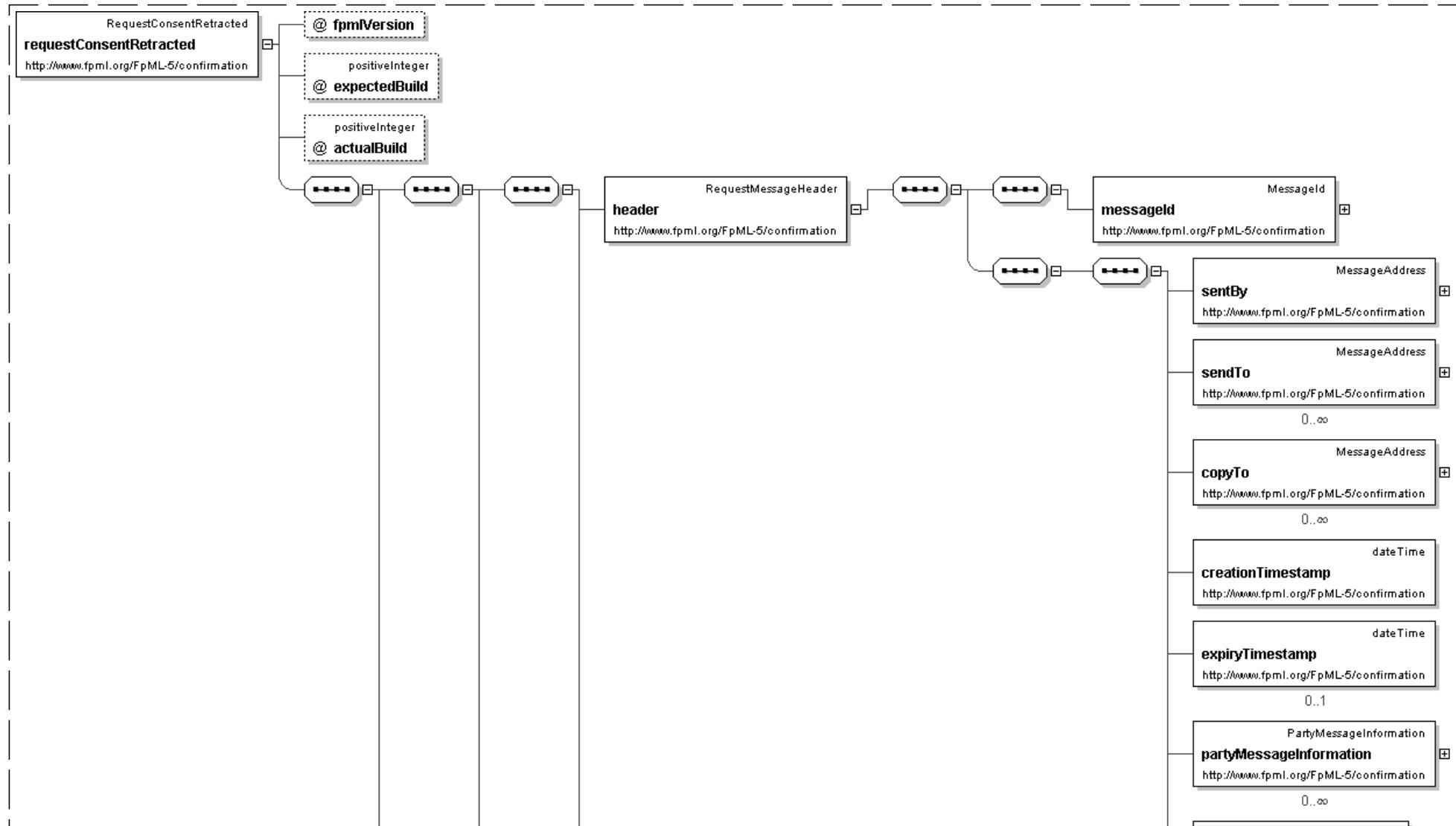
```
<xsd:element name="requestConsent" type=" RequestConsent " />
```

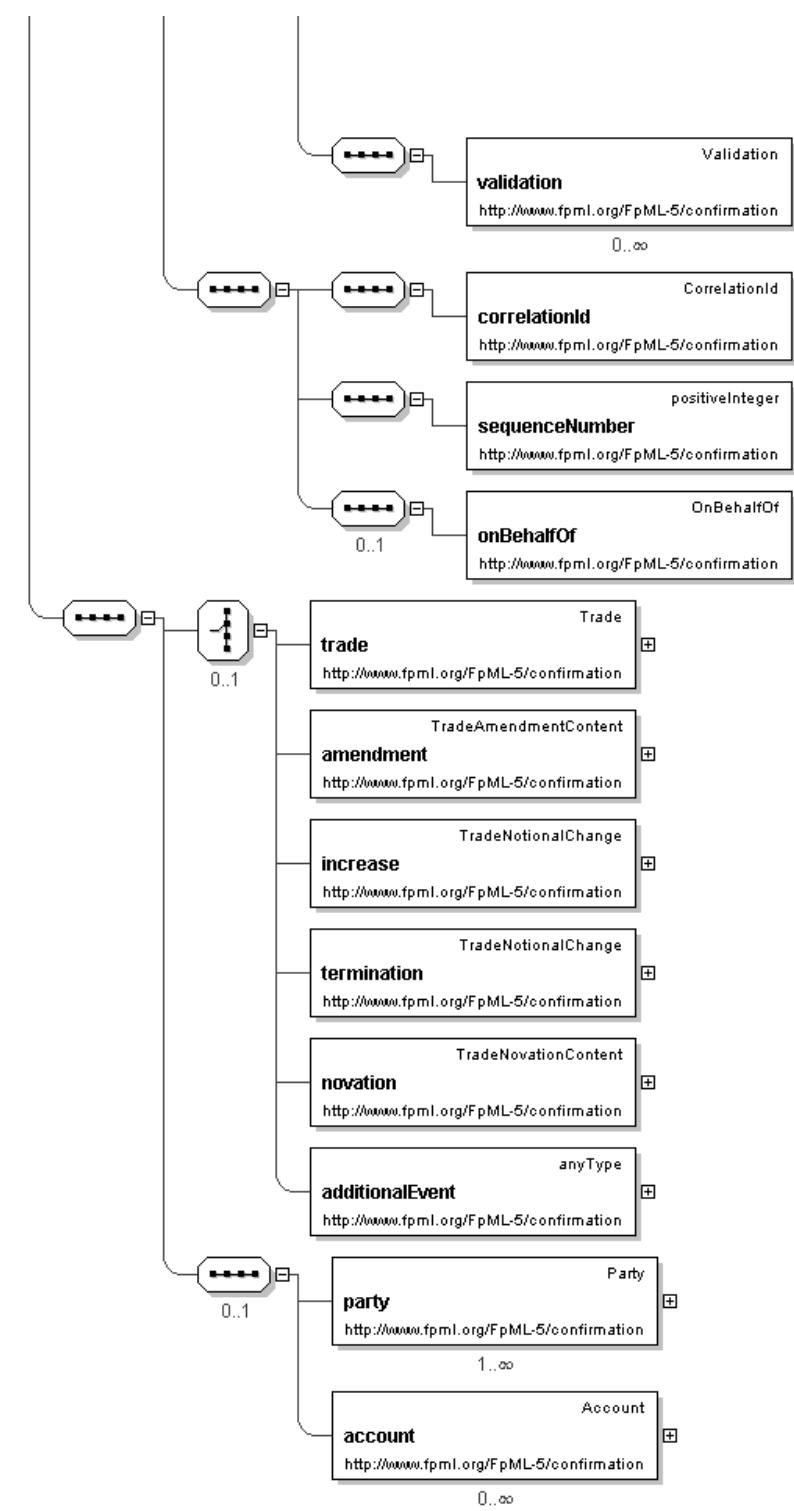
[top](#)

## Element: requestConsentRetracted

Name	requestConsentRetracted
Type	<a href="#">RequestConsentRetracted</a>
Nillable	no
Abstract	no

## Logical Diagram





**XML Instance Representation**

```

<requestConsentRetracted
  fpmlVersion=" xsd:token (value comes from list: {'5-0'}) [1]
  'Indicate which version of the FpML Schema an FpML message adheres to.'


  "
  expectedBuild=" xsd:positiveInteger [0..1]
  'This optional attribute can be supplied by a message creator in an FpML instance to
  specify which build number of the schema was used to define the message when it was generated.'


  "
  actualBuild="8 [0..1]
  'The specific build number of this schema version. This attribute is not included in
  an instance document. Instead, it is supplied by the XML parser when the document is
  validated against the FpML schema and indicates the build number of the schema file. Every
  time FpML publishes a change to the schema, validation rules, or examples within a version
  (e.g., version 4.2) the actual build number is incremented. If no changes have been
  made between releases within a version (i.e. from Trial Recommendation to Recommendation)
  the actual build number stays the same.'


  ">
    <header> RequestMessageHeader </header> [1]
    <validation> Validation </validation> [0..*]
    <correlationId> CorrelationId </correlationId> [1]
    'A qualified identifier used to correlate between messages'


    <sequenceNumber> xsd:positiveInteger </sequenceNumber> [1]
    'A numeric value that can be used to order messages with the same correlation identifier
    from the same sender.'

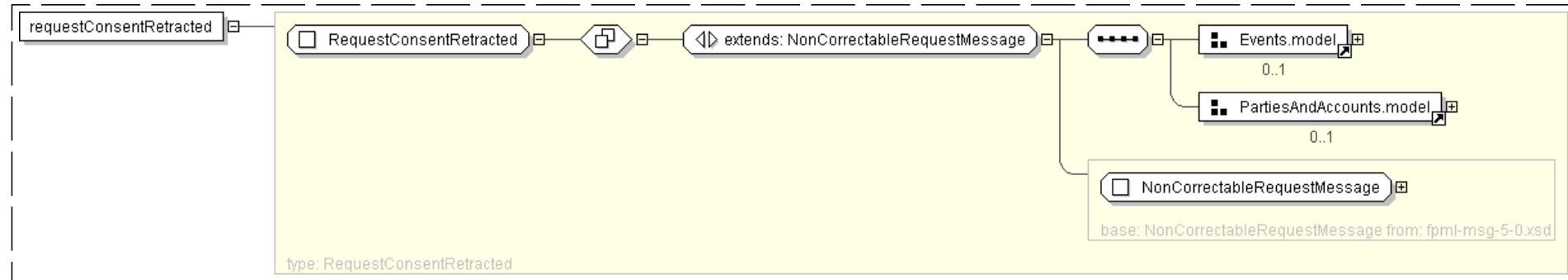

Start Group: OnBehalfOf.model [0..1]
  <onBehalfOf> OnBehalfOf </onBehalfOf> [1]
  'Indicates which party (and accounts) a trade is being processed for.'


End Group: OnBehalfOf.model
Start Group: Events.model [0..1]
Start Choice [1]
  <trade> Trade </trade> [1]
  <amendment> TradeAmendmentContent </amendment> [1]
  <increase> TradeNotionalChange </increase> [1]
  <termination> TradeNotionalChange </termination> [1]
  <novation> TradeNovationContent </novation> [1]
  <additionalEvent> ... </additionalEvent> [1]
End Choice
End Group: Events.model
Start Group: PartiesAndAccounts.model [0..1]
  <party> Party </party> [1..*]
  'A legal entity or a subdivision of a legal entity.', 'Parties can perform multiple roles in
  a trade lifecycle. For example, the principal parties obligated to make payments from time
  to time during the term of the trade, but may include other parties involved in, or
  incidental to, the trade, such as parties acting in the role of novation transferor/
  transferee, broker, calculation agent, etc. In FpML roles are defined in multiple places
  within a document.'


  <account> Account </account> [0..*]
  'Optional account information used to precisely define the origination and destination
  of financial instruments.'

```

```
| End Group: PartiesAndAccounts.model
</requestConsentRetracted>
```

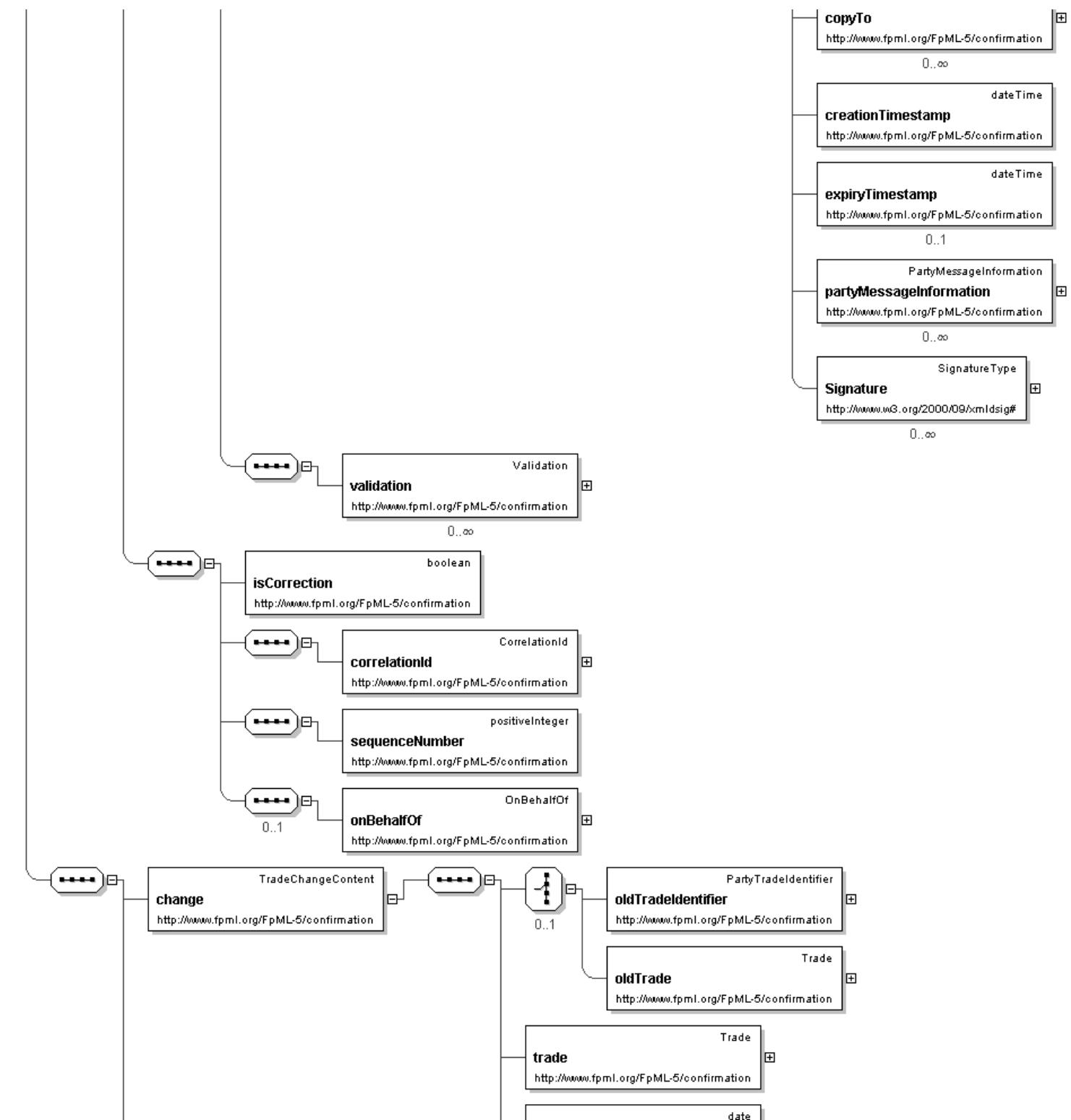
**Diagram****Schema Component Representation**

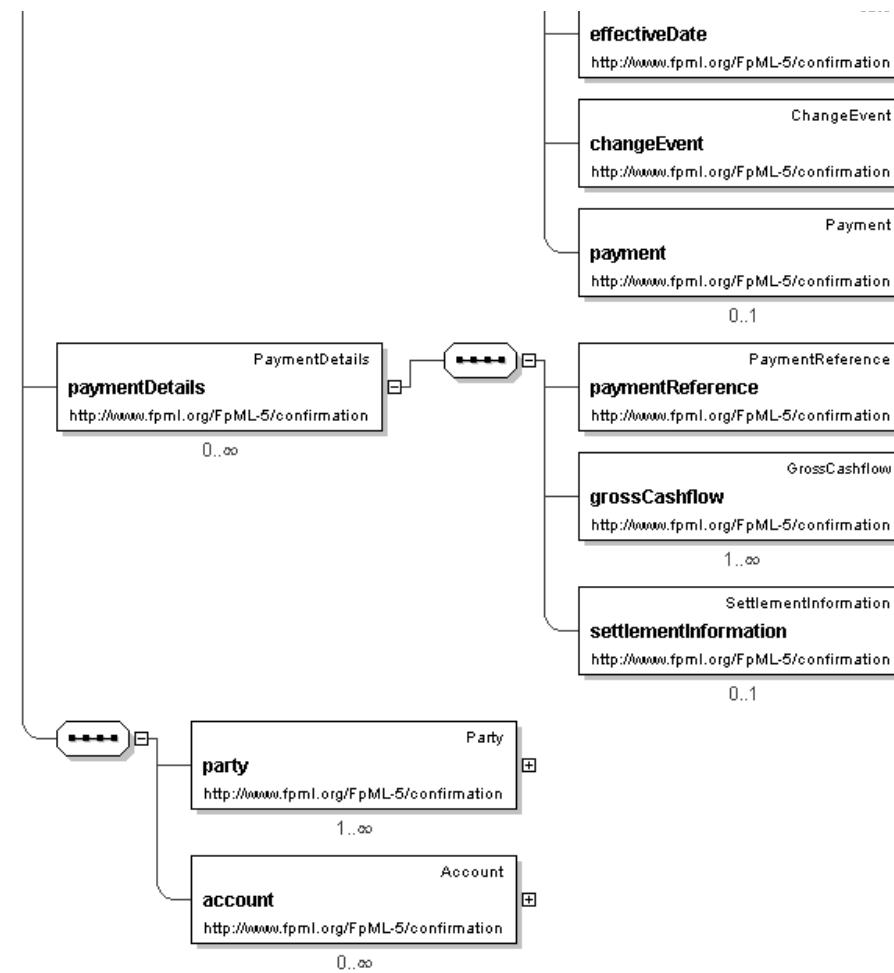
```
<xsd:element name="requestConsentRetracted" type=" RequestConsentRetracted " />
```

[top](#)**Element: tradeChangeAdvice**

<b>Name</b>	tradeChangeAdvice
<b>Type</b>	<a href="#">TradeChangeAdvice</a>
<b>Nillable</b>	no
<b>Abstract</b>	no

**Logical Diagram**



**XML Instance Representation**

```

<tradeChangeAdvice
  fpmlVersion=" xsd:token (value comes from list: {'5-0'}) [1]
  'Indicate which version of the FpML Schema an FpML message adheres to.'
  "
  expectedBuild=" xsd:positiveInteger [0..1]
  'This optional attribute can be supplied by a message creator in an FpML instance to
  specify which build number of the schema was used to define the message when it was generated.'
  "
  actualBuild="8 [0..1]
  'The specific build number of this schema version. This attribute is not included in
  an instance document. Instead, it is supplied by the XML parser when the document is
  validated against the FpML schema and indicates the build number of the schema file. Every
  time FpML publishes a change to the schema, validation rules, or examples within a version
  (e.g., version 4.2) the actual build number is incremented. If no changes have been
  made between releases within a version (i.e. from Trial Recommendation to Recommendation)
  the actual build number stays the same.'

```

```

">>
<header> RequestMessageHeader </header> [1]
<validation> Validation </validation> [0..*]
<isCorrection> xsd:boolean </isCorrection> [1]
'Indicates if this message corrects an earlier request.'

<correlationId> CorrelationId </correlationId> [1]
'A qualified identifier used to correlate between messages'

<sequenceNumber> xsd:positiveInteger </sequenceNumber> [1]
'A numeric value that can be used to order messages with the same correlation identifier
from the same sender.'

Start Group: OnBehalfOf.model [0..1]
<onBehalfOf> OnBehalfOf </onBehalfOf> [1]
'Indicates which party (and accounts) a trade is being processed for.'

End Group: OnBehalfOf.model
<change> TradeChangeContent </change> [1]
'Describes the details of the change.'

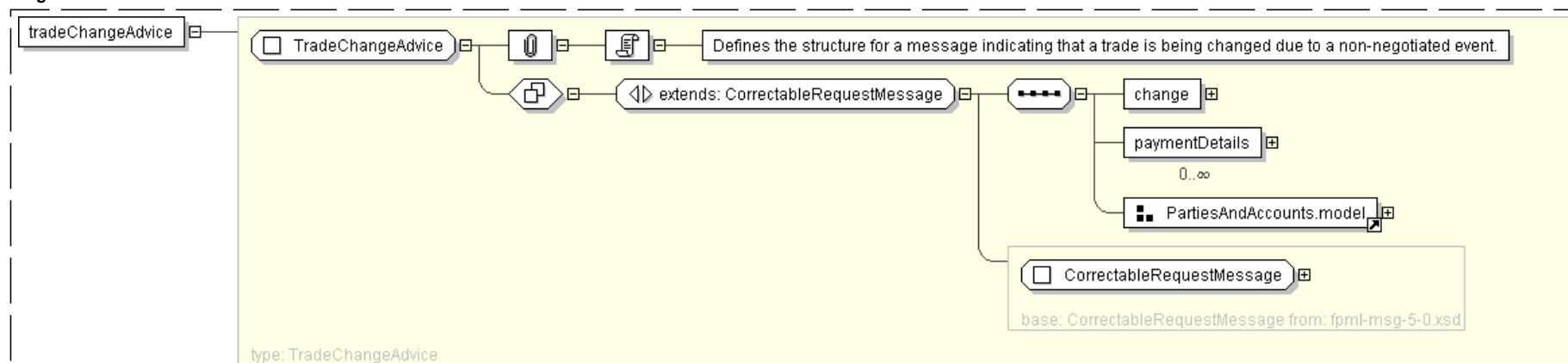
<paymentDetails> PaymentDetails </paymentDetails> [0..*]
'Details of the payments, like amount breakdowns, settlement information.'

<party> Party </party> [1..*]
'A legal entity or a subdivision of a legal entity.', 'Parties can perform multiple roles in
a trade lifecycle. For example, the principal parties obligated to make payments from time
to time during the term of the trade, but may include other parties involved in, or
incidental to, the trade, such as parties acting in the role of novation transferor/
transferee, broker, calculation agent, etc. In FpML roles are defined in multiple places
within a document..'

<account> Account </account> [0..*]
'Optional account information used to precisely define the origination and destination
of financial instruments.'

</tradeChangeAdvice>

```

**Diagram**

## Schema Component Representation

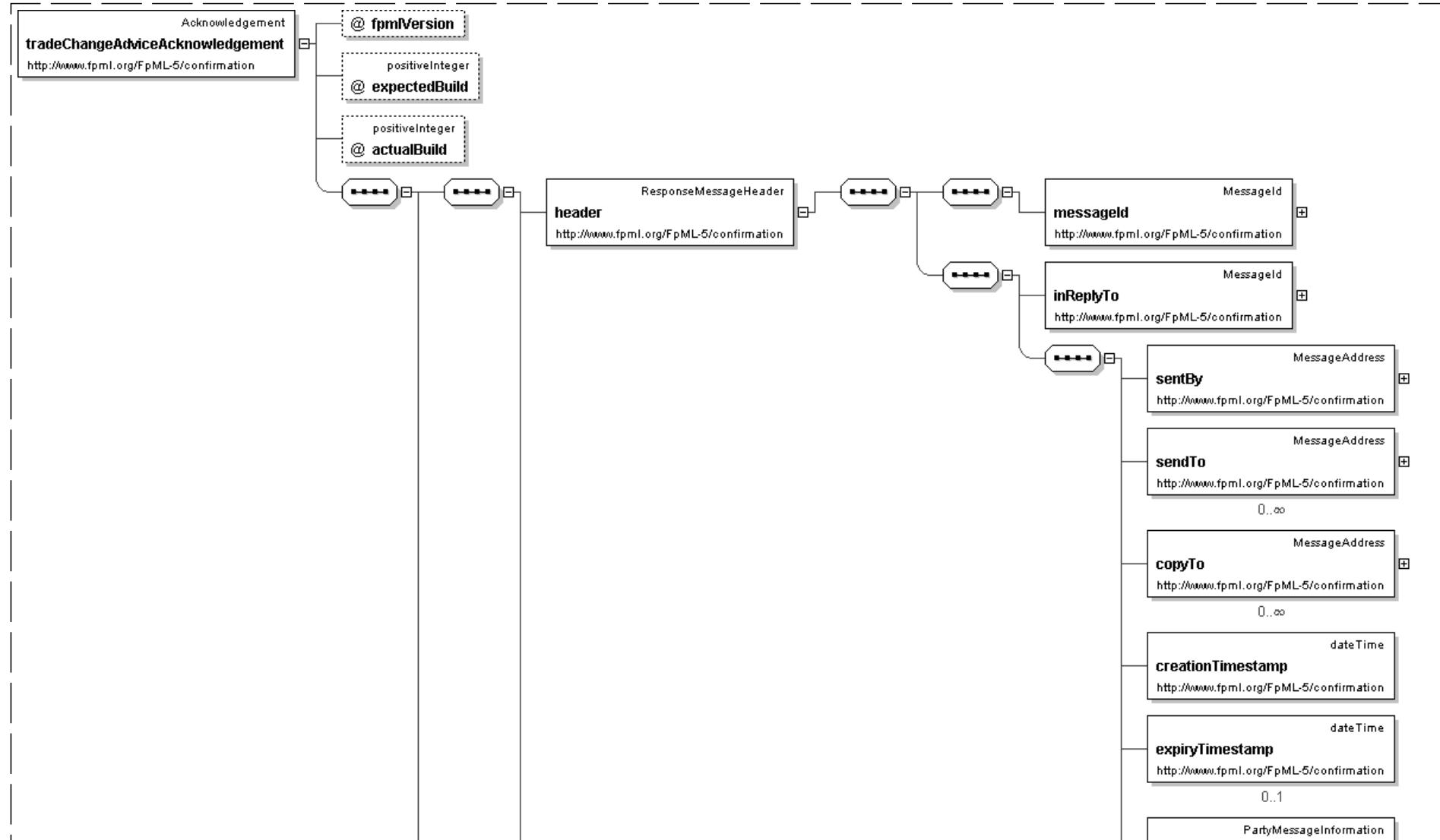
```
<xsd:element name="tradeChangeAdvice" type="TradeChangeAdvice" />
```

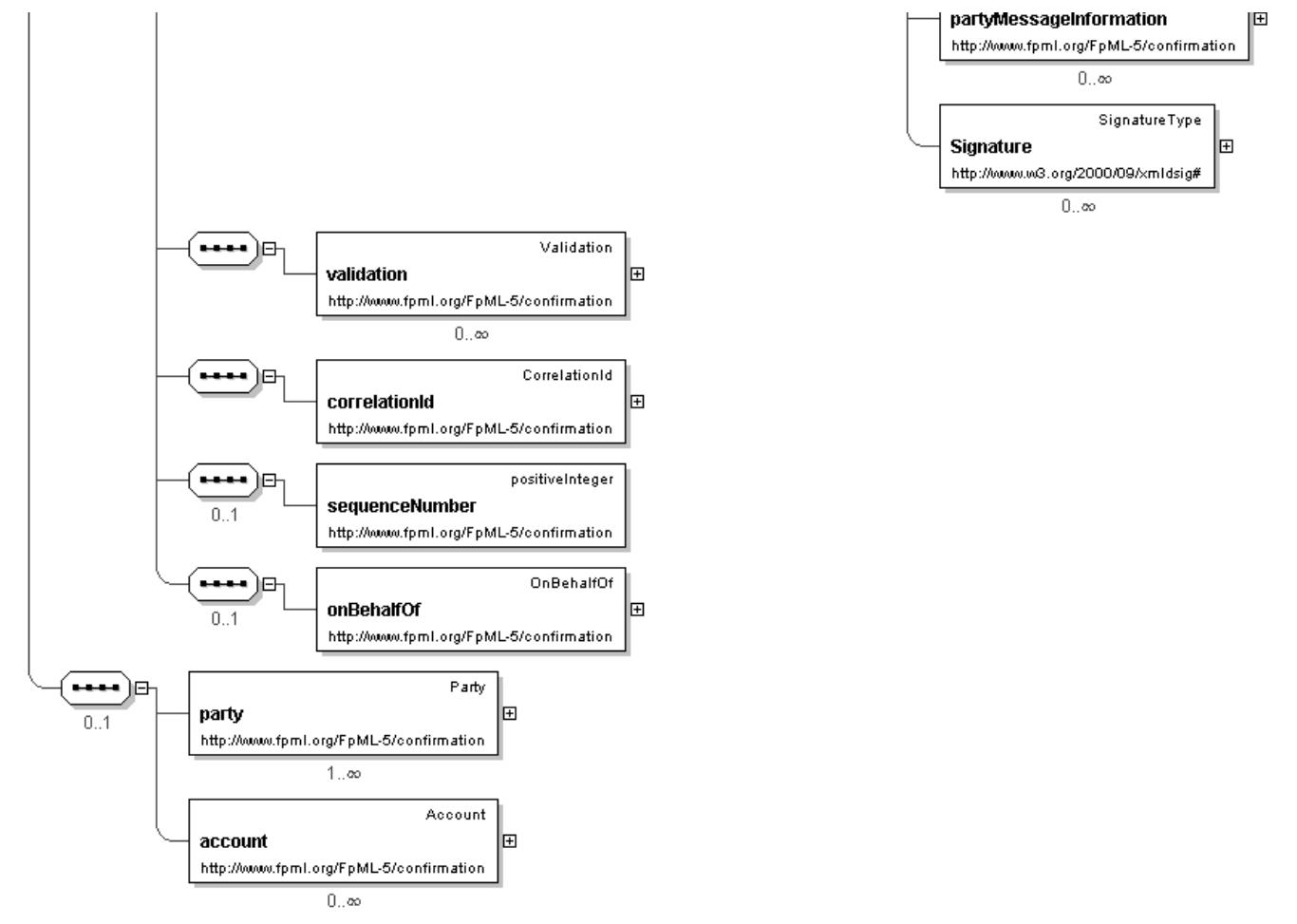
[top](#)

## Element: tradeChangeAdviceAcknowledgement

Name	tradeChangeAdviceAcknowledgement
Type	<a href="#">Acknowledgement</a>
Nillable	no
Abstract	no

## Logical Diagram



**XML Instance Representation**

```

<tradeChangeAdviceAcknowledgement
  fpmlVersion=" xsd:token (value comes from list: {'5-0'}) [1]
  'Indicate which version of the FpML Schema an FpML message adheres to.'

  "
  expectedBuild=" xsd:positiveInteger [0..1]
  'This optional attribute can be supplied by a message creator in an FpML instance to
  specify which build number of the schema was used to define the message when it was generated.'

  "
  actualBuild="8 [0..1]
  'The specific build number of this schema version. This attribute is not included in
  an instance document. Instead, it is supplied by the XML parser when the document is
  validated against the FpML schema and indicates the build number of the schema file. Every
  time FpML publishes a change to the schema, validation rules, or examples within a version
  (e.g., version 4.2) the actual build number is incremented. If no changes have been
  made between releases within a version (i.e. from Trial Recommendation to Recommendation)
  the actual build number stays the same.'
  
```

```

">>
<header> ResponseMessageHeader </header> [1]
<validation> Validation </validation> [0..*]
<correlationId> CorrelationId </correlationId> [1]
'A qualified identifier used to correlate between messages'

Start Group: Sequence.model [0..1]
<sequenceNumber> xsd:positiveInteger </sequenceNumber> [1]
'A numeric value that can be used to order messages with the same correlation identifier
from the same sender.'

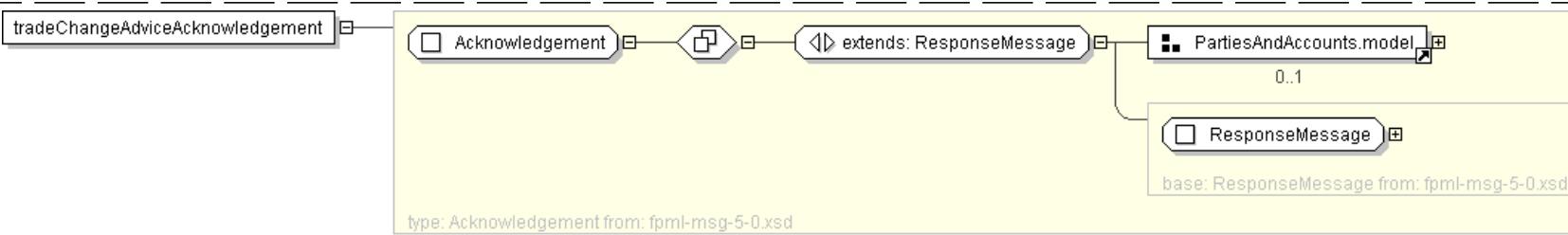
End Group: Sequence.model
Start Group: OnBehalfOf.model [0..1]
<onBehalfOf> OnBehalfOf </onBehalfOf> [1]
'Indicates which party (and accounts) a trade is being processed for.'

End Group: OnBehalfOf.model
Start Group: PartiesAndAccounts.model [0..1]
<party> Party </party> [1..*]
'A legal entity or a subdivision of a legal entity.', 'Parties can perform multiple roles in
a trade lifecycle. For example, the principal parties obligated to make payments from time
to time during the term of the trade, but may include other parties involved in, or
incidental to, the trade, such as parties acting in the role of novation transferor/
transferee, broker, calculation agent, etc. In FpML roles are defined in multiple places
within a document.'

<account> Account </account> [0..*]
'Optional account information used to precisely define the origination and destination
of financial instruments.'

End Group: PartiesAndAccounts.model
</tradeChangeAdviceAcknowledgement>

```

**Diagram****Schema Component Representation**

```
<xsd:element name="tradeChangeAdviceAcknowledgement" type=" Acknowledgement " />
```

[top](#)**Element: tradeChangeAdviceException**

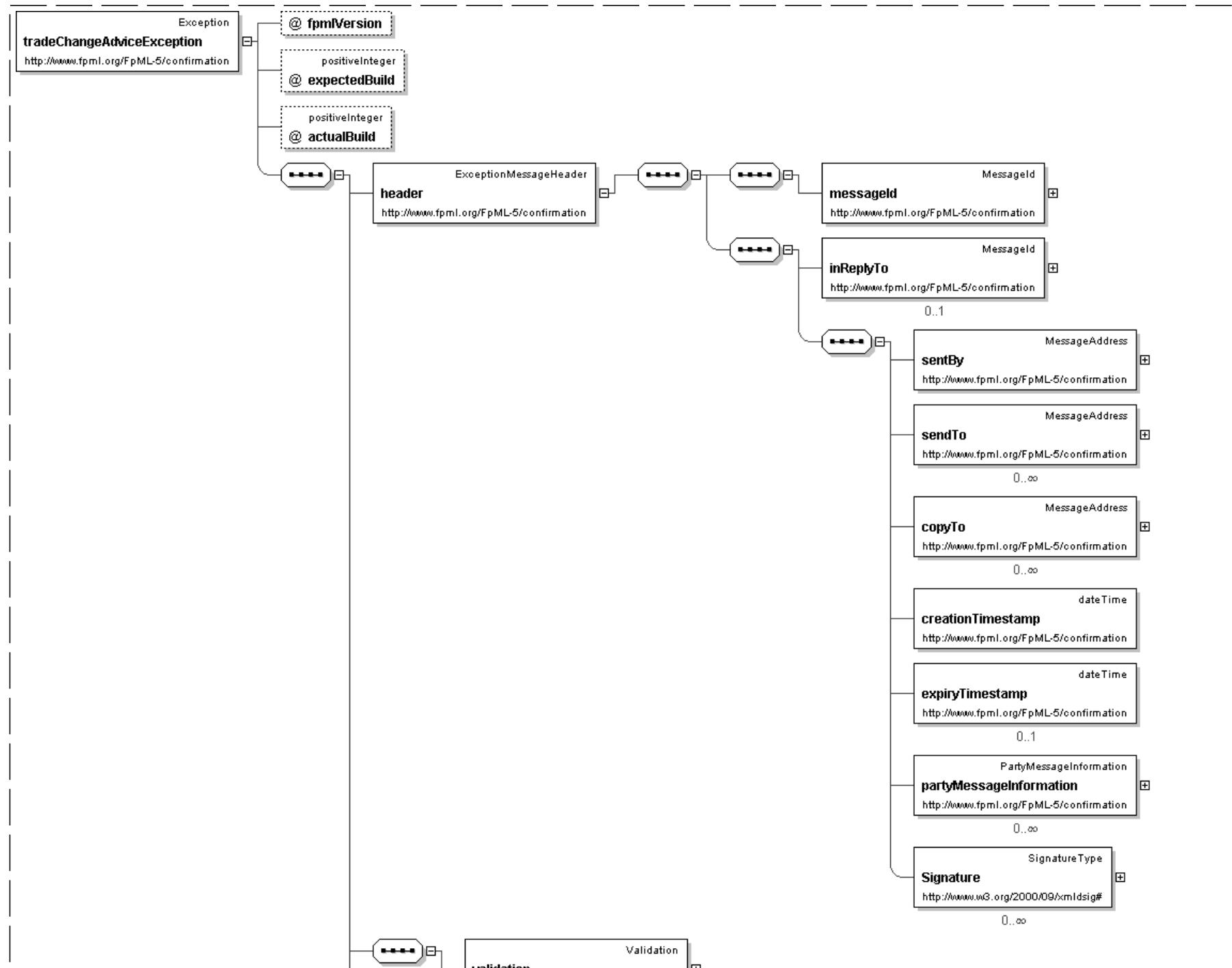
Name	tradeChangeAdviceException
Type	Exception

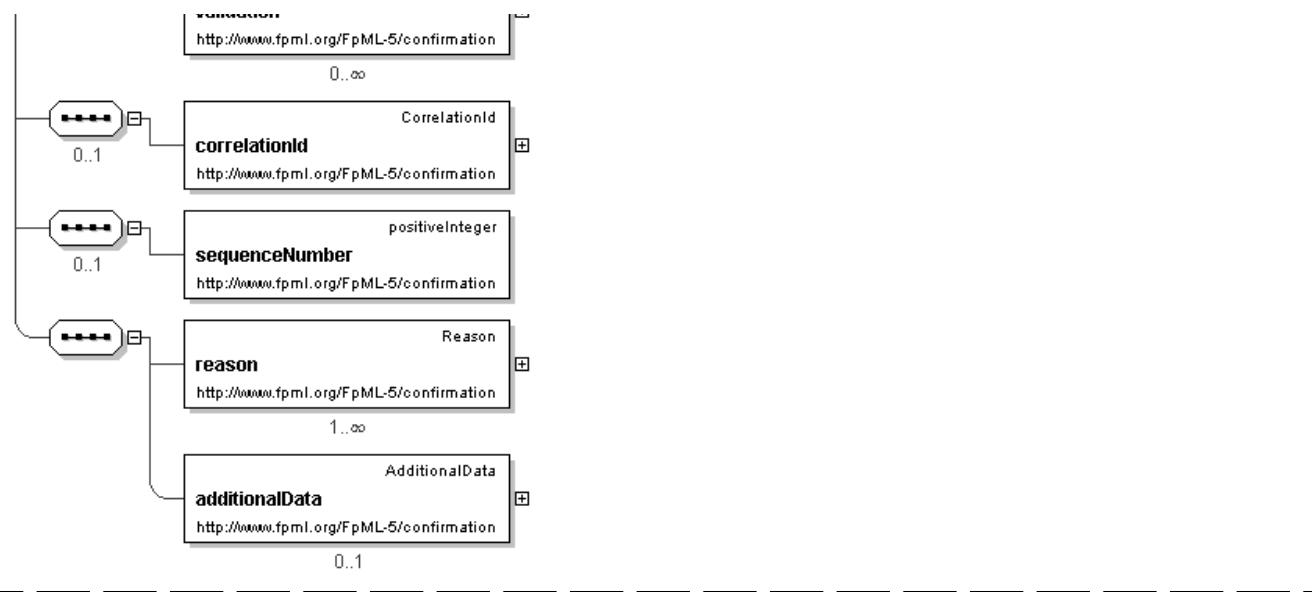
**Nillable**

no

**Abstract**

no

**Logical Diagram**

**XML Instance Representation**

```

<tradeChangeAdviceException
fpmlVersion=" xsd:token (value comes from list: {'5-0'}) [1]
'Indicate which version of the FpML Schema an FpML message adheres to.'

"
expectedBuild=" xsd:positiveInteger [0..1]
'This optional attribute can be supplied by a message creator in an FpML instance to
specify which build number of the schema was used to define the message when it was generated.'

"
actualBuild="8 [0..1]
'The specific build number of this schema version. This attribute is not included in
an instance document. Instead, it is supplied by the XML parser when the document is
validated against the FpML schema and indicates the build number of the schema file. Every
time FpML publishes a change to the schema, validation rules, or examples within a version
(e.g., version 4.2) the actual build number is incremented. If no changes have been
made between releases within a version (i.e. from Trial Recommendation to Recommendation)
the actual build number stays the same.'

">
<header> ExceptionMessageHeader </header> [1]
<validation> Validation </validation> [0..*]
Start Group: Correlation.model [0..1]
    <correlationId> CorrelationId </correlationId> [1]
    'A qualified identifier used to correlate between messages'

End Group: Correlation.model
Start Group: Sequence.model [0..1]
    <sequenceNumber> xsd:positiveInteger </sequenceNumber> [1]
    'A numeric value that can be used to order messages with the same correlation identifier
from the same sender.'

End Group: Sequence.model
    <reason> Reason </reason> [1..*]

```

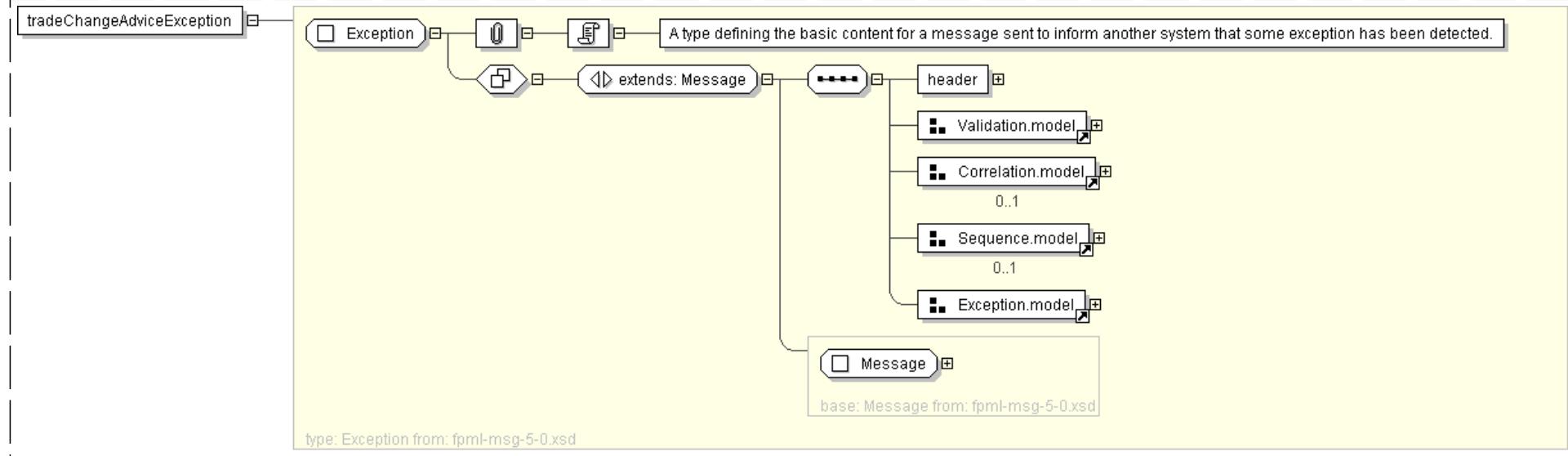
'An instance of the Reason type used to record the nature of any errors associated with a message.'

<additionalData> AdditionalData </additionalData> [0..1]

'Any string of additional data that may help the message processor, for example in a rejection message this might contain a code value or the text of the original request (within a CDATA section).'

</tradeChangeAdviceException>

#### Diagram



#### Schema Component Representation

```
<xsd:element name="tradeChangeAdviceException" type="Exception" />
```

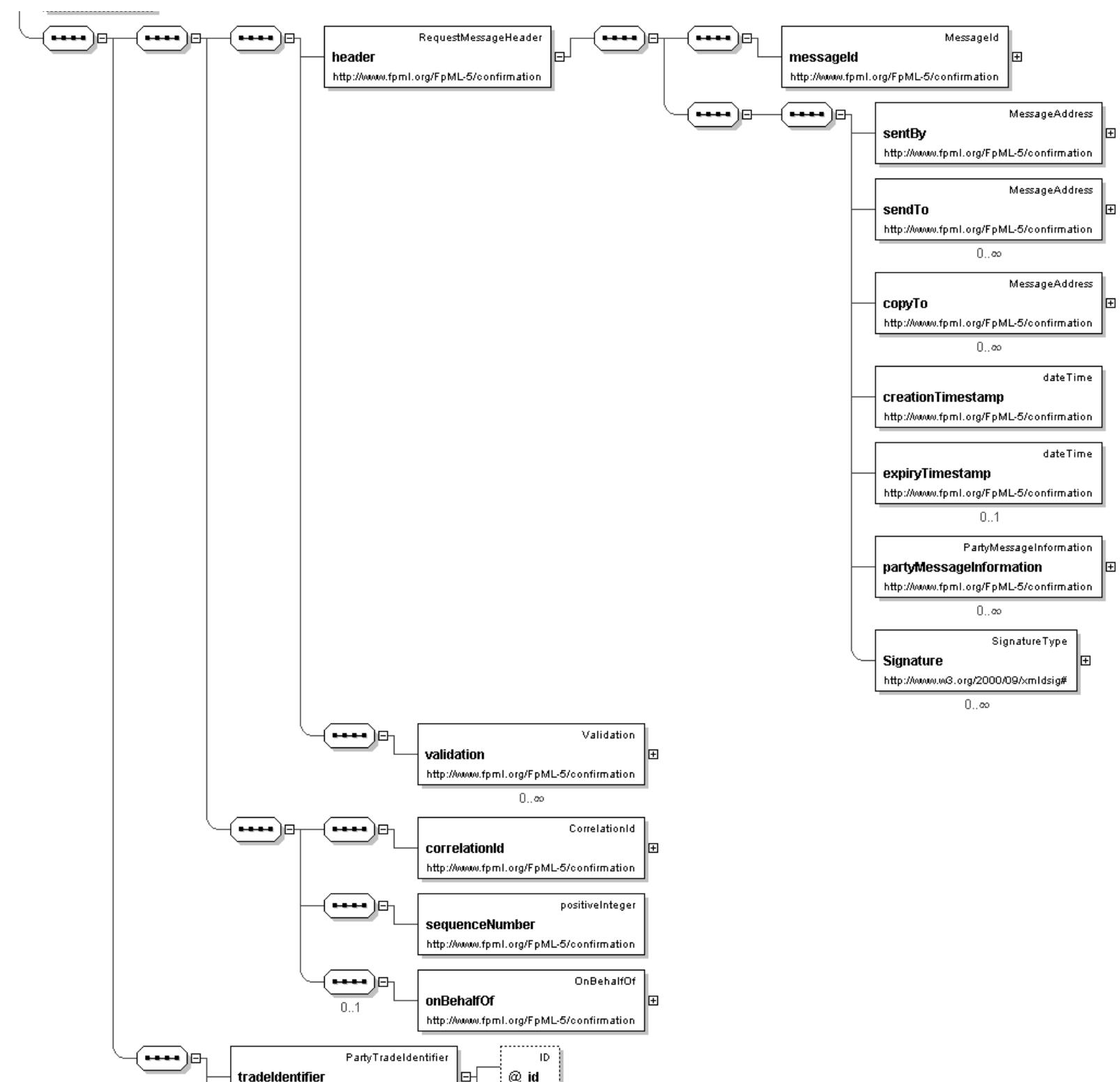
[top](#)

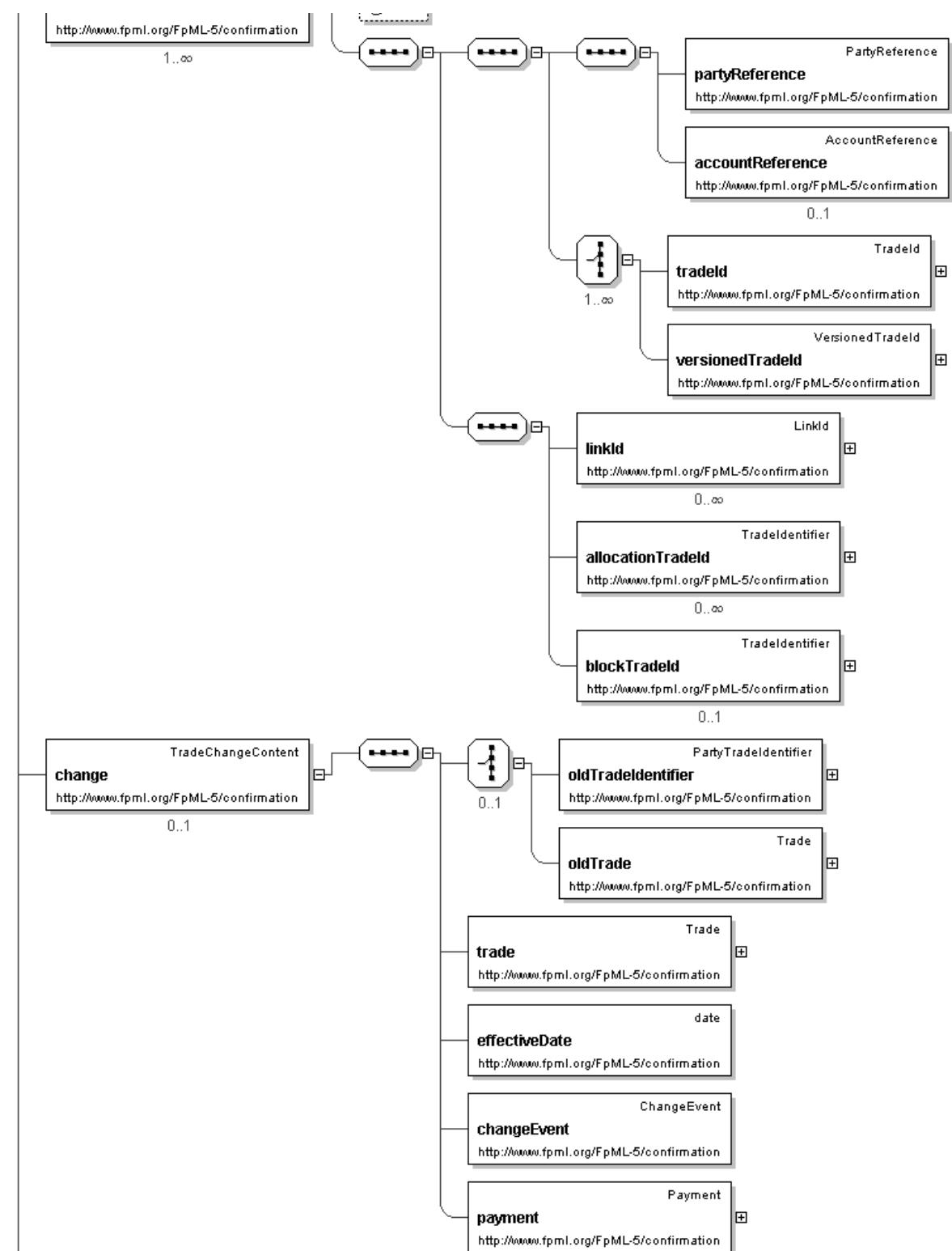
#### Element: tradeChangeAdviceRetracted

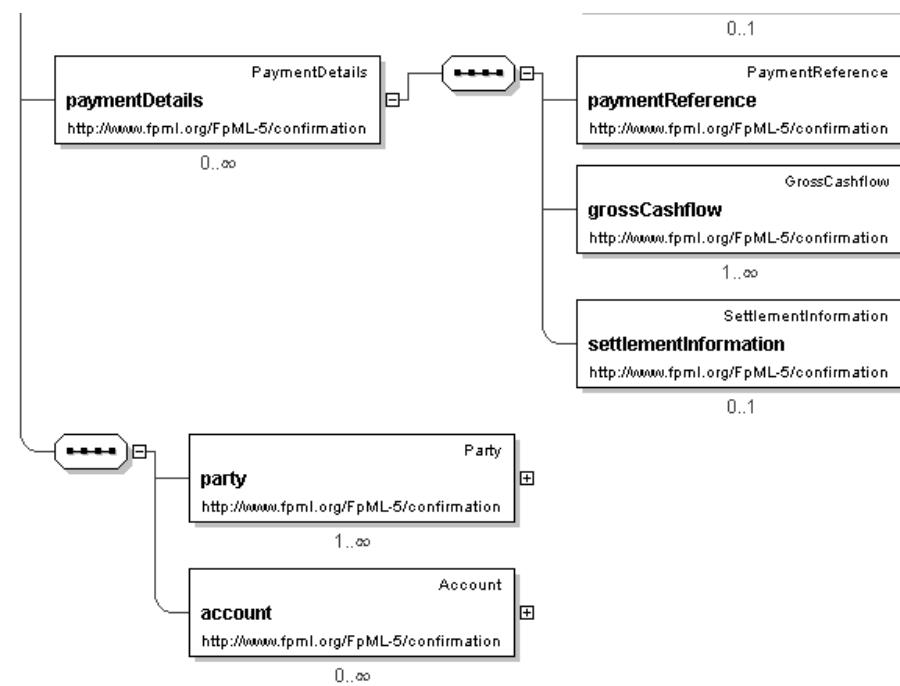
Name	tradeChangeAdviceRetracted
Type	TradeChangeAdviceRetracted
Nillable	no
Abstract	no

#### Logical Diagram







**XML Instance Representation**

```

<tradeChangeAdviceRetracted
  fpmlVersion=" xsd:token (value comes from list: {'5-0'})" [1]
  'Indicate which version of the FpML Schema an FpML message adheres to.'

  "
  expectedBuild=" xsd:positiveInteger [0..1]
  'This optional attribute can be supplied by a message creator in an FpML instance to
  specify which build number of the schema was used to define the message when it was generated.'

  "
  actualBuild="8 [0..1]
  'The specific build number of this schema version. This attribute is not included in
  an instance document. Instead, it is supplied by the XML parser when the document is
  validated against the FpML schema and indicates the build number of the schema file. Every
  time FpML publishes a change to the schema, validation rules, or examples within a version
  (e.g., version 4.2) the actual build number is incremented. If no changes have been
  made between releases within a version (i.e. from Trial Recommendation to Recommendation)
  the actual build number stays the same.'

  ">
  <header> RequestMessageHeader </header> [1]
  <validation> Validation </validation> [0..*]
  <correlationId> CorrelationId </correlationId> [1]
  'A qualified identifier used to correlate between messages'

  <sequenceNumber> xsd:positiveInteger </sequenceNumber> [1]
  'A numeric value that can be used to order messages with the same correlation identifier
  from the same sender.'
  
```

```
Start Group: OnBehalfOf.model [0..1]
  <onBehalfOf> OnBehalfOf </onBehalfOf> [1]
    'Indicates which party (and accounts) a trade is being processed for.'

End Group: OnBehalfOf.model
  <tradeIdentifier> PartyTradeIdentifier </tradeIdentifier> [1..*]
    'The qualified identifiers of the subject trade.'

  <change> TradeChangeContent </change> [0..1]
    'Describes the details of the change being retracted.'

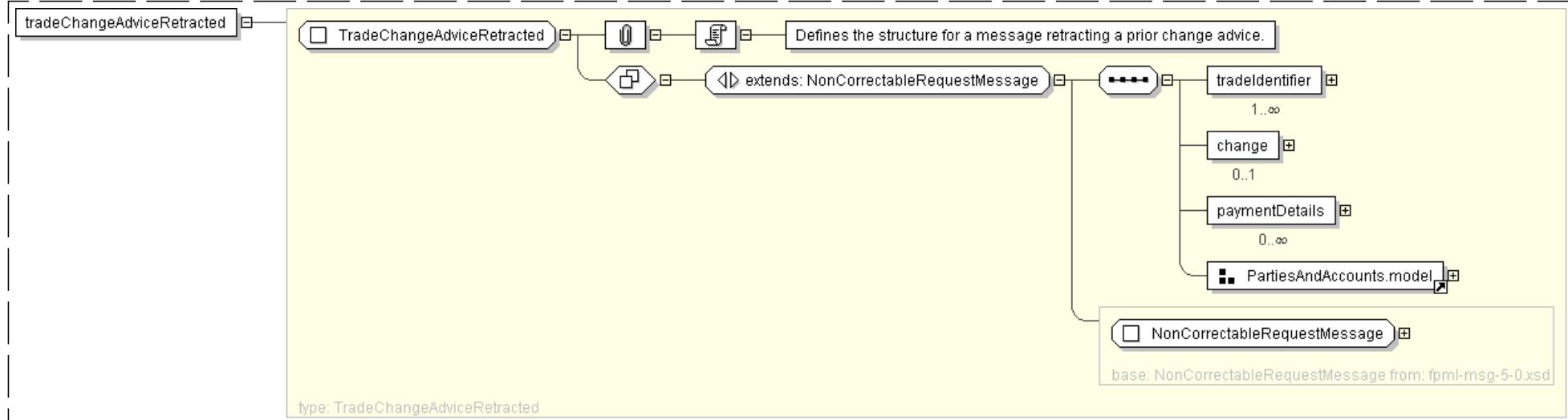
  <paymentDetails> PaymentDetails </paymentDetails> [0..*]
    'Details of the payments, like amount breakdowns, settlement information.'

  <party> Party </party> [1..*]
    'A legal entity or a subdivision of a legal entity.', 'Parties can perform multiple roles in a trade lifecycle. For example, the principal parties obligated to make payments from time to time during the term of the trade, but may include other parties involved in, or incidental to, the trade, such as parties acting in the role of novation transferor/transferee, broker, calculation agent, etc. In FpML roles are defined in multiple places within a document.'

  <account> Account </account> [0..*]
    'Optional account information used to precisely define the origination and destination of financial instruments.'

</tradeChangeAdviceRetracted>
```

## Diagram



## Schema Component Representation

```
<xsd:element name="tradeChangeAdviceRetracted" type=" TradeChangeAdviceRetracted " />
```

## Global Definitions

### Complex Type: AllocationApproved

Super-types:	<a href="#">ResponseMessage</a> < <b>AllocationApproved</b> (by extension)
Sub-types:	None

Name	AllocationApproved
Used by (from the same schema document)	Element <a href="#">allocationApproved</a>
Abstract	no

#### XML Instance Representation

```
<...
fpmVersion=" xsd:token (value comes from list: {'5-0'}) [1]
'Indicate which version of the FpML Schema an FpML message adheres to.'

"
expectedBuild=" xsd:positiveInteger [0..1]
'This optional attribute can be supplied by a message creator in an FpML instance to
specify which build number of the schema was used to define the message when it was generated.'

"
actualBuild="8 [0..1]
'The specific build number of this schema version. This attribute is not included in
an instance document. Instead, it is supplied by the XML parser when the document is
validated against the FpML schema and indicates the build number of the schema file. Every
time FpML publishes a change to the schema, validation rules, or examples within a version
(e.g., version 4.2) the actual build number is incremented. If no changes have been
made between releases within a version (i.e. from Trial Recommendation to Recommendation)
the actual build number stays the same.'

">
<header> ResponseMessageHeader </header> [1]
<validation> Validation </validation> [0..*]
<correlationId> CorrelationId </correlationId> [1]
'A qualified identifier used to correlate between messages'

Start Group: Sequence.model [0..1]
<sequenceNumber> xsd:positiveInteger </sequenceNumber> [1]
'A numeric value that can be used to order messages with the same correlation identifier
from the same sender.'

End Group: Sequence.model
Start Group: OnBehalfOf.model [0..1]
<onBehalfOf> OnBehalfOf </onBehalfOf> [1]
'Indicates which party (and accounts) a trade is being processed for.'

End Group: OnBehalfOf.model
<blockTradeIdentifier> TradeIdentifier </blockTradeIdentifier> [0..1]
<allocations> Allocations </allocations> [0..1]
Start Group: PartiesAndAccounts.model [0..1]
<party> Party </party> [1..*]
'A legal entity or a subdivision of a legal entity.', 'Parties can perform multiple roles in
a trade lifecycle. For example, the principal parties obligated to make payments from time
to time during the term of the trade, but may include other parties involved in, or
incidental to, the trade, such as parties acting in the role of novation transferor/
```

*transferee, broker, calculation agent, etc. In FpML roles are defined in multiple places within a document.'*

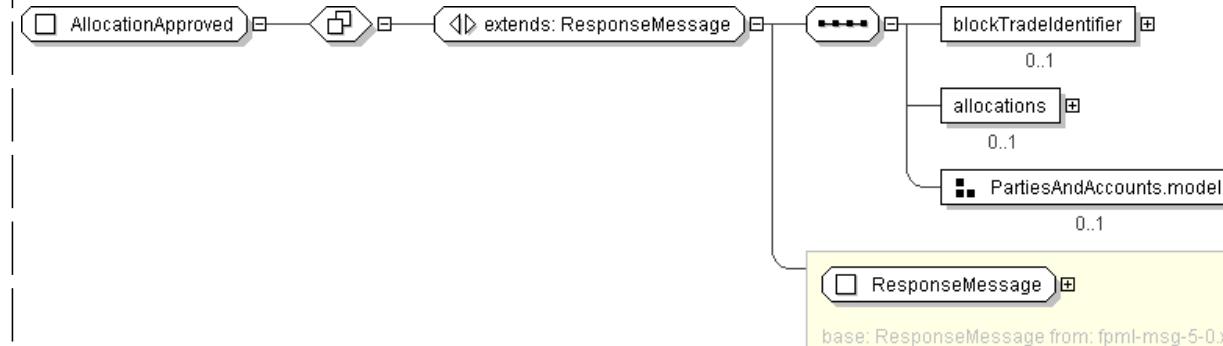
<account> [Account](#) </account> [0..\*]

'Optional account information used to precisely define the origination and destination of financial instruments.'

End Group: [PartiesAndAccounts.model](#)

</...>

#### Diagram



#### Schema Component Representation

```

<xsd:complexType name="AllocationApproved">
  <xsd:complexContent>
    <xsd:extension base="ResponseMessage">
      <xsd:sequence>
        <xsd:element name="blockTradeIdentifier" type="TradeIdentifier" minOccurs="0"/>
        <xsd:element name="allocations" type="Allocations" minOccurs="0"/>
        <xsd:group ref="PartiesAndAccounts.model" minOccurs="0"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
  
```

[top](#)

## Complex Type: [AllocationRefused](#)

Super-types:	<a href="#">ResponseMessage</a> < <b>AllocationRefused</b> (by extension)
Sub-types:	None

Name	AllocationRefused
Used by (from the same schema document)	Element <a href="#">allocationRefused</a>
Abstract	no

#### XML Instance Representation

```

<...
fpmlVersion=" xsd:token (value comes from list: {'5-0'}) [1]
'Indicate which version of the FpML Schema an FpML message adheres to.'
  
```

```

"
expectedBuild=" xsd:positiveInteger [0..1]
'This optional attribute can be supplied by a message creator in an FpML instance to
specify which build number of the schema was used to define the message when it was generated.'

"
actualBuild="8 [0..1]
'The specific build number of this schema version. This attribute is not included in
an instance document. Instead, it is supplied by the XML parser when the document is
validated against the FpML schema and indicates the build number of the schema file. Every
time FpML publishes a change to the schema, validation rules, or examples within a version
(e.g., version 4.2) the actual build number is incremented. If no changes have been
made between releases within a version (i.e. from Trial Recommendation to Recommendation)
the actual build number stays the same.'


">
<header> ResponseMessageHeader </header> [1]
<validation> Validation </validation> [0..*]
<correlationId> CorrelationId </correlationId> [1]
'A qualified identifier used to correlate between messages'

Start Group: Sequence.model [0..1]
<sequenceNumber> xsd:positiveInteger </sequenceNumber> [1]
'A numeric value that can be used to order messages with the same correlation identifier
from the same sender.'

End Group: Sequence.model
Start Group: OnBehalfOf.model [0..1]
<onBehalfOf> OnBehalfOf </onBehalfOf> [1]
'Indicates which party (and accounts) a trade is being processed for.'

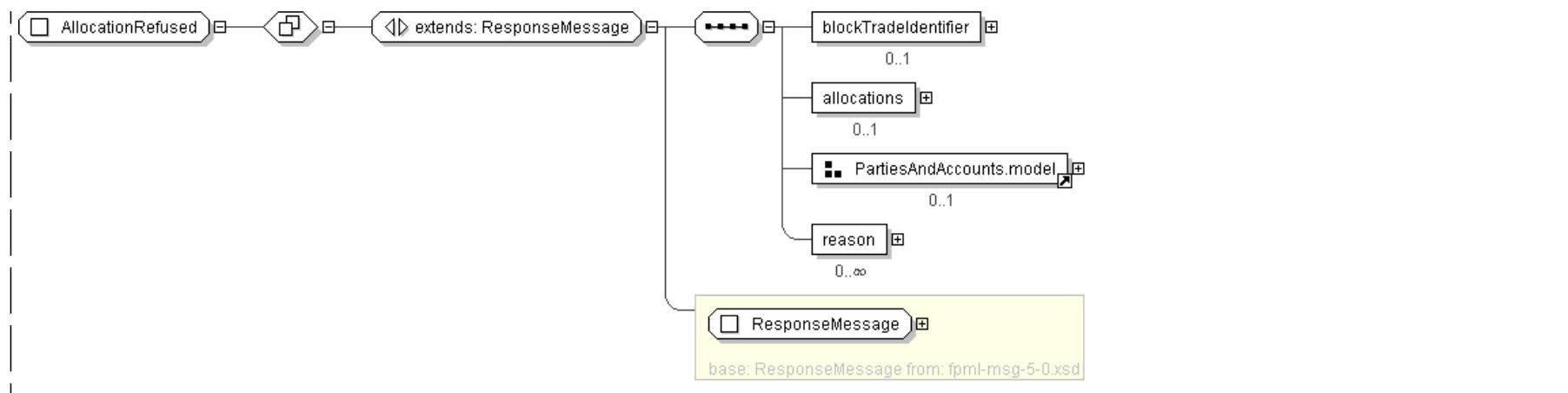
End Group: OnBehalfOf.model
<blockTradeIdentifier> TradeIdentifier </blockTradeIdentifier> [0..1]
<allocations> Allocations </allocations> [0..1]
Start Group: PartiesAndAccounts.model [0..1]
<party> Party </party> [1..*]
'A legal entity or a subdivision of a legal entity.', 'Parties can perform multiple roles in
a trade lifecycle. For example, the principal parties obligated to make payments from time
to time during the term of the trade, but may include other parties involved in, or
incidental to, the trade, such as parties acting in the role of novation transferor/
transferee, broker, calculation agent, etc. In FpML roles are defined in multiple places
within a document.'


<account> Account </account> [0..*]
'Optional account information used to precisely define the origination and destination
of financial instruments.'

End Group: PartiesAndAccounts.model
<reason> Reason </reason> [0..*]
</...>

```

**Diagram**



### Schema Component Representation

```

<xsd:complexType name="AllocationRefused">
  <xsd:complexContent>
    <xsd:extension base=" ResponseMessage ">
      <xsd:sequence>
        <xsd:element name="blockTradeIdentifier" type=" TradeIdentifier " minOccurs="0"/>
        <xsd:element name="allocations" type=" Allocations " minOccurs="0"/>
        <xsd:group ref=" PartiesAndAccounts.model " minOccurs="0"/>
        <xsd:element name="reason" type=" Reason " minOccurs="0" maxOccurs="unbounded"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
  
```

[top](#)

### Complex Type: [ClearingConfirmed](#)

<b>Super-types:</b>	<a href="#">NotificationMessage</a> < <b>ClearingConfirmed</b> (by extension)
<b>Sub-types:</b>	None
<b>Name</b>	ClearingConfirmed
<b>Used by (from the same schema document)</b>	Element <a href="#">clearingConfirmed</a>
<b>Abstract</b>	no

### XML Instance Representation

```

<...
fpmlVersion=" xsd:token (value comes from list: {'5-0'}) [1]
'Indicate which version of the FpML Schema an FpML message adheres to.'

"
expectedBuild=" xsd:positiveInteger [0..1]
'This optional attribute can be supplied by a message creator in an FpML instance to
specify which build number of the schema was used to define the message when it was generated.'

"
actualBuild="8 [0..1]
  
```

'The specific build number of this schema version. This attribute is not included in an instance document. Instead, it is supplied by the XML parser when the document is validated against the FpML schema and indicates the build number of the schema file. Every time FpML publishes a change to the schema, validation rules, or examples within a version (e.g., version 4.2) the actual build number is incremented. If no changes have been made between releases within a version (i.e. from Trial Recommendation to Recommendation) the actual build number stays the same.'

```
">   <header> NotificationMessageHeader </header> [1]
    <validation> Validation </validation> [0..*]
Start Group: Correlation.model [0..1]
  <correlationId> CorrelationId </correlationId> [1]
    'A qualified identifier used to correlate between messages'
```

End Group: Correlation.model  
Start Group: Sequence.model [0..1]  
 <sequenceNumber> xsd:positiveInteger </sequenceNumber> [1]  
'A numeric value that can be used to order messages with the same correlation identifier from the same sender.'

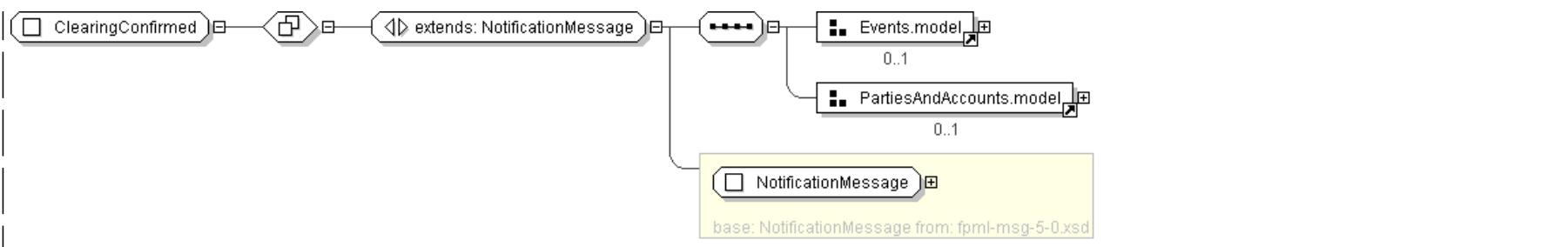
End Group: Sequence.model  
Start Group: OnBehalfOf.model [0..1]  
 <onBehalfOf> OnBehalfOf </onBehalfOf> [1]  
'Indicates which party (and accounts) a trade is being processed for.'

End Group: OnBehalfOf.model  
Start Group: Events.model [0..1]  
Start Choice [1]
 <trade> Trade </trade> [1]
 <amendment> TradeAmendmentContent </amendment> [1]
 <increase> TradeNotionalChange </increase> [1]
 <termination> TradeNotionalChange </termination> [1]
 <novation> TradeNovationContent </novation> [1]
 <additionalEvent> ... </additionalEvent> [1]
End Choice  
End Group: Events.model  
Start Group: PartiesAndAccounts.model [0..1]  
 <party> Party </party> [1..\*]  
'A legal entity or a subdivision of a legal entity.', 'Parties can perform multiple roles in a trade lifecycle. For example, the principal parties obligated to make payments from time to time during the term of the trade, but may include other parties involved in, or incidental to, the trade, such as parties acting in the role of novation transferor/transferee, broker, calculation agent, etc. In FpML roles are defined in multiple places within a document.'

```
<account> Account </account> [0..*]
'Optional account information used to precisely define the origination and destination of financial instruments.'
```

End Group: PartiesAndAccounts.model  
</...>

## Diagram



### Schema Component Representation

```

<xsd:complexType name="ClearingConfirmed">
  <xsd:complexContent>
    <xsd:extension base=" NotificationMessage ">
      <xsd:sequence>
        <xsd:group ref=" Events.model " minOccurs="0"/>
        <xsd:group ref=" PartiesAndAccounts.model " minOccurs="0"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
  
```

[top](#)

### Complex Type: **ClearingRefused**

<b>Super-types:</b>	<a href="#">NotificationMessage</a> < <b>ClearingRefused</b> (by extension)
<b>Sub-types:</b>	None

<b>Name</b>	ClearingRefused
<b>Used by (from the same schema document)</b>	Element <a href="#">clearingRefused</a>
<b>Abstract</b>	no

### XML Instance Representation

```

<...
  fpmlVersion=" xsd:token (value comes from list: {'5-0'}) [1]
  'Indicate which version of the FpML Schema an FpML message adheres to.'

  "
  expectedBuild=" xsd:positiveInteger [0..1]
  'This optional attribute can be supplied by a message creator in an FpML instance to
  specify which build number of the schema was used to define the message when it was generated.'

  "
  actualBuild="8 [0..1]
  'The specific build number of this schema version. This attribute is not included in
  an instance document. Instead, it is supplied by the XML parser when the document is
  validated against the FpML schema and indicates the build number of the schema file. Every
  time FpML publishes a change to the schema, validation rules, or examples within a version
  (e.g., version 4.2) the actual build number is incremented. If no changes have been
  made between releases within a version (i.e. from Trial Recommendation to Recommendation)
  the actual build number stays the same.'

  >
  
```

```

|   <header> NotificationMessageHeader </header> [1]
|   <validation> Validation </validation> [0..*]
Start Group: Correlation.model [0..1]
  <correlationId> CorrelationId </correlationId> [1]
    'A qualified identifier used to correlate between messages'

End Group: Correlation.model
Start Group: Sequence.model [0..1]
  <sequenceNumber> xsd:positiveInteger </sequenceNumber> [1]
    'A numeric value that can be used to order messages with the same correlation identifier
    from the same sender.'

End Group: Sequence.model
Start Group: OnBehalfOf.model [0..1]
  <onBehalfOf> OnBehalfOf </onBehalfOf> [1]
    'Indicates which party (and accounts) a trade is being processed for.'

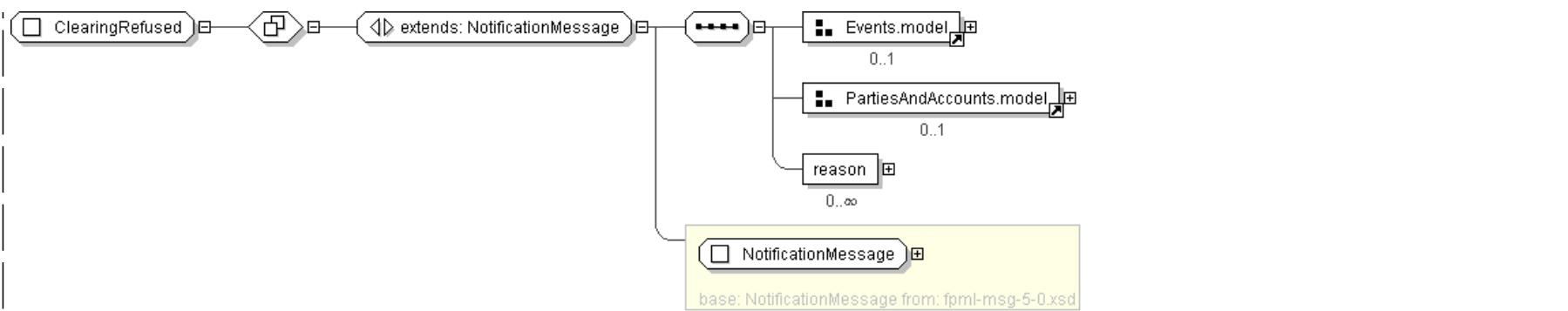
End Group: OnBehalfOf.model
Start Group: Events.model [0..1]
Start Choice [1]
  <trade> Trade </trade> [1]
  <amendment> TradeAmendmentContent </amendment> [1]
  <increase> TradeNotionalChange </increase> [1]
  <termination> TradeNotionalChange </termination> [1]
  <novation> TradeNovationContent </novation> [1]
  <additionalEvent> ... </additionalEvent> [1]
End Choice
End Group: Events.model
Start Group: PartiesAndAccounts.model [0..1]
  <party> Party </party> [1..*]
    'A legal entity or a subdivision of a legal entity.', 'Parties can perform multiple roles in
    a trade lifecycle. For example, the principal parties obligated to make payments from time
    to time during the term of the trade, but may include other parties involved in, or
    incidental to, the trade, such as parties acting in the role of novation transferor/
    transferee, broker, calculation agent, etc. In FpML roles are defined in multiple places
    within a document.'

  <account> Account </account> [0..*]
    'Optional account information used to precisely define the origination and destination
    of financial instruments.'

End Group: PartiesAndAccounts.model
  <reason> Reason </reason> [0..*]
</...>

```

**Diagram**



### Schema Component Representation

```

<xsd:complexType name="ClearingRefused">
  <xsd:complexContent>
    <xsd:extension base=" NotificationMessage ">
      <xsd:sequence>
        <xsd:group ref=" Events.model " minOccurs="0"/>
        <xsd:group ref=" PartiesAndAccounts.model " minOccurs="0"/>
        <xsd:element name="reason" type=" Reason " minOccurs="0" maxOccurs="unbounded"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

[top](#)

### Complex Type: ConfirmationAgreed

**Super-types:** [ResponseMessage](#) < **ConfirmationAgreed** (by extension)

**Sub-types:** None

<b>Name</b>	ConfirmationAgreed
-------------	--------------------

<b>Used by (from the same schema document)</b>	Element <a href="#">confirmationAgreed</a>
--	--

<b>Abstract</b>	no
-----------------	----

#### XML Instance Representation

```

<...
  fpmlVersion=" xsd:token (value comes from list: {'5-0'}) [1]
  'Indicate which version of the FpML Schema an FpML message adheres to.'
  "
  expectedBuild=" xsd:positiveInteger [0..1]
  'This optional attribute can be supplied by a message creator in an FpML instance to
  specify which build number of the schema was used to define the message when it was generated.'
  "
  actualBuild="8 [0..1]
  'The specific build number of this schema version. This attribute is not included in
  an instance document. Instead, it is supplied by the XML parser when the document is
  validated against the FpML schema and indicates the build number of the schema file. Every
  time FpML publishes a change to the schema, validation rules, or examples within a version
  (e.g., version 4.2) the actual build number is incremented. If no changes have been
  '

```

*made between releases within a version (i.e. from Trial Recommendation to Recommendation)  
the actual build number stays the same.'*

">  
<header> ResponseMessageHeader </header> [1]  
<validation> Validation </validation> [0..\*]  
<correlationId> CorrelationId </correlationId> [1]  
*'A qualified identifier used to correlate between messages'*

Start Group: Sequence.model [0..1]  
<sequenceNumber> xsd:positiveInteger </sequenceNumber> [1]  
*'A numeric value that can be used to order messages with the same correlation identifier  
from the same sender.'*

End Group: Sequence.model  
Start Group: OnBehalfOf.model [0..1]  
<onBehalfOf> OnBehalfOf </onBehalfOf> [1]  
*'Indicates which party (and accounts) a trade is being processed for.'*

End Group: OnBehalfOf.model  
Start Group: Events.model [0..1]  
Start Choice [1]  
  <trade> Trade </trade> [1]  
  <amendment> TradeAmendmentContent </amendment> [1]  
  <increase> TradeNotionalChange </increase> [1]  
  <termination> TradeNotionalChange </termination> [1]  
  <novation> TradeNovationContent </novation> [1]  
  <additionalEvent> ... </additionalEvent> [1]

End Choice  
End Group: Events.model  
Start Group: PartiesAndAccounts.model [0..1]  
  <party> Party </party> [1..\*]  
*'A legal entity or a subdivision of a legal entity.', 'Parties can perform multiple roles in  
a trade lifecycle. For example, the principal parties obligated to make payments from time  
to time during the term of the trade, but may include other parties involved in, or  
incidental to, the trade, such as parties acting in the role of novation transferor/  
transferee, broker, calculation agent, etc. In FpML roles are defined in multiple places  
within a document.'*

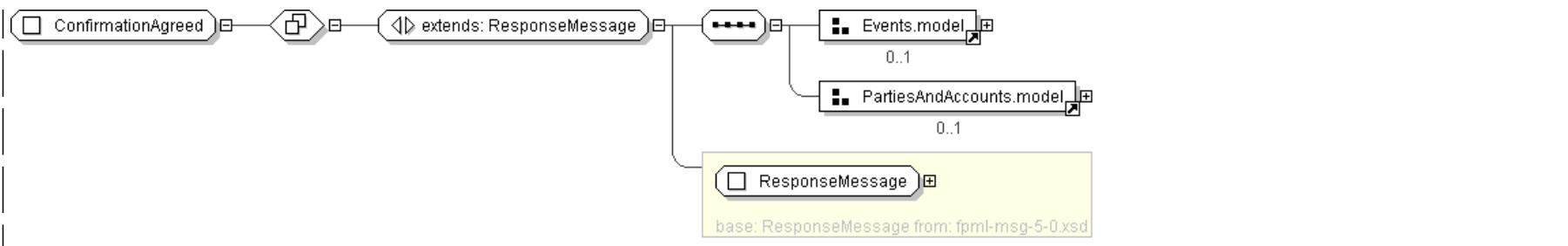
  <account> Account </account> [0..\*]  
*'Optional account information used to precisely define the origination and destination  
of financial instruments.'*

End Group: PartiesAndAccounts.model

</...>

## Diagram



**Schema Component Representation**

```

<xsd:complexType name="ConfirmationAgreed">
  <xsd:complexContent>
    <xsd:extension base=" ResponseMessage ">
      <xsd:sequence>
        <xsd:group ref=" Events.model " minOccurs="0"/>
        <xsd:group ref=" PartiesAndAccounts.model " minOccurs="0"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
  
```

[top](#)**Complex Type: ConfirmationDisputed**

<b>Super-types:</b>	<a href="#">ResponseMessage</a> < <b>ConfirmationDisputed</b> (by extension)
<b>Sub-types:</b>	None

<b>Name</b>	ConfirmationDisputed
-------------	----------------------

<b>Used by (from the same schema document)</b>	Element <a href="#">confirmationDisputed</a>
--	--

<b>Abstract</b>	no
-----------------	----

**XML Instance Representation**

```

<...
  fpmlVersion=" xsd:token (value comes from list: {'5-0'}) [1]
  'Indicate which version of the FpML Schema an FpML message adheres to.'

  "
  expectedBuild=" xsd:positiveInteger [0..1]
  'This optional attribute can be supplied by a message creator in an FpML instance to
  specify which build number of the schema was used to define the message when it was generated.'

  "
  actualBuild="8 [0..1]
  'The specific build number of this schema version. This attribute is not included in
  an instance document. Instead, it is supplied by the XML parser when the document is
  validated against the FpML schema and indicates the build number of the schema file. Every
  time FpML publishes a change to the schema, validation rules, or examples within a version
  (e.g., version 4.2) the actual build number is incremented. If no changes have been
  made between releases within a version (i.e. from Trial Recommendation to Recommendation)
  the actual build number stays the same.'

  >
  
```

```

<header> ResponseMessageHeader </header> [1]
<validation> Validation </validation> [0..*]
<correlationId> CorrelationId </correlationId> [1]

'A qualified identifier used to correlate between messages'

Start Group: Sequence.model [0..1]
<sequenceNumber> xsd:positiveInteger </sequenceNumber> [1]

'A numeric value that can be used to order messages with the same correlation identifier
from the same sender.'

End Group: Sequence.model
Start Group: OnBehalfOf.model [0..1]
<onBehalfOf> OnBehalfOf </onBehalfOf> [1]

'Indicates which party (and accounts) a trade is being processed for.'

End Group: OnBehalfOf.model
Start Group: Events.model [0..1]
Start Choice [1]
  <trade> Trade </trade> [1]
  <amendment> TradeAmendmentContent </amendment> [1]
  <increase> TradeNotionalChange </increase> [1]
  <termination> TradeNotionalChange </termination> [1]
  <novation> TradeNovationContent </novation> [1]
  <additionalEvent> ... </additionalEvent> [1]
End Choice
End Group: Events.model
Start Group: PartiesAndAccounts.model [0..1]
<party> Party </party> [1..*]

'A legal entity or a subdivision of a legal entity.', 'Parties can perform multiple roles in
a trade lifecycle. For example, the principal parties obligated to make payments from time
to time during the term of the trade, but may include other parties involved in, or
incidental to, the trade, such as parties acting in the role of novation transferor/
transferee, broker, calculation agent, etc. In FpML roles are defined in multiple places
within a document.'

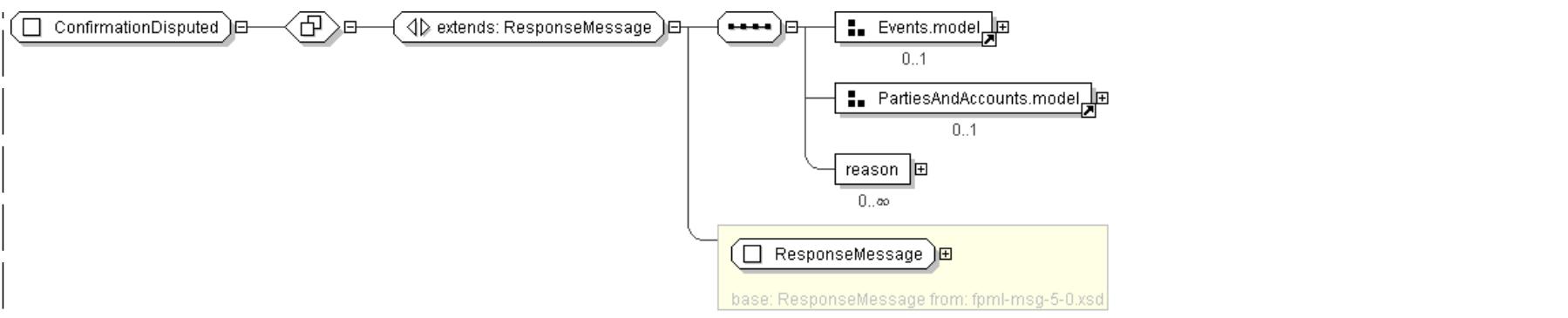
<account> Account </account> [0..*]

'Optional account information used to precisely define the origination and destination
of financial instruments.'

End Group: PartiesAndAccounts.model
<reason> Reason </reason> [0..*]
<...>

```

**Diagram**



### Schema Component Representation

```

<xsd:complexType name="ConfirmationDisputed">
  <xsd:complexContent>
    <xsd:extension base=" ResponseMessage ">
      <xsd:sequence>
        <xsd:group ref=" Events.model " minOccurs="0"/>
        <xsd:group ref=" PartiesAndAccounts.model " minOccurs="0"/>
        <xsd:element name="reason" type=" Reason " minOccurs="0" maxOccurs="unbounded"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
  
```

[top](#)

### Complex Type: ConfirmationRetracted

**Super-types:** [NonCorrectableRequestMessage](#) < **ConfirmationRetracted** (by extension)

**Sub-types:** None

<b>Name</b>	ConfirmationRetracted
-------------	-----------------------

<b>Used by (from the same schema document)</b>	Element <a href="#">requestConfirmationRetracted</a>
--	--

<b>Abstract</b>	no
-----------------	----

#### XML Instance Representation

```

<...
  fpmlVersion=" xsd:token (value comes from list: {'5-0'}) [1]
  'Indicate which version of the FpML Schema an FpML message adheres to.'
  "
  expectedBuild=" xsd:positiveInteger [0..1]
  'This optional attribute can be supplied by a message creator in an FpML instance to
  specify which build number of the schema was used to define the message when it was generated.'
  "
  actualBuild="8 [0..1]
  'The specific build number of this schema version. This attribute is not included in
  an instance document. Instead, it is supplied by the XML parser when the document is
  validated against the FpML schema and indicates the build number of the schema file. Every
  time FpML publishes a change to the schema, validation rules, or examples within a version
  (e.g., version 4.2) the actual build number is incremented. If no changes have been
  '
  
```

made between releases within a version (i.e. from Trial Recommendation to Recommendation)  
the actual build number stays the same.'

```
">
<header> RequestMessageHeader </header> [1]
<validation> Validation </validation> [0..*]
<correlationId> CorrelationId </correlationId> [1]
'A qualified identifier used to correlate between messages'

<sequenceNumber> xsd:positiveInteger </sequenceNumber> [1]
'A numeric value that can be used to order messages with the same correlation identifier
from the same sender.'
```

Start Group: OnBehalfOf.model [0..1]  
 <onBehalfOf> OnBehalfOf </onBehalfOf> [1]  
*'Indicates which party (and accounts) a trade is being processed for.'*

End Group: OnBehalfOf.model

Start Group: Events.model [0..1]  
 Start Choice [1]
 <trade> Trade </trade> [1]
 <amendment> TradeAmendmentContent </amendment> [1]
 <increase> TradeNotionalChange </increase> [1]
 <termination> TradeNotionalChange </termination> [1]
 <novation> TradeNovationContent </novation> [1]
 <additionalEvent> ... </additionalEvent> [1]

End Choice

End Group: Events.model

Start Group: PartiesAndAccounts.model [0..1]  
 <party> Party </party> [1..\*]
 *'A legal entity or a subdivision of a legal entity.' , 'Parties can perform multiple roles in
 a trade lifecycle. For example, the principal parties obligated to make payments from time
 to time during the term of the trade, but may include other parties involved in, or
 incidental to, the trade, such as parties acting in the role of novation transferor/
 transferee, broker, calculation agent, etc. In FpML roles are defined in multiple places
 within a document.'*

<account> Account </account> [0..\*]
*'Optional account information used to precisely define the origination and destination
 of financial instruments.'*

End Group: PartiesAndAccounts.model

</...>

#### Diagram



**Schema Component Representation**

```

<xsd:complexType name="ConfirmationRetracted">
  <xsd:complexContent>
    <xsd:extension base="NonCorrectableRequestMessage">
      <xsd:sequence>
        <xsd:group ref="Events.model" minOccurs="0"/>
        <xsd:group ref="PartiesAndAccounts.model" minOccurs="0"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

[top](#)**Complex Type: ConfirmationStatus**

**Super-types:** [ResponseMessage](#) < **ConfirmationStatus** (by extension)

**Sub-types:** None

**Name** ConfirmationStatus

**Used by (from the same schema document)** Element [confirmationStatus](#)

**Abstract** no

**XML Instance Representation**

```

<...
  fpmVersion="xsd:token (value comes from list: {'5-0'}) [1]
  'Indicate which version of the FpML Schema an FpML message adheres to.'

  "
  expectedBuild="xsd:positiveInteger [0..1]
  'This optional attribute can be supplied by a message creator in an FpML instance to
  specify which build number of the schema was used to define the message when it was generated.'

  "
  actualBuild="8 [0..1]
  'The specific build number of this schema version. This attribute is not included in
  an instance document. Instead, it is supplied by the XML parser when the document is
  validated against the FpML schema and indicates the build number of the schema file. Every
  time FpML publishes a change to the schema, validation rules, or examples within a version
  (e.g., version 4.2) the actual build number is incremented. If no changes have been
  made between releases within a version (i.e. from Trial Recommendation to Recommendation)
  the actual build number stays the same.'

  ">
  <header> ResponseMessageHeader </header> [1]
  <validation> Validation </validation> [0..*]
  <correlationId> CorrelationId </correlationId> [1]
  'A qualified identifier used to correlate between messages'

  Start Group: Sequence.model [0..1]
    <sequenceNumber> xsd:positiveInteger </sequenceNumber> [1]
    'A numeric value that can be used to order messages with the same correlation identifier
    from the same sender.'

  End Group: Sequence.model

```

| Start Group: OnBehalfOf.model [0..1]  
 | <onBehalfOf> OnBehalfOf </onBehalfOf> [1]  
 | 'Indicates which party (and accounts) a trade is being processed for.'

| End Group: OnBehalfOf.model  
 | <status> EventStatus </status> [1]  
 | 'Defines the confirmation status of a trade or post-trade event (e.g. Matched, Mismatched, Unmatched, Alleged).'

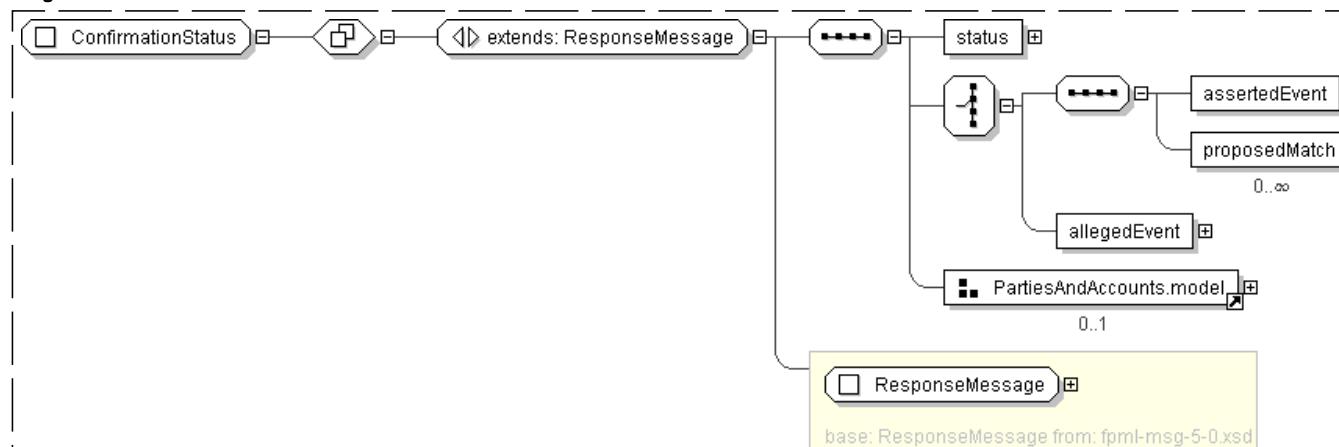
Start Choice [1]  
 <assertedEvent> EventsChoice </assertedEvent> [1]  
 <proposedMatch> EventProposedMatch </proposedMatch> [0..\*]  
 <allegedEvent> EventsChoice </allegedEvent> [1]

End Choice

Start Group: PartiesAndAccounts.model [0..1]  
 <party> Party </party> [1..\*]  
 'A legal entity or a subdivision of a legal entity.' , 'Parties can perform multiple roles in a trade lifecycle. For example, the principal parties obligated to make payments from time to time during the term of the trade, but may include other parties involved in, or incidental to, the trade, such as parties acting in the role of novation transferor/transferee, broker, calculation agent, etc. In FpML roles are defined in multiple places within a document.'

<account> Account </account> [0..\*]  
 'Optional account information used to precisely define the origination and destination of financial instruments.'

End Group: PartiesAndAccounts.model  
 </...>

**Diagram****Schema Component Representation**

```

<xsd:complexType name="ConfirmationStatus">
  <xsd:complexContent>
    <xsd:extension base=" ResponseMessage ">
      <xsd:sequence>
        <xsd:element name="status" type=" EventStatus " />
        <xsd:choice>
          <xsd:sequence>
  
```

```

<xsd:element name="assertedEvent" type="EventsChoice" />
<xsd:element name="proposedMatch" type="EventProposedMatch"
  minOccurs="0" maxOccurs="unbounded" />
</xsd:sequence>
<xsd:element name="allegedEvent" type="EventsChoice" />
</xsd:choice>
<xsd:group ref=" PartiesAndAccounts.model " minOccurs="0" />
</xsd:sequence>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>
</xsd:complexType>

```

[top](#)

## Complex Type: ConsentGranted

<b>Super-types:</b>	<a href="#">ResponseMessage</a> < <b>ConsentGranted</b> (by extension)
<b>Sub-types:</b>	None

<b>Name</b>	ConsentGranted
<b>Used by (from the same schema document)</b>	Element <a href="#">consentGranted</a>
<b>Abstract</b>	no

### XML Instance Representation

```

<...
  fpmlVersion=" xsd:token (value comes from list: {'5-0'}) [1]
  'Indicate which version of the FpML Schema an FpML message adheres to.'

  "
  expectedBuild=" xsd:positiveInteger [0..1]
  'This optional attribute can be supplied by a message creator in an FpML instance to
  specify which build number of the schema was used to define the message when it was generated.'

  "
  actualBuild="8 [0..1]
  'The specific build number of this schema version. This attribute is not included in
  an instance document. Instead, it is supplied by the XML parser when the document is
  validated against the FpML schema and indicates the build number of the schema file. Every
  time FpML publishes a change to the schema, validation rules, or examples within a version
  (e.g., version 4.2) the actual build number is incremented. If no changes have been
  made between releases within a version (i.e. from Trial Recommendation to Recommendation)
  the actual build number stays the same.'

  ">
  <header> ResponseMessageHeader </header> [1]
  <validation> Validation </validation> [0..*]
  <correlationId> CorrelationId </correlationId> [1]
  'A qualified identifier used to correlate between messages'

  Start Group: Sequence.model [0..1]
    <sequenceNumber> xsd:positiveInteger </sequenceNumber> [1]
    'A numeric value that can be used to order messages with the same correlation identifier
    from the same sender.'

  End Group: Sequence.model

```

```

Start Group: OnBehalfOf.model [0..1]
<onBehalfOf> OnBehalfOf </onBehalfOf> [1]
'Indicates which party (and accounts) a trade is being processed for.'

End Group: OnBehalfOf.model
Start Group: Events.model [0..1]
Start Choice [1]
  <trade> Trade </trade> [1]
  <amendment> TradeAmendmentContent </amendment> [1]
  <increase> TradeNotionalChange </increase> [1]
  <termination> TradeNotionalChange </termination> [1]
  <novation> TradeNovationContent </novation> [1]
  <additionalEvent> ... </additionalEvent> [1]
End Choice
End Group: Events.model
Start Group: PartiesAndAccounts.model [0..1]
  <party> Party </party> [1..*]
'A legal entity or a subdivision of a legal entity.', 'Parties can perform multiple roles in a trade lifecycle. For example, the principal parties obligated to make payments from time to time during the term of the trade, but may include other parties involved in, or incidental to, the trade, such as parties acting in the role of novation transferor/transferee, broker, calculation agent, etc. In FpML roles are defined in multiple places within a document.'
  <account> Account </account> [0..*]
'Optional account information used to precisely define the origination and destination of financial instruments.'
End Group: PartiesAndAccounts.model
</...>

```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="ConsentGranted">
  <xsd:complexContent>
    <xsd:extension base="ResponseMessage">
      <xsd:sequence>
        <xsd:group ref="Events.model" minOccurs="0"/>
        <xsd:group ref="PartiesAndAccounts.model" minOccurs="0"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

**Complex Type: ConsentRefused**

<b>Super-types:</b>	<a href="#">ResponseMessage</a> < <b>ConsentRefused</b> (by extension)
<b>Sub-types:</b>	None

<b>Name</b>	ConsentRefused
<b>Used by (from the same schema document)</b>	Element <a href="#">consentRefused</a>
<b>Abstract</b>	no

**XML Instance Representation**

```

<...
  fpmlVersion=" xsd:token (value comes from list: {'5-0'}) [1]
  'Indicate which version of the FpML Schema an FpML message adheres to.'

  "
  expectedBuild=" xsd:positiveInteger [0..1]
  'This optional attribute can be supplied by a message creator in an FpML instance to
  specify which build number of the schema was used to define the message when it was generated.'

  "
  actualBuild="8 [0..1]
  'The specific build number of this schema version. This attribute is not included in
  an instance document. Instead, it is supplied by the XML parser when the document is
  validated against the FpML schema and indicates the build number of the schema file. Every
  time FpML publishes a change to the schema, validation rules, or examples within a version
  (e.g., version 4.2) the actual build number is incremented. If no changes have been
  made between releases within a version (i.e. from Trial Recommendation to Recommendation)
  the actual build number stays the same.'

  ">
  <header> ResponseMessageHeader </header> [1]
  <validation> Validation </validation> [0..*]
  <correlationId> CorrelationId </correlationId> [1]
  'A qualified identifier used to correlate between messages'

  Start Group: Sequence.model [0..1]
    <sequenceNumber> xsd:positiveInteger </sequenceNumber> [1]
    'A numeric value that can be used to order messages with the same correlation identifier
    from the same sender.'

  End Group: Sequence.model
  Start Group: OnBehalfOf.model [0..1]
    <onBehalfOf> OnBehalfOf </onBehalfOf> [1]
    'Indicates which party (and accounts) a trade is being processed for.'

  End Group: OnBehalfOf.model
  Start Group: Events.model [0..1]
  Start Choice [1]
    <trade> Trade </trade> [1]
    <amendment> TradeAmendmentContent </amendment> [1]
    <increase> TradeNotionalChange </increase> [1]
    <termination> TradeNotionalChange </termination> [1]
    <novation> TradeNovationContent </novation> [1]
    <additionalEvent> ... </additionalEvent> [1]
  End Choice

```

```

End Group: Events.model
Start Group: PartiesAndAccounts.model [0..1]
<party> Party </party> [1..*]

'A legal entity or a subdivision of a legal entity.', 'Parties can perform multiple roles in
a trade lifecycle. For example, the principal parties obligated to make payments from time
to time during the term of the trade, but may include other parties involved in, or
incidental to, the trade, such as parties acting in the role of novation transferor/
transferee, broker, calculation agent, etc. In FpML roles are defined in multiple places
within a document.'

```

```

<account> Account </account> [0..*]

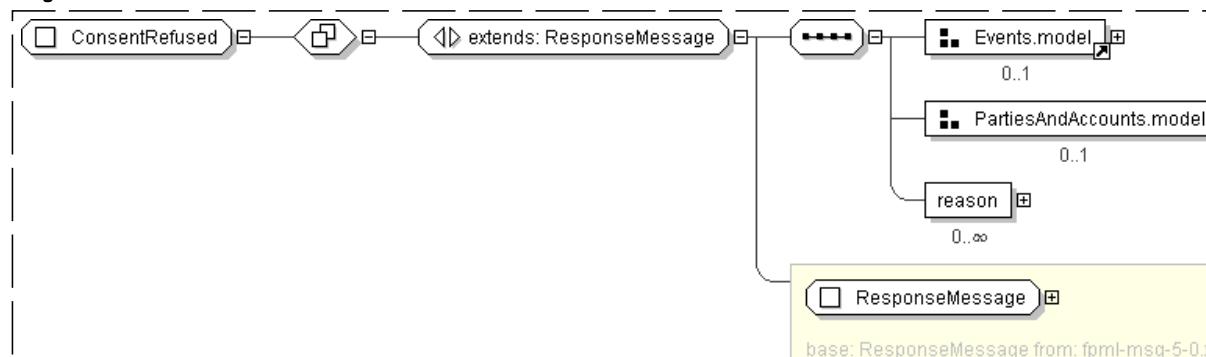
'Optional account information used to precisely define the origination and destination
of financial instruments.'

```

```

End Group: PartiesAndAccounts.model
<reason> Reason </reason> [0..*]
</...>

```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="ConsentRefused">
  <xsd:complexContent>
    <xsd:extension base=" ResponseMessage ">
      <xsd:sequence>
        <xsd:group ref=" Events.model " minOccurs="0"/>
        <xsd:group ref=" PartiesAndAccounts.model " minOccurs="0"/>
        <xsd:element name="reason" type=" Reason " minOccurs="0" maxOccurs="unbounded" />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

[top](#)**Complex Type: ExecutionAdvice**

<b>Super-types:</b>	<u>CorrectableRequestMessage</u> < <b>ExecutionAdvice</b> (by extension)
<b>Sub-types:</b>	None

<b>Name</b>	ExecutionAdvice
-------------	-----------------

**Used by (from the same schema document)**Element [executionAdvice](#)**Abstract**

no

**XML Instance Representation**

```

<...
  fpmlVersion=" xsd:token (value comes from list: {'5-0'}) [1]
  'Indicate which version of the FpML Schema an FpML message adheres to.'

  "
  expectedBuild=" xsd:positiveInteger [0..1]
  'This optional attribute can be supplied by a message creator in an FpML instance to
  specify which build number of the schema was used to define the message when it was generated.'

  "
  actualBuild="8 [0..1]
  'The specific build number of this schema version. This attribute is not included in
  an instance document. Instead, it is supplied by the XML parser when the document is
  validated against the FpML schema and indicates the build number of the schema file. Every
  time FpML publishes a change to the schema, validation rules, or examples within a version
  (e.g., version 4.2) the actual build number is incremented. If no changes have been
  made between releases within a version (i.e. from Trial Recommendation to Recommendation)
  the actual build number stays the same.'

  ">
  <header> RequestMessageHeader </header> [1]
  <validation> Validation </validation> [0..*]
  <isCorrection> xsd:boolean </isCorrection> [1]
  'Indicates if this message corrects an earlier request.'

  <correlationId> CorrelationId </correlationId> [1]
  'A qualified identifier used to correlate between messages'

  <sequenceNumber> xsd:positiveInteger </sequenceNumber> [1]
  'A numeric value that can be used to order messages with the same correlation identifier
  from the same sender.'

Start Group: OnBehalfOf.model [0..1]
  <onBehalfOf> OnBehalfOf </onBehalfOf> [1]
  'Indicates which party (and accounts) a trade is being processed for.'

End Group: OnBehalfOf.model
Start Choice [1]
  <trade> Trade </trade> [1]
  <amendment> TradeAmendmentContent </amendment> [1]
  <increase> TradeNotionalChange </increase> [1]
  <termination> TradeNotionalChange </termination> [1]
  <novation> TradeNovationContent </novation> [1]
  <additionalEvent> ... </additionalEvent> [1]
End Choice
  <paymentDetails> PaymentDetails </paymentDetails> [0..*]
  'Details of the payments, like amount breakdowns, settlement information.'

  <party> Party </party> [1..*]
  'A legal entity or a subdivision of a legal entity.', 'Parties can perform multiple roles in
  a trade lifecycle. For example, the principal parties obligated to make payments from time
  to time during the term of the trade, but may include other parties involved in, or

```

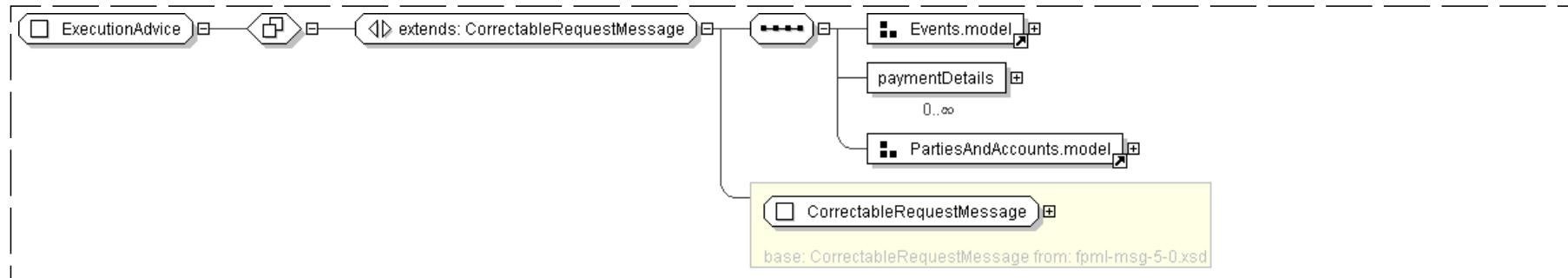
*incidental to, the trade, such as parties acting in the role of novation transferor/transferee, broker, calculation agent, etc. In FpML roles are defined in multiple places within a document.'*

<account> Account </account> [0..\*]

'Optional account information used to precisely define the origination and destination of financial instruments.'

</...>

#### Diagram



#### Schema Component Representation

```

<xsd:complexType name="ExecutionAdvice">
  <xsd:complexContent>
    <xsd:extension base=" CorrectableRequestMessage ">
      <xsd:sequence>
        <xsd:group ref=" Events.model "/>
        <xsd:element name="paymentDetails" type=" PaymentDetails " minOccurs="0" maxOccurs="unbounded"/>
        <xsd:group ref=" PartiesAndAccounts.model "/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

top

## Complex Type: ExecutionAdviceRetracted

Super-types:	<a href="#">NonCorrectableRequestMessage</a> < <b>ExecutionAdviceRetracted</b> (by extension)
Sub-types:	None

Name	ExecutionAdviceRetracted
Used by (from the same schema document)	Element <a href="#">executionAdviceRetracted</a>
Abstract	no

#### XML Instance Representation

```

<...
  fpmlVersion=" xsd:token (value comes from list: {'5-0'}) [1]
  'Indicate which version of the FpML Schema an FpML message adheres to.'
  "

```

```
| expectedBuild= "xsd:positiveInteger [0..1]
```

'This optional attribute can be supplied by a message creator in an FpML instance to specify which build number of the schema was used to define the message when it was generated.'

```
"  
actualBuild="8 [0..1]
```

'The specific build number of this schema version. This attribute is not included in an instance document. Instead, it is supplied by the XML parser when the document is validated against the FpML schema and indicates the build number of the schema file. Every time FpML publishes a change to the schema, validation rules, or examples within a version (e.g., version 4.2) the actual build number is incremented. If no changes have been made between releases within a version (i.e. from Trial Recommendation to Recommendation) the actual build number stays the same.'

```
">  
<header> RequestMessageHeader </header> [1]  
<validation> Validation </validation> [0..*]
```

<correlationId> CorrelationId </correlationId> [1]

'A qualified identifier used to correlate between messages'

```
<sequenceNumber> xsd:positiveInteger </sequenceNumber> [1]
```

'A numeric value that can be used to order messages with the same correlation identifier from the same sender.'

```
Start Group: OnBehalfOf.model [0..1]
```

<onBehalfOf> OnBehalfOf </onBehalfOf> [1]

'Indicates which party (and accounts) a trade is being processed for.'

End Group: OnBehalfOf.model

```
Start Group: Events.model [0..1]
```

Start Choice [1]

```
    <trade> Trade </trade> [1]  
    <amendment> TradeAmendmentContent </amendment> [1]  
    <increase> TradeNotionalChange </increase> [1]  
    <termination> TradeNotionalChange </termination> [1]  
    <novation> TradeNovationContent </novation> [1]  
    <additionalEvent> ... </additionalEvent> [1]
```

End Choice

End Group: Events.model

```
<paymentDetails> PaymentDetails </paymentDetails> [0..*]
```

'Details of the payments, like amount breakdowns, settlement information.'

```
Start Group: PartiesAndAccounts.model [0..1]
```

<party> Party </party> [1..\*]

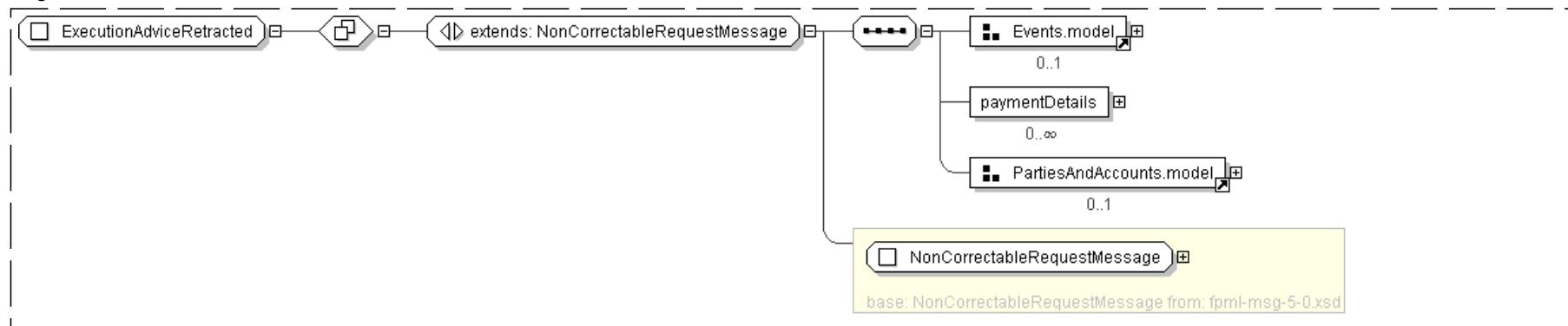
'A legal entity or a subdivision of a legal entity.' , 'Parties can perform multiple roles in a trade lifecycle. For example, the principal parties obligated to make payments from time to time during the term of the trade, but may include other parties involved in, or incidental to, the trade, such as parties acting in the role of novation transferor/transferee, broker, calculation agent, etc. In FpML roles are defined in multiple places within a document.'

```
<account> Account </account> [0..*]
```

'Optional account information used to precisely define the origination and destination of financial instruments.'

End Group: PartiesAndAccounts.model

</...>

**Diagram****Schema Component Representation**

```

<xsd:complexType name="ExecutionAdviceRetracted">
  <xsd:complexContent>
    <xsd:extension base=" NonCorrectableRequestMessage ">
      <xsd:sequence>
        <xsd:group ref=" Events.model " minOccurs="0"/>
        <xsd:element name="paymentDetails" type=" PaymentDetails " minOccurs="0" maxOccurs="unbounded" />
        <xsd:group ref=" PartiesAndAccounts.model " minOccurs="0" />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
  
```

[top](#)**Complex Type: ExecutionNotification**

**Super-types:** [CorrectableRequestMessage](#) < **ExecutionNotification** (by extension)

**Sub-types:** None

<b>Name</b>	ExecutionNotification
-------------	-----------------------

<b>Used by (from the same schema document)</b>	Element <a href="#">executionNotification</a>
--	---

<b>Abstract</b>	no
-----------------	----

**XML Instance Representation**

```

<...
  fpmlVersion=" xsd:token (value comes from list: {'5-0'}) [1]
  'Indicate which version of the FpML Schema an FpML message adheres to.'

  "
  expectedBuild=" xsd:positiveInteger [0..1]
  'This optional attribute can be supplied by a message creator in an FpML instance to
  specify which build number of the schema was used to define the message when it was generated.'

  "
  actualBuild="8 [0..1]
  'The specific build number of this schema version. This attribute is not included in
  an instance document. Instead, it is supplied by the XML parser when the document is
  validated against the FpML schema and indicates the build number of the schema file. Every
  
```

time FpML publishes a change to the schema, validation rules, or examples within a version (e.g., version 4.2) the actual build number is incremented. If no changes have been made between releases within a version (i.e. from Trial Recommendation to Recommendation) the actual build number stays the same.'

```
">
<header> RequestMessageHeader </header> [1]
<validation> Validation </validation> [0..*]
<isCorrection> xsd:boolean </isCorrection> [1]
'Indicates if this message corrects an earlier request.'

<correlationId> CorrelationId </correlationId> [1]
'A qualified identifier used to correlate between messages'

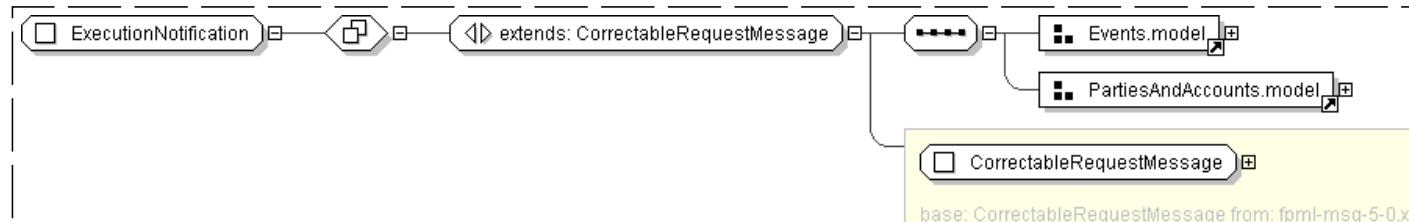
<sequenceNumber> xsd:positiveInteger </sequenceNumber> [1]
'A numeric value that can be used to order messages with the same correlation identifier
from the same sender.'

Start Group: OnBehalfOf.model [0..1]
<onBehalfOf> OnBehalfOf </onBehalfOf> [1]
'Indicates which party (and accounts) a trade is being processed for.'

End Group: OnBehalfOf.model
Start Choice [1]
<trade> Trade </trade> [1]
<amendment> TradeAmendmentContent </amendment> [1]
<increase> TradeNotionalChange </increase> [1]
<termination> TradeNotionalChange </termination> [1]
<novation> TradeNovationContent </novation> [1]
<additionalEvent> ... </additionalEvent> [1]
End Choice
<party> Party </party> [1..*]
'A legal entity or a subdivision of a legal entity.', 'Parties can perform multiple roles in
a trade lifecycle. For example, the principal parties obligated to make payments from time
to time during the term of the trade, but may include other parties involved in, or
incidental to, the trade, such as parties acting in the role of novation transferor/
transferee, broker, calculation agent, etc. In FpML roles are defined in multiple places
within a document.'

<account> Account </account> [0..*]
'Optional account information used to precisely define the origination and destination
of financial instruments.

</...>
```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="ExecutionNotification">
  <xsd:complexContent>
    <xsd:extension base=" CorrectableRequestMessage ">
      <xsd:sequence>
        <xsd:group ref=" Events.model "/>
        <xsd:group ref=" PartiesAndAccounts.model "/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

[top](#)

## Complex Type: [ExecutionRetracted](#)

**Super-types:** [NonCorrectableRequestMessage](#) < **ExecutionRetracted** (by extension)

**Sub-types:** None

**Name** ExecutionRetracted

**Used by (from the same schema document)** Element [executionRetracted](#)

**Abstract** no

### XML Instance Representation

```

<...
  fpmlVersion=" xsd:token (value comes from list: {'5-0'}) [1]
  'Indicate which version of the FpML Schema an FpML message adheres to.'

  "
  expectedBuild=" xsd:positiveInteger [0..1]
  'This optional attribute can be supplied by a message creator in an FpML instance to
  specify which build number of the schema was used to define the message when it was generated.'

  "
  actualBuild="8 [0..1]
  'The specific build number of this schema version. This attribute is not included in
  an instance document. Instead, it is supplied by the XML parser when the document is
  validated against the FpML schema and indicates the build number of the schema file. Every
  time FpML publishes a change to the schema, validation rules, or examples within a version
  (e.g., version 4.2) the actual build number is incremented. If no changes have been
  made between releases within a version (i.e. from Trial Recommendation to Recommendation)
  the actual build number stays the same.'

  "
  <header> RequestMessageHeader </header> [1]
  <validation> Validation </validation> [0..*]
  <correlationId> CorrelationId </correlationId> [1]
  'A qualified identifier used to correlate between messages'

  <sequenceNumber> xsd:positiveInteger </sequenceNumber> [1]
  'A numeric value that can be used to order messages with the same correlation identifier
  from the same sender.'

  Start Group: OnBehalfOf.model [0..1]
  <onBehalfOf> OnBehalfOf </onBehalfOf> [1]
  'Indicates which party (and accounts) a trade is being processed for.'

```

```

End Group: OnBehalfOf.model
Start Group: Events.model [0..1]
Start Choice [1]
  <trade> Trade </trade> [1]
  <amendment> TradeAmendmentContent </amendment> [1]
  <increase> TradeNotionalChange </increase> [1]
  <termination> TradeNotionalChange </termination> [1]
  <novation> TradeNovationContent </novation> [1]
  <additionalEvent> ... </additionalEvent> [1]
End Choice
End Group: Events.model
Start Group: PartiesAndAccounts.model [0..1]
  <party> Party </party> [1..*]
    'A legal entity or a subdivision of a legal entity.', 'Parties can perform multiple roles in a trade lifecycle. For example, the principal parties obligated to make payments from time to time during the term of the trade, but may include other parties involved in, or incidental to, the trade, such as parties acting in the role of novation transferor/ transferee, broker, calculation agent, etc. In FpML roles are defined in multiple places within a document.'
  <account> Account </account> [0..*]
    'Optional account information used to precisely define the origination and destination of financial instruments.'
End Group: PartiesAndAccounts.model
</...>

```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="ExecutionRetracted">
  <xsd:complexContent>
    <xsd:extension base="NonCorrectableRequestMessage">
      <xsd:sequence>
        <xsd:group ref="Events.model" minOccurs="0"/>
        <xsd:group ref="PartiesAndAccounts.model" minOccurs="0"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

**Complex Type: RequestAllocation**

**Super-types:** [CorrectableRequestMessage](#) < **RequestAllocation** (by extension)

**Sub-types:** None

**Name** RequestAllocation

**Used by (from the same schema document)** Element [requestAllocation](#)

**Abstract** no

#### XML Instance Representation

```
<...>
  fpmVersion=" xsd:token (value comes from list: {'5-0'}) [1]
  'Indicate which version of the FpML Schema an FpML message adheres to.'

  "
  expectedBuild=" xsd:positiveInteger [0..1]
  'This optional attribute can be supplied by a message creator in an FpML instance to
  specify which build number of the schema was used to define the message when it was generated.'

  "
  actualBuild="8 [0..1]
  'The specific build number of this schema version. This attribute is not included in
  an instance document. Instead, it is supplied by the XML parser when the document is
  validated against the FpML schema and indicates the build number of the schema file. Every
  time FpML publishes a change to the schema, validation rules, or examples within a version
  (e.g., version 4.2) the actual build number is incremented. If no changes have been
  made between releases within a version (i.e. from Trial Recommendation to Recommendation)
  the actual build number stays the same.'

  ">
    <header> RequestMessageHeader </header> [1]
    <validation> Validation </validation> [0..*]
    <isCorrection> xsd:boolean </isCorrection> [1]
    'Indicates if this message corrects an earlier request.'

    <correlationId> CorrelationId </correlationId> [1]
    'A qualified identifier used to correlate between messages'

    <sequenceNumber> xsd:positiveInteger </sequenceNumber> [1]
    'A numeric value that can be used to order messages with the same correlation identifier
    from the same sender.'

  Start Group: OnBehalfOf.model [0..1]
    <onBehalfOf> OnBehalfOf </onBehalfOf> [1]
    'Indicates which party (and accounts) a trade is being processed for.'

  End Group: OnBehalfOf.model
    <relatedParty> RelatedParty </relatedParty> [0..*]
    'Identifies a related party performing a role within the transaction.'

    <blockTradeIdentifier> TradeIdentifier </blockTradeIdentifier> [1]
    <allocations> Allocations </allocations> [1]
    <party> Party </party> [1..*]
    'A legal entity or a subdivision of a legal entity.', 'Parties can perform multiple roles in
    a trade lifecycle. For example, the principal parties obligated to make payments from time
    to time during the term of the trade, but may include other parties involved in, or
    incidental to, the trade, such as parties acting in the role of novation transferor/
    transferee, broker, calculation agent, etc. In FpML roles are defined in multiple places'
```

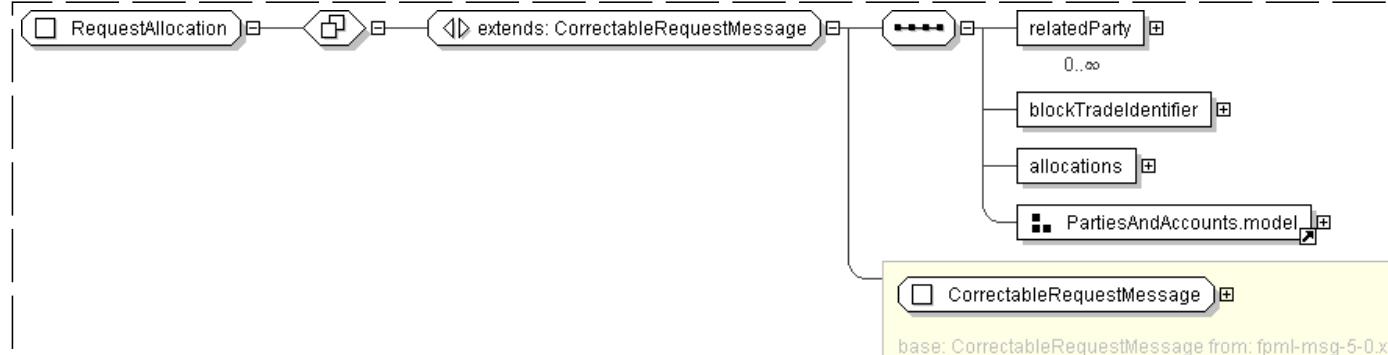
*within a document.'*

<account> Account </account> [0..\*]

'Optional account information used to precisely define the origination and destination of financial instruments.'

</...>

#### Diagram



#### Schema Component Representation

```

<xsd:complexType name="RequestAllocation">
  <xsd:complexContent>
    <xsd:extension base="CorrectableRequestMessage">
      <xsd:sequence>
        <xsd:element name="relatedParty" type="RelatedParty" minOccurs="0" maxOccurs="unbounded"/>
        <xsd:element name="blockTradeIdentifier" type="TradeIdentifier"/>
        <xsd:element name="allocations" type="Allocations"/>
        <xsd:group ref="PartiesAndAccounts.model"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

[top](#)

### Complex Type: RequestAllocationRetracted

Super-types: [NonCorrectableRequestMessage](#) < RequestAllocationRetracted (by extension)

Sub-types: None

Name	RequestAllocationRetracted
------	----------------------------

Used by (from the same schema document)	Element <a href="#">requestAllocationRetracted</a>
---	--

Abstract	no
----------	----

#### XML Instance Representation

```

<...
fpmlVersion="xsd:token (value comes from list: {'5-0'})" [1]
'Indicate which version of the FpML Schema an FpML message adheres to.'

```

```

"
expectedBuild= "xsd:positiveInteger [0..1]
'This optional attribute can be supplied by a message creator in an FpML instance to
specify which build number of the schema was used to define the message when it was generated.'

"
actualBuild="8 [0..1]
'The specific build number of this schema version. This attribute is not included in
an instance document. Instead, it is supplied by the XML parser when the document is
validated against the FpML schema and indicates the build number of the schema file. Every
time FpML publishes a change to the schema, validation rules, or examples within a version
(e.g., version 4.2) the actual build number is incremented. If no changes have been
made between releases within a version (i.e. from Trial Recommendation to Recommendation)
the actual build number stays the same.'


">
<header> RequestMessageHeader </header> [1]
<validation> Validation </validation> [0..*]
<correlationId> CorrelationId </correlationId> [1]
'A qualified identifier used to correlate between messages'

<sequenceNumber> xsd:positiveInteger </sequenceNumber> [1]
'A numeric value that can be used to order messages with the same correlation identifier
from the same sender.'


Start Group: OnBehalfOf.model [0..1]
<onBehalfOf> OnBehalfOf </onBehalfOf> [1]
'Indicates which party (and accounts) a trade is being processed for.'

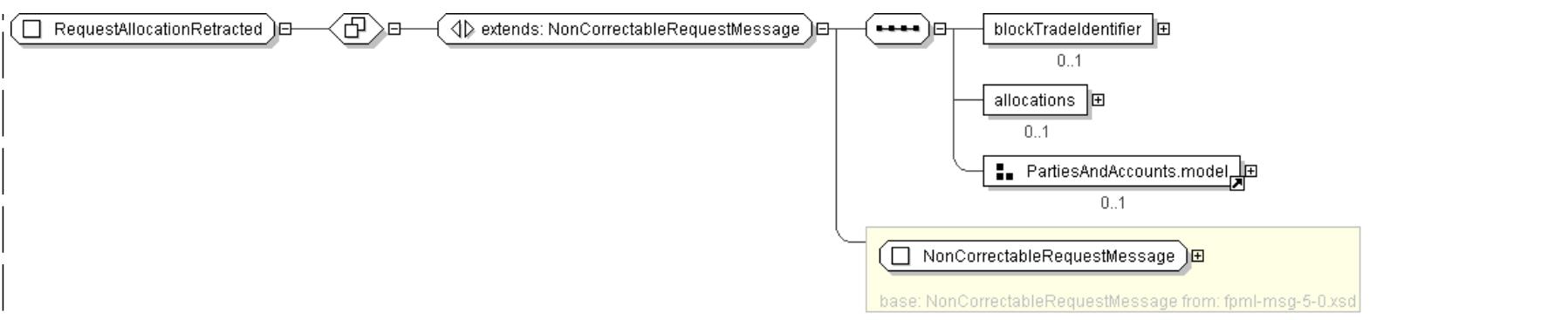

End Group: OnBehalfOf.model
<blockTradeIdentifier> TradeIdentifier </blockTradeIdentifier> [0..1]
<allocations> Allocations </allocations> [0..1]
Start Group: PartiesAndAccounts.model [0..1]
<party> Party </party> [1..*]
'A legal entity or a subdivision of a legal entity.', 'Parties can perform multiple roles in
a trade lifecycle. For example, the principal parties obligated to make payments from time
to time during the term of the trade, but may include other parties involved in, or
incidental to, the trade, such as parties acting in the role of novation transferor/
transferee, broker, calculation agent, etc. In FpML roles are defined in multiple places
within a document.'


<account> Account </account> [0..*]
'Optional account information used to precisely define the origination and destination
of financial instruments.'


End Group: PartiesAndAccounts.model
</...>

```

**Diagram**



### Schema Component Representation

```

<xsd:complexType name="RequestAllocationRetracted">
    <xsd:complexContent>
        <xsd:extension base=" NonCorrectableRequestMessage ">
            <xsd:sequence>
                <xsd:element name="blockTradeIdentifier" type=" TradeIdentifier " minOccurs="0"/>
                <xsd:element name="allocations" type=" Allocations " minOccurs="0"/>
                <xsd:group ref=" PartiesAndAccounts.model " minOccurs="0"/>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

```

top

### Complex Type: RequestClearing

**Super-types:** [CorrectableRequestMessage](#) < **RequestClearing** (by extension)

**Sub-types:** None

**Name** RequestClearing

**Used by (from the same schema document)** Element [requestClearing](#)

**Abstract** no

#### XML Instance Representation

```

<...
fpmlVersion=" xsd:token (value comes from list: {'5-0'}) [1]
'Indicate which version of the FpML Schema an FpML message adheres to.'

"
expectedBuild=" xsd:positiveInteger [0..1]
'This optional attribute can be supplied by a message creator in an FpML instance to
specify which build number of the schema was used to define the message when it was generated.'
"
```

```

actualBuild="8 [0..1]
'The specific build number of this schema version. This attribute is not included in
an instance document. Instead, it is supplied by the XML parser when the document is
validated against the FpML schema and indicates the build number of the schema file. Every
time FpML publishes a change to the schema, validation rules, or examples within a version
(e.g., version 4.2) the actual build number is incremented. If no changes have been

```

made between releases within a version (i.e. from Trial Recommendation to Recommendation)  
the actual build number stays the same.'

">

```

<header> RequestMessageHeader </header> [1]
<validation> Validation </validation> [0..*]
<isCorrection> xsd:boolean </isCorrection> [1]
'Indicates if this message corrects an earlier request.'

<correlationId> CorrelationId </correlationId> [1]
'A qualified identifier used to correlate between messages'

<sequenceNumber> xsd:positiveInteger </sequenceNumber> [1]
'A numeric value that can be used to order messages with the same correlation identifier
from the same sender.'
```

Start Group: OnBehalfOf.model [0..1]

```

<onBehalfOf> OnBehalfOf </onBehalfOf> [1]
'Indicates which party (and accounts) a trade is being processed for.'
```

End Group: OnBehalfOf.model

Start Choice [1]

```

<trade> Trade </trade> [1]
<amendment> TradeAmendmentContent </amendment> [1]
<increase> TradeNotionalChange </increase> [1]
<termination> TradeNotionalChange </termination> [1]
<novation> TradeNovationContent </novation> [1]
<additionalEvent> ... </additionalEvent> [1]
```

End Choice

<party> Party </party> [1..\*]

'A legal entity or a subdivision of a legal entity.', 'Parties can perform multiple roles in  
a trade lifecycle. For example, the principal parties obligated to make payments from time  
to time during the term of the trade, but may include other parties involved in, or  
incidental to, the trade, such as parties acting in the role of novation transferor/  
transferee, broker, calculation agent, etc. In FpML roles are defined in multiple places  
within a document.'

<account> Account </account> [0..\*]

'Optional account information used to precisely define the origination and destination  
of financial instruments.'

<...>

#### Diagram



#### Schema Component Representation

```

<xsd:complexType name="RequestClearing">
  <xsd:complexContent>
```

```

<xsd:extension base="CorrectableRequestMessage">
  <xsd:sequence>
    <xsd:group ref="Events.model" />
    <xsd:group ref="PartiesAndAccounts.model" />
  </xsd:sequence>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>

```

[top](#)

## Complex Type: [RequestClearingRetracted](#)

**Super-types:** [NonCorrectableRequestMessage](#) < **RequestClearingRetracted** (by extension)

**Sub-types:** None

<b>Name</b>	RequestClearingRetracted
<b>Used by (from the same schema document)</b>	Element <a href="#">requestClearingRetracted</a>
<b>Abstract</b>	no

### XML Instance Representation

```

<...
fpmVersion=" xsd:token (value comes from list: {'5-0'}) [1]
'Indicate which version of the FpML Schema an FpML message adheres to.'

"
expectedBuild=" xsd:positiveInteger [0..1]
'This optional attribute can be supplied by a message creator in an FpML instance to
specify which build number of the schema was used to define the message when it was generated.'

"
actualBuild="8 [0..1]
'The specific build number of this schema version. This attribute is not included in
an instance document. Instead, it is supplied by the XML parser when the document is
validated against the FpML schema and indicates the build number of the schema file. Every
time FpML publishes a change to the schema, validation rules, or examples within a version
(e.g., version 4.2) the actual build number is incremented. If no changes have been
made between releases within a version (i.e. from Trial Recommendation to Recommendation)
the actual build number stays the same.'

">
<header> RequestMessageHeader </header> [1]
<validation> Validation </validation> [0..*]
<correlationId> CorrelationId </correlationId> [1]
'A qualified identifier used to correlate between messages'

<sequenceNumber> xsd:positiveInteger </sequenceNumber> [1]
'A numeric value that can be used to order messages with the same correlation identifier
from the same sender.'

Start Group: OnBehalfOf.model [0..1]
<onBehalfOf> OnBehalfOf </onBehalfOf> [1]
'Indicates which party (and accounts) a trade is being processed for.'

End Group: OnBehalfOf.model

```

```

Start Group: Events.model [0..1]
Start Choice [1]
  <trade> Trade </trade> [1]
  <amendment> TradeAmendmentContent </amendment> [1]
  <increase> TradeNotionalChange </increase> [1]
  <termination> TradeNotionalChange </termination> [1]
  <novation> TradeNovationContent </novation> [1]
  <additionalEvent> ... </additionalEvent> [1]
End Choice
End Group: Events.model
Start Group: PartiesAndAccounts.model [0..1]
  <party> Party </party> [1..*]
    'A legal entity or a subdivision of a legal entity.', 'Parties can perform multiple roles in a trade lifecycle. For example, the principal parties obligated to make payments from time to time during the term of the trade, but may include other parties involved in, or incidental to, the trade, such as parties acting in the role of novation transferor/transferee, broker, calculation agent, etc. In FpML roles are defined in multiple places within a document.'
  <account> Account </account> [0..*]
    'Optional account information used to precisely define the origination and destination of financial instruments.'
End Group: PartiesAndAccounts.model
</...>

```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="RequestClearingRetracted">
  <xsd:complexContent>
    <xsd:extension base=" NonCorrectableRequestMessage ">
      <xsd:sequence>
        <xsd:group ref=" Events.model " minOccurs="0"/>
        <xsd:group ref=" PartiesAndAccounts.model " minOccurs="0"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

[top](#)**Complex Type: RequestConfirmation**

Super-types:

[CorrectableRequestMessage](#) < **RequestConfirmation** (by extension)

Sub-types:

None

Name	RequestConfirmation
Used by (from the same schema document)	Element <a href="#">requestConfirmation</a>
Abstract	no

**XML Instance Representation**

```

<...
  fpmlVersion=" xsd:token (value comes from list: {'5-0'}) [1]
  'Indicate which version of the FpML Schema an FpML message adheres to.'

  "
  expectedBuild=" xsd:positiveInteger [0..1]
  'This optional attribute can be supplied by a message creator in an FpML instance to
  specify which build number of the schema was used to define the message when it was generated.'

  "
  actualBuild="8 [0..1]
  'The specific build number of this schema version. This attribute is not included in
  an instance document. Instead, it is supplied by the XML parser when the document is
  validated against the FpML schema and indicates the build number of the schema file. Every
  time FpML publishes a change to the schema, validation rules, or examples within a version
  (e.g., version 4.2) the actual build number is incremented. If no changes have been
  made between releases within a version (i.e. from Trial Recommendation to Recommendation)
  the actual build number stays the same.'

  ">
    <header> RequestMessageHeader </header> [1]
    <validation> Validation </validation> [0..*]
    <isCorrection> xsd:boolean </isCorrection> [1]
    'Indicates if this message corrects an earlier request.'

    <correlationId> CorrelationId </correlationId> [1]
    'A qualified identifier used to correlate between messages'

    <sequenceNumber> xsd:positiveInteger </sequenceNumber> [1]
    'A numeric value that can be used to order messages with the same correlation identifier
    from the same sender.'

  Start Group: OnBehalfOf.model [0..1]
    <onBehalfOf> OnBehalfOf </onBehalfOf> [1]
    'Indicates which party (and accounts) a trade is being processed for.'

  End Group: OnBehalfOf.model
  Start Choice [1]
    <trade> Trade </trade> [1]
    <amendment> TradeAmendmentContent </amendment> [1]
    <increase> TradeNotionalChange </increase> [1]
    <termination> TradeNotionalChange </termination> [1]
    <novation> TradeNovationContent </novation> [1]
    <additionalEvent> ... </additionalEvent> [1]
  End Choice
    <party> Party </party> [1..*]
    'A legal entity or a subdivision of a legal entity.', 'Parties can perform multiple roles in
    a trade lifecycle. For example, the principal parties obligated to make payments from time
    to time during the term of the trade, but may include other parties involved in, or
    incidental to, the trade, such as parties acting in the role of novation transferor/

```

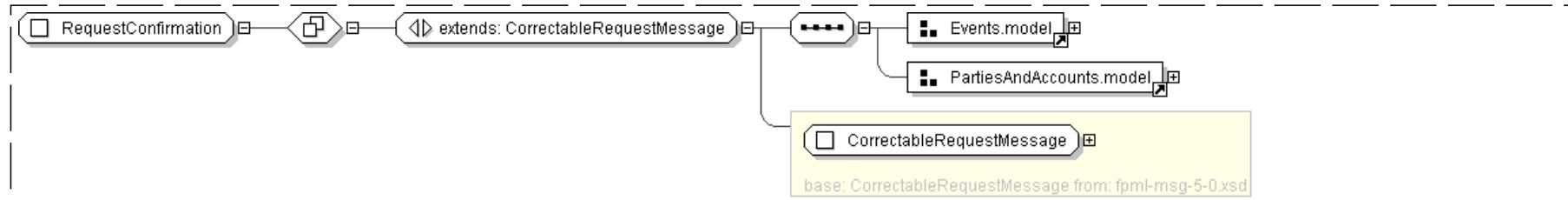
*transferee, broker, calculation agent, etc. In FpML roles are defined in multiple places within a document.'*

<account> Account </account> [0..\*]

'Optional account information used to precisely define the origination and destination of financial instruments.'

</...>

#### Diagram



#### Schema Component Representation

```

<xsd:complexType name="RequestConfirmation">
  <xsd:complexContent>
    <xsd:extension base=" CorrectableRequestMessage ">
      <xsd:sequence>
        <xsd:group ref=" Events.model "/>
        <xsd:group ref=" PartiesAndAccounts.model "/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
  
```

[top](#)

## Complex Type: RequestConsent

Super-types:	<a href="#">CorrectableRequestMessage</a> < <b>RequestConsent</b> (by extension)
--------------	--

Sub-types:	None
------------	------

Name	RequestConsent
------	----------------

Used by (from the same schema document)	Element <a href="#">requestConsent</a>
---	--

Abstract	no
----------	----

#### XML Instance Representation

```

<...
  fpmlVersion= " xsd:token (value comes from list: {'5-0'}) [1]
  'Indicate which version of the FpML Schema an FpML message adheres to.'
  "
  expectedBuild= " xsd:positiveInteger [0..1]
  'This optional attribute can be supplied by a message creator in an FpML instance to
  specify which build number of the schema was used to define the message when it was generated.'
  "
  
```

```
actualBuild="8 [0..1]
```

*'The specific build number of this schema version. This attribute is not included in an instance document. Instead, it is supplied by the XML parser when the document is validated against the FpML schema and indicates the build number of the schema file. Every time FpML publishes a change to the schema, validation rules, or examples within a version (e.g., version 4.2) the actual build number is incremented. If no changes have been made between releases within a version (i.e. from Trial Recommendation to Recommendation) the actual build number stays the same.'*

```
">
  <header> RequestMessageHeader </header> [1]
  <validation> Validation </validation> [0..*]
  <isCorrection> xsd:boolean </isCorrection> [1]
    'Indicates if this message corrects an earlier request.'

  <correlationId> CorrelationId </correlationId> [1]
    'A qualified identifier used to correlate between messages'

  <sequenceNumber> xsd:positiveInteger </sequenceNumber> [1]
    'A numeric value that can be used to order messages with the same correlation identifier from the same sender.'
```

```
Start Group: OnBehalfOf.model [0..1]
  <onBehalfOf> OnBehalfOf </onBehalfOf> [1]
    'Indicates which party (and accounts) a trade is being processed for.'
```

End Group: OnBehalfOf.model

```
Start Choice [1]
  <trade> Trade </trade> [1]
  <amendment> TradeAmendmentContent </amendment> [1]
  <increase> TradeNotionalChange </increase> [1]
  <termination> TradeNotionalChange </termination> [1]
  <novation> TradeNovationContent </novation> [1]
  <additionalEvent> ... </additionalEvent> [1]
```

End Choice

```
<party> Party </party> [1..*]
```

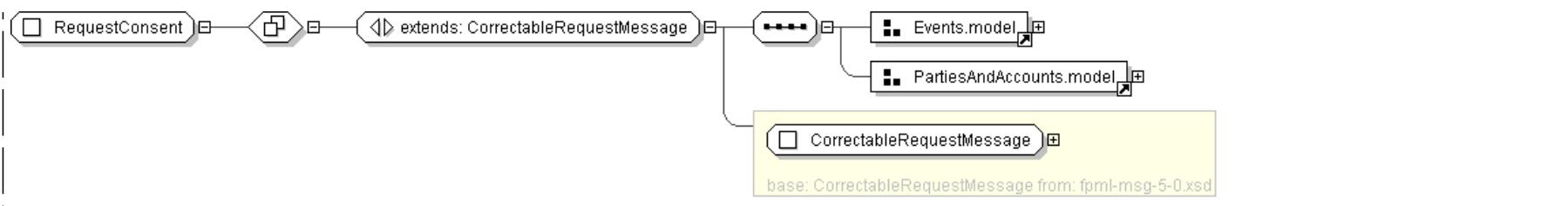
*'A legal entity or a subdivision of a legal entity.', 'Parties can perform multiple roles in a trade lifecycle. For example, the principal parties obligated to make payments from time to time during the term of the trade, but may include other parties involved in, or incidental to, the trade, such as parties acting in the role of novation transferor/transferee, broker, calculation agent, etc. In FpML roles are defined in multiple places within a document.'*

```
<account> Account </account> [0..*]
```

*'Optional account information used to precisely define the origination and destination of financial instruments.'*

```
</...>
```

Diagram



### Schema Component Representation

```

<xsd:complexType name="RequestConsent">
  <xsd:complexContent>
    <xsd:extension base=" CorrectableRequestMessage ">
      <xsd:sequence>
        <xsd:group ref=" Events.model "/>
        <xsd:group ref=" PartiesAndAccounts.model "/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
  
```

[top](#)

### Complex Type: RequestConsentRetracted

<b>Super-types:</b>	<a href="#">NonCorrectableRequestMessage</a> < <b>RequestConsentRetracted</b> (by extension)
<b>Sub-types:</b>	None

**Name** RequestConsentRetracted

**Used by (from the same schema document)** Element [requestConsentRetracted](#)

**Abstract** no

### XML Instance Representation

```

<...
  fpmlVersion=" xsd:token (value comes from list: {'5-0'}) [1]
  'Indicate which version of the FpML Schema an FpML message adheres to.'

  "
  expectedBuild=" xsd:positiveInteger [0..1]
  'This optional attribute can be supplied by a message creator in an FpML instance to
  specify which build number of the schema was used to define the message when it was generated.'

  "
  actualBuild="8 [0..1]
  'The specific build number of this schema version. This attribute is not included in
  an instance document. Instead, it is supplied by the XML parser when the document is
  validated against the FpML schema and indicates the build number of the schema file. Every
  time FpML publishes a change to the schema, validation rules, or examples within a version
  (e.g., version 4.2) the actual build number is incremented. If no changes have been
  made between releases within a version (i.e. from Trial Recommendation to Recommendation)
  the actual build number stays the same.'

  ">
  <header> RequestMessageHeader </header> [1]
  <validation> Validation </validation> [0..*]
  
```

```

<correlationId> CorrelationId </correlationId> [1]
'A qualified identifier used to correlate between messages'

<sequenceNumber> xsd:positiveInteger </sequenceNumber> [1]
'A numeric value that can be used to order messages with the same correlation identifier
from the same sender.'

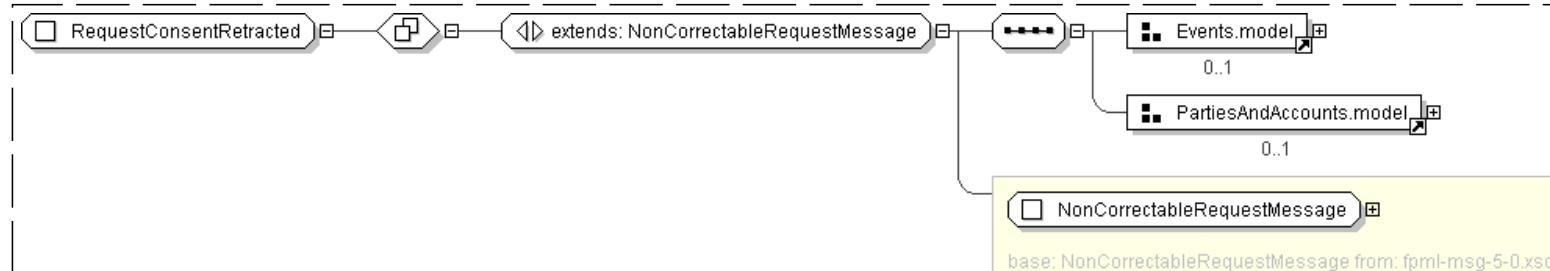
Start Group: OnBehalfOf.model [0..1]
<onBehalfOf> OnBehalfOf </onBehalfOf> [1]
'Indicates which party (and accounts) a trade is being processed for.'

End Group: OnBehalfOf.model
Start Group: Events.model [0..1]
Start Choice [1]
    <trade> Trade </trade> [1]
    <amendment> TradeAmendmentContent </amendment> [1]
    <increase> TradeNotionalChange </increase> [1]
    <termination> TradeNotionalChange </termination> [1]
    <novation> TradeNovationContent </novation> [1]
    <additionalEvent> ... </additionalEvent> [1]
End Choice
End Group: Events.model
Start Group: PartiesAndAccounts.model [0..1]
<party> Party </party> [1..*]
'A legal entity or a subdivision of a legal entity.', 'Parties can perform multiple roles in
a trade lifecycle. For example, the principal parties obligated to make payments from time
to time during the term of the trade, but may include other parties involved in, or
incidental to, the trade, such as parties acting in the role of novation transferor/
transferee, broker, calculation agent, etc. In FpML roles are defined in multiple places
within a document.'

<account> Account </account> [0..*]
'Optional account information used to precisely define the origination and destination
of financial instruments.'

End Group: PartiesAndAccounts.model
</...>

```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="RequestConsentRetracted">
    <xsd:complexContent>
        <xsd:extension base=" NonCorrectableRequestMessage ">
            <xsd:sequence>
                <xsd:group ref=" Events.model " minOccurs="0" />

```

```

<xsd:group ref="PartiesAndAccounts.model" minOccurs="0" />
</xsd:sequence>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>

```

## Complex Type: [TradeChangeAdvice](#)

<b>Super-types:</b>	<a href="#">CorrectableRequestMessage</a> < <b>TradeChangeAdvice</b> (by extension)
<b>Sub-types:</b>	None

<b>Name</b>	TradeChangeAdvice
<b>Used by (from the same schema document)</b>	Element <a href="#">tradeChangeAdvice</a>
<b>Abstract</b>	no
<b>Documentation</b>	Defines the structure for a message indicating that a trade is being changed due to a non-negotiated event.

### XML Instance Representation

```

<...
fpmVersion=" xsd:token (value comes from list: {'5-0'}) [1]
'Indicate which version of the FpML Schema an FpML message adheres to.'

"
expectedBuild=" xsd:positiveInteger [0..1]
'This optional attribute can be supplied by a message creator in an FpML instance to
specify which build number of the schema was used to define the message when it was generated.'

"
actualBuild="8 [0..1]
'The specific build number of this schema version. This attribute is not included in
an instance document. Instead, it is supplied by the XML parser when the document is
validated against the FpML schema and indicates the build number of the schema file. Every
time FpML publishes a change to the schema, validation rules, or examples within a version
(e.g., version 4.2) the actual build number is incremented. If no changes have been
made between releases within a version (i.e. from Trial Recommendation to Recommendation)
the actual build number stays the same.'

">
<header> RequestMessageHeader </header> [1]
<validation> Validation </validation> [0..*]
<isCorrection> xsd:boolean </isCorrection> [1]
'Indicates if this message corrects an earlier request.'

<correlationId> CorrelationId </correlationId> [1]
'A qualified identifier used to correlate between messages'

<sequenceNumber> xsd:positiveInteger </sequenceNumber> [1]
'A numeric value that can be used to order messages with the same correlation identifier
from the same sender.'

Start Group: OnBehalfOf.model [0..1]
<onBehalfOf> OnBehalfOf </onBehalfOf> [1]
'Indicates which party (and accounts) a trade is being processed for.'

```

```

End Group: OnBehalfOf.model
<change> TradeChangeContent </change> [1]
'Describes the details of the change.'

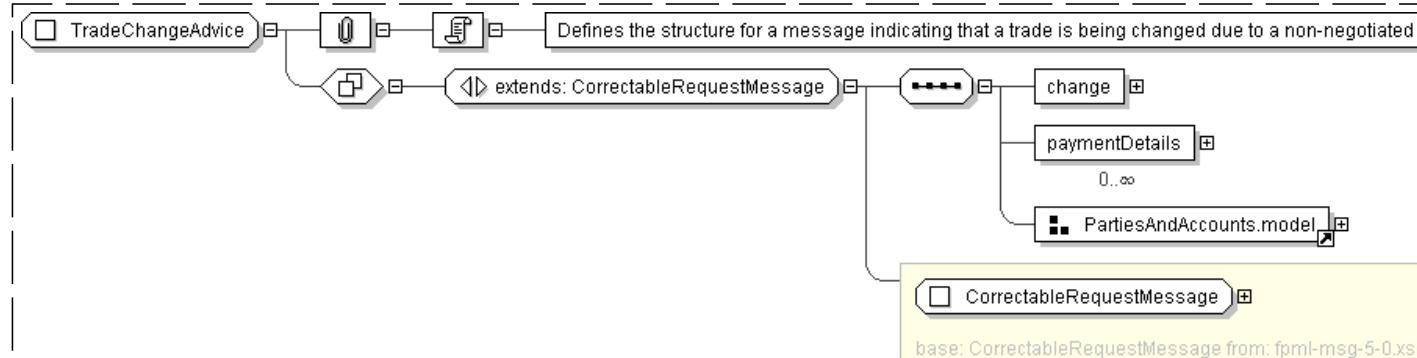
<paymentDetails> PaymentDetails </paymentDetails> [0..*]
'Details of the payments, like amount breakdowns, settlement information.'

<party> Party </party> [1..*]
'A legal entity or a subdivision of a legal entity.', 'Parties can perform multiple roles in
a trade lifecycle. For example, the principal parties obligated to make payments from time
to time during the term of the trade, but may include other parties involved in, or
incidental to, the trade, such as parties acting in the role of novation transferor/
transferee, broker, calculation agent, etc. In FpML roles are defined in multiple places
within a document.'

<account> Account </account> [0..*]
'Optional account information used to precisely define the origination and destination
of financial instruments.'

</...>

```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="TradeChangeAdvice">
  <xsd:complexContent>
    <xsd:extension base=" CorrectableRequestMessage ">
      <xsd:sequence>
        <xsd:element name="change" type=" TradeChangeContent " />
        <xsd:element name="paymentDetails" type=" PaymentDetails " minOccurs="0" maxOccurs="unbounded" />
        <xsd:group ref=" PartiesAndAccounts.model " />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

[top](#)**Complex Type: TradeChangeAdviceRetracted**

Super-types:

[NonCorrectableRequestMessage](#) < **TradeChangeAdviceRetracted** (by extension)

Sub-types:

None

Name	TradeChangeAdviceRetracted
Used by (from the same schema document)	Element <a href="#">tradeChangeAdviceRetracted</a>
Abstract	no
Documentation	Defines the structure for a message retracting a prior change advice.

**XML Instance Representation**

```

<...
  fpmlVersion=" xsd:token (value comes from list: {'5-0'}) [1]
  'Indicate which version of the FpML Schema an FpML message adheres to.'

  "
  expectedBuild=" xsd:positiveInteger [0..1]
  'This optional attribute can be supplied by a message creator in an FpML instance to
  specify which build number of the schema was used to define the message when it was generated.'

  "
  actualBuild="8 [0..1]
  'The specific build number of this schema version. This attribute is not included in
  an instance document. Instead, it is supplied by the XML parser when the document is
  validated against the FpML schema and indicates the build number of the schema file. Every
  time FpML publishes a change to the schema, validation rules, or examples within a version
  (e.g., version 4.2) the actual build number is incremented. If no changes have been
  made between releases within a version (i.e. from Trial Recommendation to Recommendation)
  the actual build number stays the same.'

  ">
  <header> RequestMessageHeader </header> [1]
  <validation> Validation </validation> [0..*]
  <correlationId> CorrelationId </correlationId> [1]
  'A qualified identifier used to correlate between messages'

  <sequenceNumber> xsd:positiveInteger </sequenceNumber> [1]
  'A numeric value that can be used to order messages with the same correlation identifier
  from the same sender.'

Start Group: OnBehalfOf.model [0..1]
  <onBehalfOf> OnBehalfOf </onBehalfOf> [1]
  'Indicates which party (and accounts) a trade is being processed for.'

End Group: OnBehalfOf.model
  <tradeIdentifier> PartyTradeIdentifier </tradeIdentifier> [1..*]
  'The qualified identifiers of the subject trade.'

  <change> TradeChangeContent </change> [0..1]
  'Describes the details of the change being retracted.'

  <paymentDetails> PaymentDetails </paymentDetails> [0..*]
  'Details of the payments, like amount breakdowns, settlement information.'

  <party> Party </party> [1..*]
  'A legal entity or a subdivision of a legal entity.', 'Parties can perform multiple roles in
  a trade lifecycle. For example, the principal parties obligated to make payments from time
  to time during the term of the trade, but may include other parties involved in, or
  incidental to, the trade, such as parties acting in the role of novation transferor/

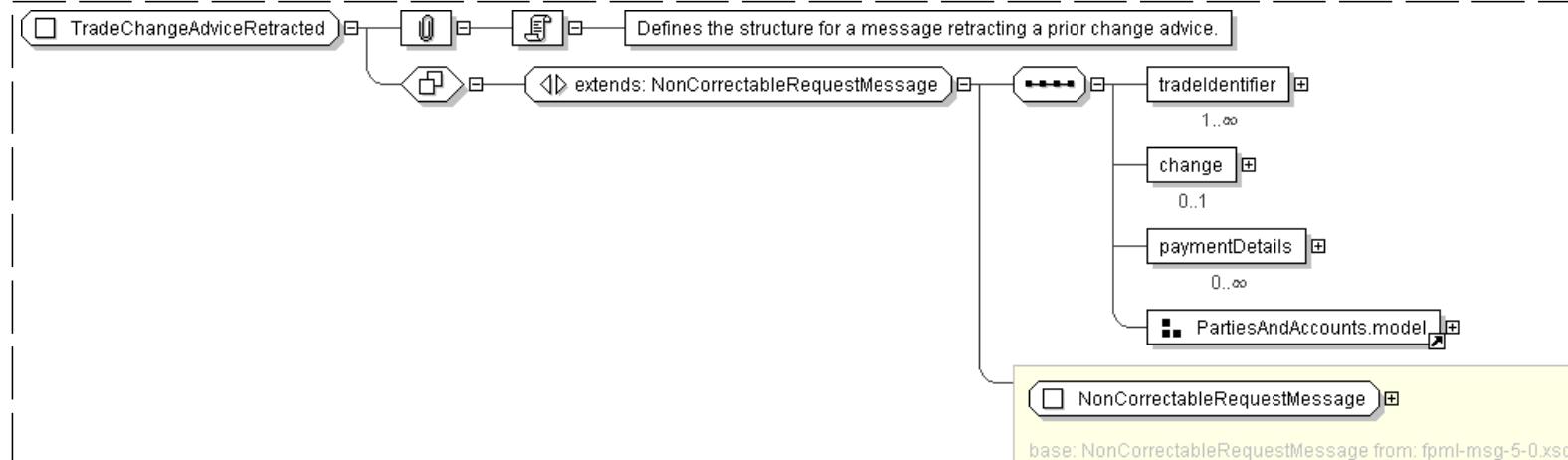
```

*transferee, broker, calculation agent, etc. In FpML roles are defined in multiple places within a document.'*

<account> [Account](#) </account> [0..\*]

'Optional account information used to precisely define the origination and destination of financial instruments.'

&lt;/...&gt;

**Diagram****Schema Component Representation**

```

<xsd:complexType name="TradeChangeAdviceRetracted">
  <xsd:complexContent>
    <xsd:extension base=" NonCorrectableRequestMessage ">
      <xsd:sequence>
        <xsd:element name="tradelIdentifier" type=" PartyTradeIdentifier " maxOccurs="unbounded"/>
        <xsd:element name="change" type=" TradeChangeContent " minOccurs="0"/>
        <xsd:element name="paymentDetails" type=" PaymentDetails " minOccurs="0" maxOccurs="unbounded"/>
        <xsd:group ref=" PartiesAndAccounts.model "/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
  
```

top

**Legend**

**Complex Type:**  
Schema Component Type

**AusAddress**  
Schema Component Name

**Super-types:**  
**Sub-types:**

[Address](#) < AusAddress (by extension)  
 • [QLDAddress](#) (by restriction)

If this schema component is a type definition, its type hierarchy is shown in a gray-bordered box.

<b>Name</b>	AusAddress
<b>Abstract</b>	no

The table above displays the properties of this schema component.

#### XML Instance Representation

```
<... country="Australia" >
<unitNo> string </unitNo> [0..1]
<houseNo> string </houseNo> [1]
<street> string </street> [1]
Start Choice [1]
<city> string </city> [1]
<town> string </town> [1]
End Choice
<state> AusStates </state> [1]
<postcode> string <><pattern = [1-9][0-9]{3}></postcode> [1]
</...>
```

The XML Instance Representation table above shows the schema component's content as an XML instance.

- The minimum and maximum occurrence of elements and attributes are provided in square brackets, e.g. [0..1].
- Model group information are shown in gray, e.g. Start Choice ... End Choice.
- For type derivations, the elements and attributes that have been added to or changed from the base type's content are shown in **bold**.
- If an element/attribute has a fixed value, the fixed value is shown in green, e.g. country="Australia".
- Otherwise, the type of the element/attribute is displayed.
  - If the element/attribute's type is in the schema, a link is provided to it.
  - For local simple type definitions, the constraints are displayed in angle brackets, e.g. <><pattern = [1-9][0-9]{3}></>.

#### Schema Component Representation

```
<complexType name="AusAddress">
<complexContent>
<extension base=" Address ">
<sequence>
<element name="state" type=" AusStates " />
<element name="postcode">
<simpleType>
<restriction base=" string ">
<pattern value="[1-9][0-9]{3}" />
</restriction>
</simpleType>
</element>
</sequence>
<attribute name="country" type=" string " fixed="Australia" />
</extension>
</complexContent>
</complexType>
```

The Schema Component Representation table above displays the underlying XML representation of the schema component. (Annotations are not shown.)

[top](#)

## Glossary

**Abstract** (Applies to complex type definitions and element declarations). An abstract element or complex type cannot be used to validate an element instance. If there is a reference to an abstract element, only element declarations that can substitute the abstract element can be used to validate the instance. For references to abstract type definitions, only derived types can be used.

**All Model Group** Child elements can be provided *in any order* in instances. See: <http://www.w3.org/TR/xmlschema-1/#element-all>.

**Choice Model Group** Only one from the list of child elements and model groups can be provided in instances. See: <http://www.w3.org/TR/xmlschema-1/#element-choice>.

**Collapse Whitespace Policy** Replace tab, line feed, and carriage return characters with space character (Unicode character 32). Then, collapse contiguous sequences of space characters into single space character, and remove leading and trailing space characters.

**Disallowed Substitutions** (Applies to element declarations). If *substitution* is specified, then *substitution group* members cannot be used in place of the given element declaration to validate element instances. If *derivation methods*, e.g. extension, restriction, are specified, then the given element declaration will not validate element instances that have types derived from the element declaration's type using the specified derivation methods. Normally, element instances can override their declaration's type by specifying an *xsi:type* attribute.

**Key Constraint** Like *Uniqueness Constraint*, but additionally requires that the specified value(s) must be provided. See: [http://www.w3.org/TR/xmlschema-1/#cIdentity-constraint\\_Definitions](http://www.w3.org/TR/xmlschema-1/#cIdentity-constraint_Definitions).

**Key Reference Constraint** Ensures that the specified value(s) must match value(s) from a *Key Constraint* or *Uniqueness Constraint*. See: [http://www.w3.org/TR/xmlschema-1/#cIdentity-constraint\\_Definitions](http://www.w3.org/TR/xmlschema-1/#cIdentity-constraint_Definitions).

**Model Group** Groups together element content, specifying the order in which the element content can occur and the number of times the group of element content may be repeated. See: [http://www.w3.org/TR/xmlschema-1/#Model\\_Groups](http://www.w3.org/TR/xmlschema-1/#Model_Groups).

**Nillable** (Applies to element declarations). If an element declaration is nillable, instances can use the *xsi:nil* attribute. The *xsi:nil* attribute is the boolean attribute, *nil*, from the <http://www.w3.org/2001/XMLSchema-instance> namespace. If an element instance has an *xsi:nil* attribute set to true, it can be left empty, even though its element declaration may have required content.

**Notation** A notation is used to identify the format of a piece of data. Values of elements and attributes that are of type, NOTATION, must come from the names of declared notations. See: [http://www.w3.org/TR/xmlschema-1/#cNotation\\_Declarations](http://www.w3.org/TR/xmlschema-1/#cNotation_Declarations).

**Preserve Whitespace Policy** Preserve whitespaces exactly as they appear in instances.

**Prohibited Derivations** (Applies to type definitions). Derivation methods that cannot be used to create sub-types from a given type definition.

**Prohibited Substitutions** (Applies to complex type definitions). Prevents sub-types that have been derived using the specified derivation methods from validating element instances in place of the given type definition.

**Replace Whitespace Policy** Replace tab, line feed, and carriage return characters with space character (Unicode character 32).

**Sequence Model Group** Child elements and model groups must be provided *in the specified order* in instances. See: <http://www.w3.org/TR/xmlschema-1/#element-sequence>.

**Substitution Group** Elements that are *members* of a substitution group can be used wherever the *head* element of the substitution group is referenced.

**Substitution Group Exclusions** (Applies to element declarations). Prohibits element declarations from nominating themselves as being able to substitute a given element declaration, if they have types that are derived from the original element's type using the specified derivation methods.

**Target Namespace** The target namespace identifies the namespace that components in this schema belongs to. If no target namespace is provided, then the schema components do not belong to any namespace.

**Uniqueness Constraint** Ensures uniqueness of an element/attribute value, or a combination of values, within a specified scope. See: [http://www.w3.org/TR/xmlschema-1/#cIdentity-constraint\\_Definitions](http://www.w3.org/TR/xmlschema-1/#cIdentity-constraint_Definitions).



# XML Schema Documentation

## Table of Contents

- [Schema Document Properties](#)
- [Global Declarations](#)
  - [Element: correlationSwap](#)
- [Global Definitions](#)
  - Complex Type: [CorrelationAmount](#)
  - Complex Type: [CorrelationLeg](#)
  - Complex Type: [CorrelationSwap](#)
- [Legend](#)
- [Glossary](#)

[top](#)

## Schema Document Properties

<b>Target Namespace</b>	<a href="http://www.fpml.org/FpML-5/confirmation">http://www.fpml.org/FpML-5/confirmation</a>
<b>Version</b>	\$Revision: 2527 \$
<b>Element and Attribute Namespaces</b>	<ul style="list-style-type: none"> <li>• Global element and attribute declarations belong to this schema's target namespace.</li> <li>• By default, local element declarations belong to this schema's target namespace.</li> <li>• By default, local attribute declarations have no namespace.</li> </ul>
<b>Schema Composition</b>	<ul style="list-style-type: none"> <li>• This schema includes components from the following schema document(s):           <ul style="list-style-type: none"> <li>◦ <a href="#">fpml-eq-shared-5-0.xsd</a></li> </ul> </li> </ul>

## Declared Namespaces

Prefix	Namespace
Default namespace	<a href="http://www.fpml.org/FpML-5/confirmation">http://www.fpml.org/FpML-5/confirmation</a>
xml	<a href="http://www.w3.org/XML/1998/namespace">http://www.w3.org/XML/1998/namespace</a>
fpml-annotation	<a href="http://www.fpml.org/annotation">http://www.fpml.org/annotation</a>
xsd	<a href="http://www.w3.org/2001/XMLSchema">http://www.w3.org/2001/XMLSchema</a>
ecore	<a href="http://www.eclipse.org/emf/2002/Ecore">http://www.eclipse.org/emf/2002/Ecore</a>

## Schema Component Representation

```

<xsd:schema targetNamespace="http://www.fpml.org/FpML-5/confirmation"
  documentRoot="FpML" nsPrefix="conf" package="org.fpml.confirmation" version="$Revision: 2527 "
  elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xsd:include schemaLocation="fpml-eq-shared-5-0.xsd"/>
  ...
</xsd:schema>

```

[top](#)

## Global Declarations

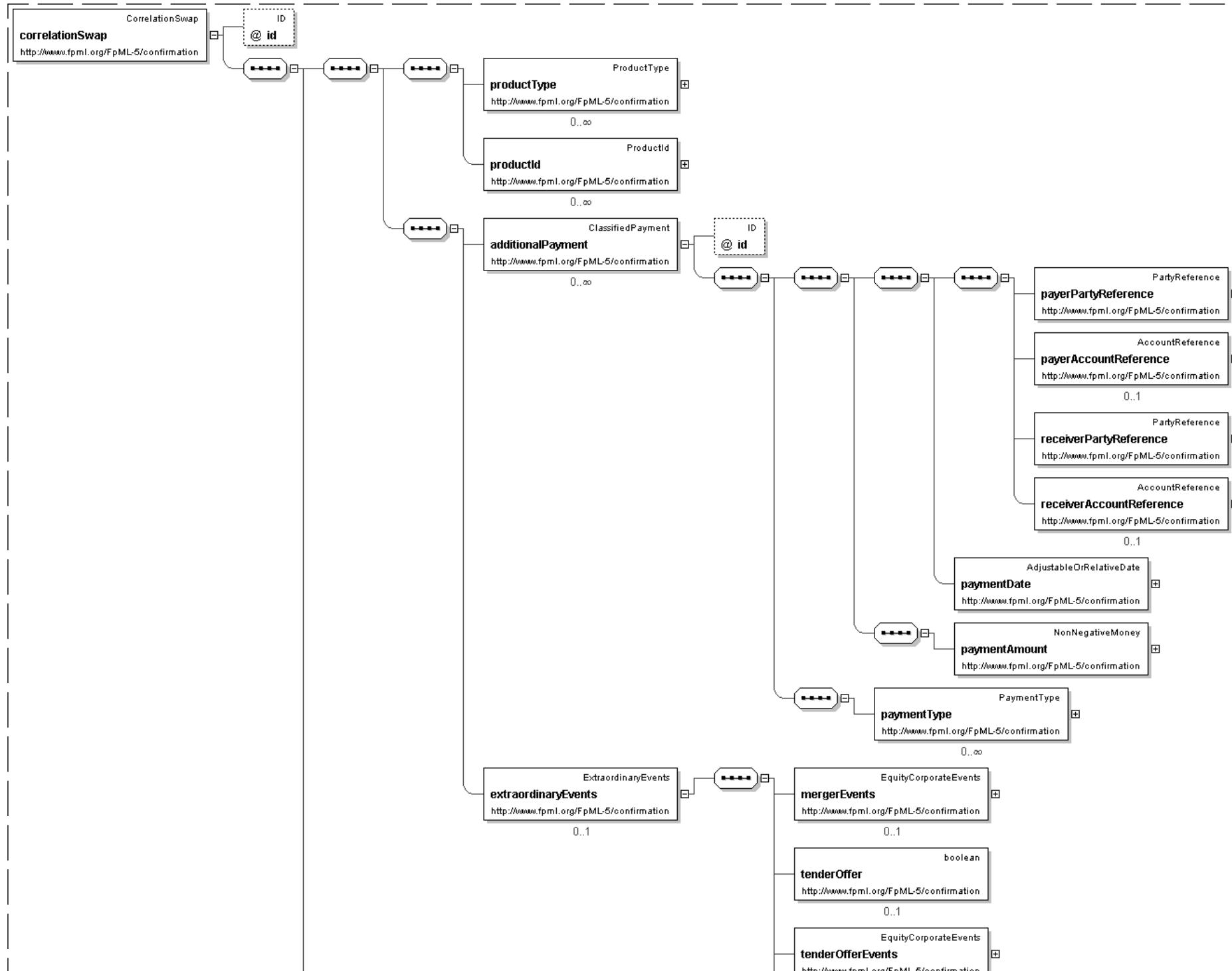
### Element: correlationSwap

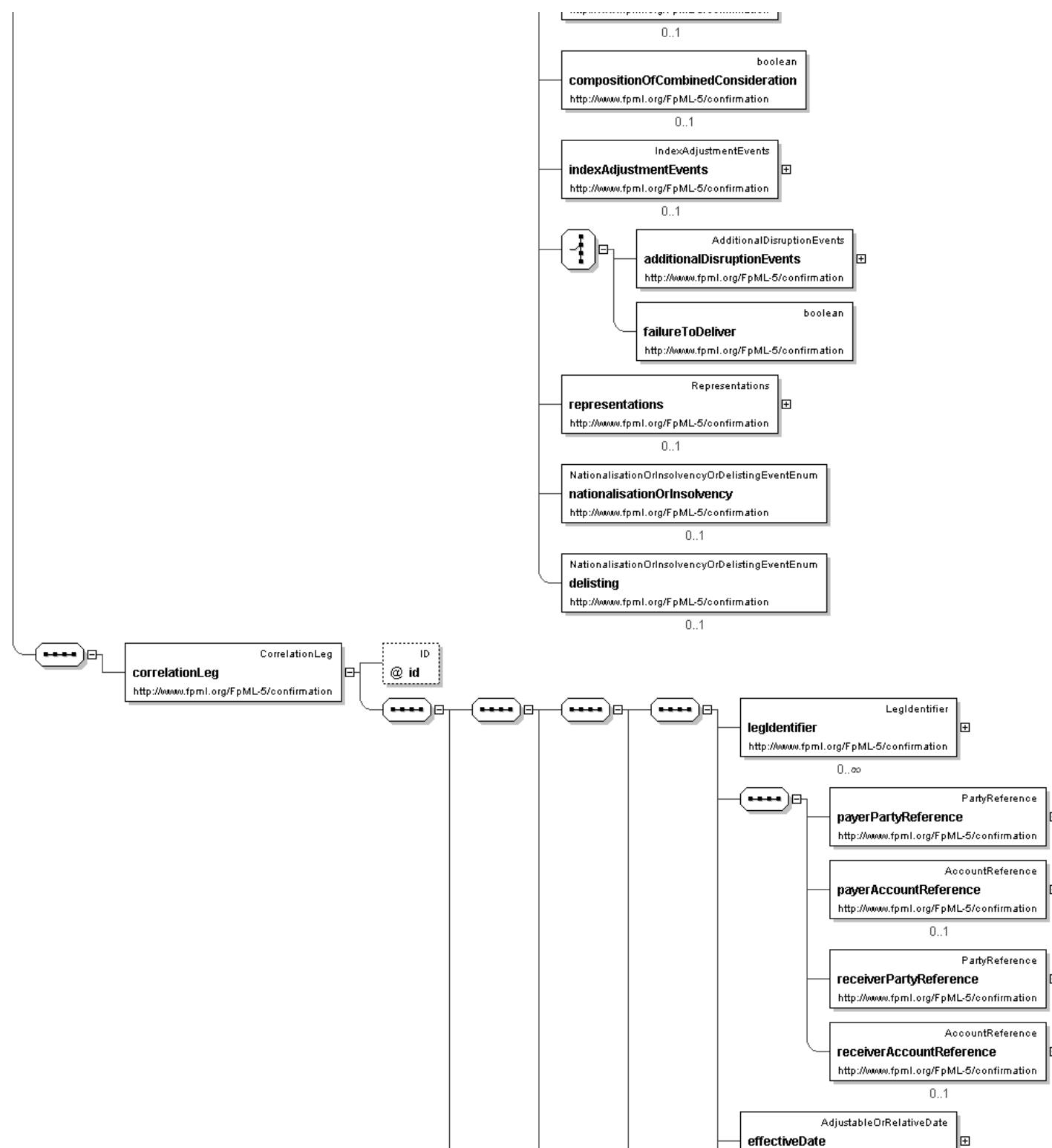
- This element can be used wherever the following element is referenced:
  - [product](#)

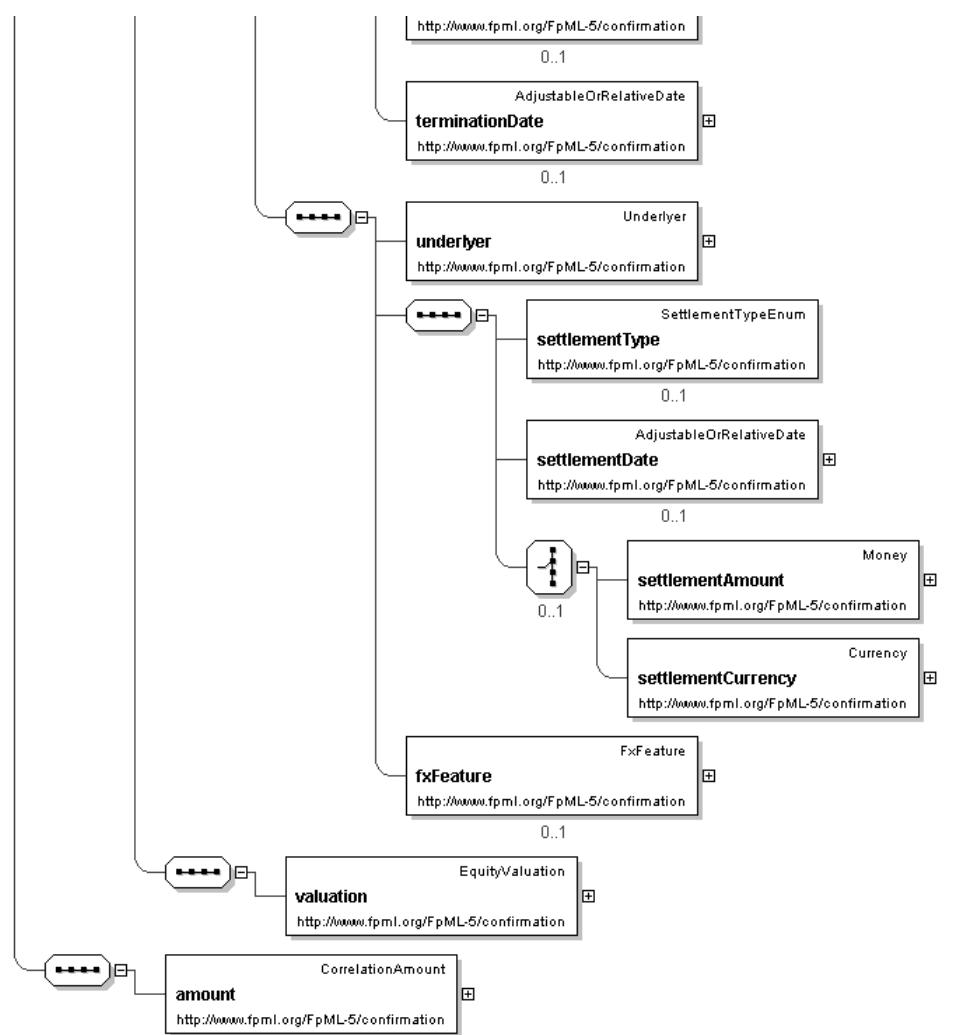
<b>Name</b>	correlationSwap
<b>Type</b>	<a href="#">CorrelationSwap</a>
<b>Nillable</b>	no
<b>Abstract</b>	no

**Documentation**

Specifies the structure of a correlation swap.

**Logical Diagram**



**XML Instance Representation**

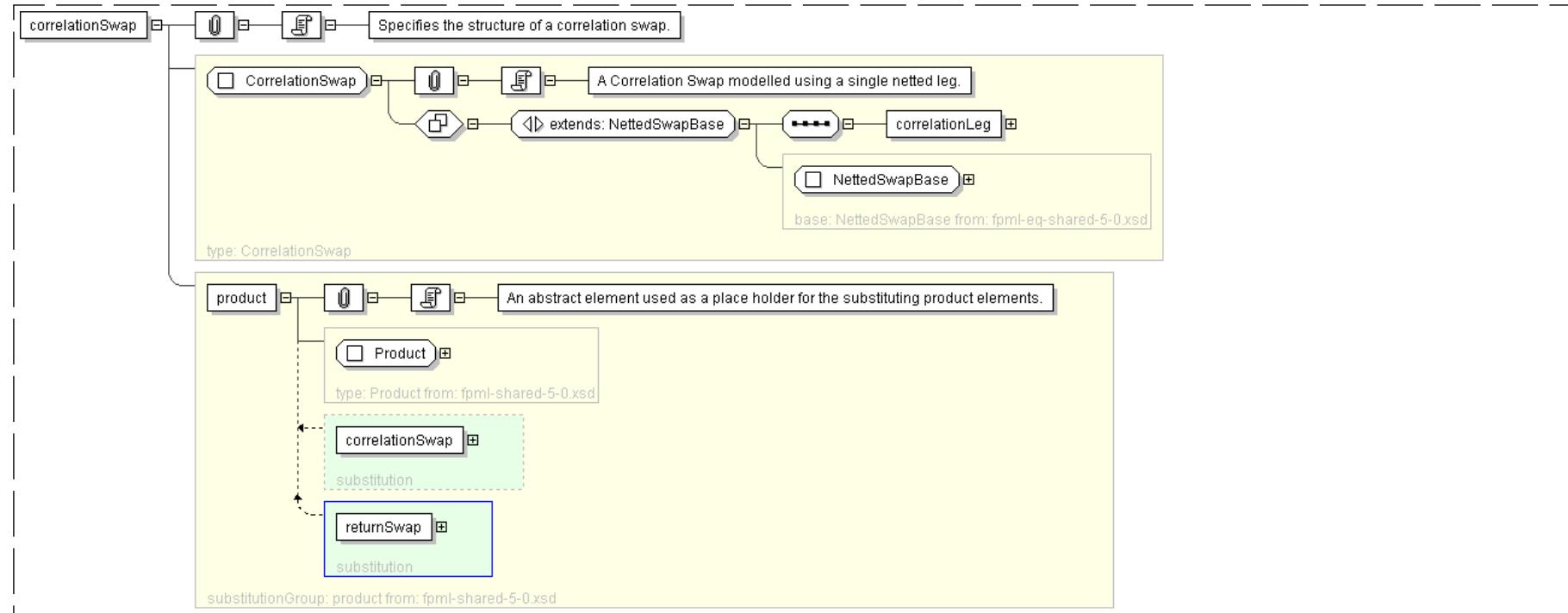
```

<correlationSwap
  id="xsd:ID [0..1]">
  <productType> ProductType </productType> [0..*]
  'A classification of the type of product. FpML defines a simple product categorization using
  a coding scheme.'
  <productId> ProductId </productId> [0..*]
  'A product 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..'
  <additionalPayment> ClassifiedPayment </additionalPayment> [0..*]
  'Specifies additional payment(s) between the principal parties to the netted swap.'
  <extraordinaryEvents> ExtraordinaryEvents </extraordinaryEvents> [0..1]
  'Where the underlying is shares, specifies events affecting the issuer of those shares that
  may require the terms of the transaction to be adjusted.'

```

```
<correlationLeg> CorrelationLeg </correlationLeg> [1]
'Correlation Leg. Correlation Buyer is deemed to be the Equity Amount Receiver,
Correlation Seller is deemed to be the Equity Amount Payer.'
```

&lt;/correlationSwap&gt;

**Diagram****Schema Component Representation**

```
<xsd:element name="correlationSwap" type="CorrelationSwap" substitutionGroup="product"/>
```

top

**Global Definitions****Complex Type: CorrelationAmount**

Super-types:	<a href="#">CalculatedAmount</a> < <b>CorrelationAmount</b> (by extension)
--------------	--

Sub-types:	None
------------	------

Name	CorrelationAmount
Used by (from the same schema document)	Complex Type <a href="#">CorrelationLeg</a>
Abstract	no
Documentation	Correlation Amount.

**XML Instance Representation**

```
<...>
<calculationDates> AdjustableRelativeOrPeriodicDates </calculationDates> [0..1]
'Specifies the date on which a calculation or an observation will be performed for the
purpose of calculating the amount.'
```

```

<observationStartDate> AdjustableOrRelativeDate </observationStartDate> [0..1]
'The start of the period over which observations are made which are used in the
calculation Used when the observation start date differs from the trade date such as
for forward starting swaps.'

<optionsExchangeDividends> xsd:boolean </optionsExchangeDividends> [0..1]
'If present and true, then options exchange dividends are applicable.'

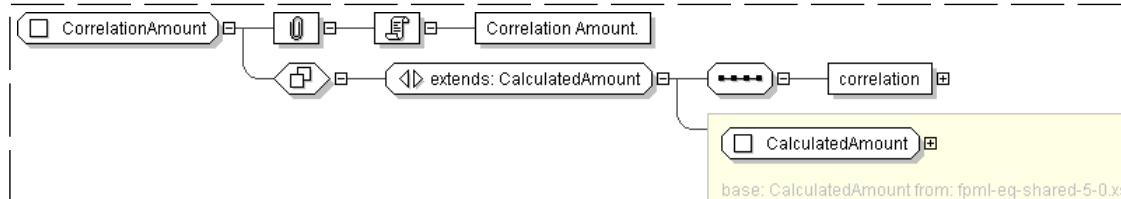
<additionalDividends> xsd:boolean </additionalDividends> [0..1]
'If present and true, then additional dividends are applicable.'

<allDividends> xsd:boolean </allDividends> [0..1]
'Represents the European Master Confirmation value of \'All Dividends\' which, when
applicable, signifies that, for a given Ex-Date, the daily observed Share Price for that day
is adjusted (reduced) by the cash dividend and/or the cash value of any non cash dividend
per Share (including Extraordinary Dividends) declared by the Issuer.'

<correlation> Correlation </correlation> [1]
'Specifies Correlation.'

<!-->

```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="CorrelationAmount">
  <xsd:complexContent>
    <xsd:extension base=" CalculatedAmount ">
      <xsd:sequence>
        <xsd:element name="correlation" type=" Correlation " />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

top

**Complex Type: CorrelationLeg**

<b>Super-types:</b>	DirectionalLegUnderlyerValuation < <b>CorrelationLeg</b> (by extension)
<b>Sub-types:</b>	None
<b>Name</b>	CorrelationLeg
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">CorrelationSwap</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type describing return which is driven by a Correlation calculation.

**XML Instance Representation**

```

<...
  id=" xsd:ID [0..1]">
  <legIdentifier> LegIdentifier </legIdentifier> [0..*]

```

```
'Version aware identification of this leg.'

<payerPartyReference> PartyReference </payerPartyReference> [1]
'A reference to the party responsible for making the payments defined by this structure.'

<payerAccountReference> AccountReference </payerAccountReference> [0..1]
'A reference to the account responsible for making the payments defined by this structure.'

<receiverPartyReference> PartyReference </receiverPartyReference> [1]
'A reference to the party that receives the payments corresponding to this structure.'

<receiverAccountReference> AccountReference </receiverAccountReference> [0..1]
'A reference to the account that receives the payments corresponding to this structure.'

<effectiveDate> AdjustableOrRelativeDate </effectiveDate> [0..1]
'Specifies the effective date of this leg of the swap. When defined in relation to a
date specified somewhere else in the document (through the relativeDate component),
this element will typically point to the effective date of the other leg of the swap.'

<terminationDate> AdjustableOrRelativeDate </terminationDate> [0..1]
'Specifies the termination date of this leg of the swap. When defined in relation to a
date specified somewhere else in the document (through the relativeDate component),
this element will typically point to the termination date of the other leg of the swap.'

<underlyer> Underlyer </underlyer> [1]
'Specifies the underlyer of the leg.'

<settlementType> SettlementTypeEnum </settlementType> [0..1]
<settlementDate> AdjustableOrRelativeDate </settlementDate> [0..1]
Start Group: SettlementAmountOrCurrency.model [0..1]
Start Choice [1]
    <settlementAmount> Money </settlementAmount> [1]
        'Settlement Amount'

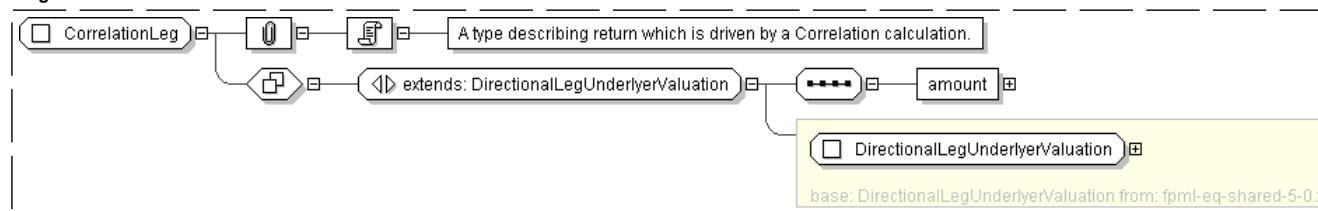
        <settlementCurrency> Currency </settlementCurrency> [1]
        'Settlement Currency for use where the Settlement Amount cannot be known in advance'

End Choice
End Group: SettlementAmountOrCurrency.model
<fxFeature> FxFeature </fxFeature> [0..1]
'Quanto, Composite, or Cross Currency FX features.'

<valuation> EquityValuation </valuation> [1]
'Valuation of the underlyer.'

<amount> CorrelationAmount </amount> [1]
'Specifies, in relation to each Equity Payment Date, the Equity Amount to which the
Equity Payment Date relates. Unless otherwise specified, this term has the meaning defined
in the ISDA 2002 Equity Derivatives Definitions.'

<!-->
```

**Diagram**

**Schema Component Representation**

```
<xsd:complexType name="CorrelationLeg">
  <xsd:complexContent>
    <xsd:extension base=" DirectionalLegUnderlyerValuation ">
      <xsd:sequence>
        <xsd:element name="amount" type=" CorrelationAmount " />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

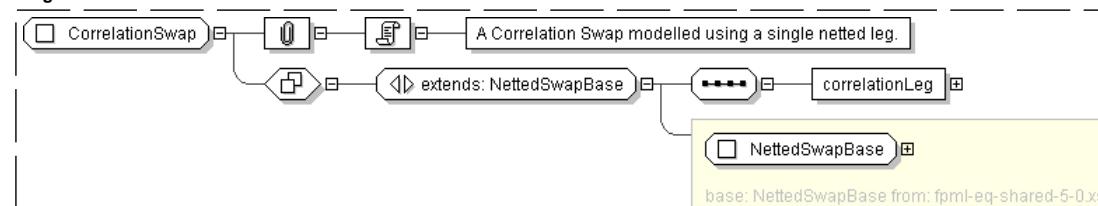
[top](#)**Complex Type: CorrelationSwap**

<b>Super-types:</b>	<a href="#">NettedSwapBase</a> < <b>CorrelationSwap</b> (by extension)
<b>Sub-types:</b>	None

<b>Name</b>	CorrelationSwap
<b>Used by (from the same schema document)</b>	Element <a href="#">correlationSwap</a>
<b>Abstract</b>	no
<b>Documentation</b>	A Correlation Swap modelled using a single netted leg.

**XML Instance Representation**

```
<...>
<id=" xsd:ID [0..1]">
  <productType> ProductType </productType> [0..*]
  'A classification of the type of product. FpML defines a simple product categorization using
  a coding scheme.'
<productId> ProductId </productId> [0..*]
  'A product 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.'
<additionalPayment> ClassifiedPayment </additionalPayment> [0..*]
  'Specifies additional payment(s) between the principal parties to the netted swap.'
<extraordinaryEvents> ExtraordinaryEvents </extraordinaryEvents> [0..1]
  'Where the underlying is shares, specifies events affecting the issuer of those shares that
  may require the terms of the transaction to be adjusted.'
<correlationLeg> CorrelationLeg </correlationLeg> [1]
  'Correlation Leg. Correlation Buyer is deemed to be the Equity Amount Receiver,
  Correlation Seller is deemed to be the Equity Amount Payer.'
</...>
```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="CorrelationSwap">
  <xsd:complexContent>
    <xsd:extension base=" NettedSwapBase ">
      <xsd:sequence>
        <xsd:element name="correlationLeg" type=" CorrelationLeg " />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

## Legend

**Complex Type:**  
Schema Component Type

**AusAddress**  
Schema Component Name

<b>Super-types:</b>	<a href="#">Address</a> < AusAddress (by extension)
<b>Sub-types:</b>	• <a href="#">OLDAddress</a> (by restriction)

If this schema component is a type definition, its type hierarchy is shown in a gray-bordered box.

<b>Name</b>	AusAddress
<b>Abstract</b>	no

The table above displays the properties of this schema component.

### XML Instance Representation

```

<... country="Australia" >
<unitNo> string </unitNo> [0..1]
<houseNo> string </houseNo> [1]
<street> string </street> [1]
Start Choice [1]
<city> string </city> [1]
<town> string </town> [1]
End Choice
<state> AusStates </state> [1]
<postcode> string <><pattern = "[1-9][0-9]{3}></postcode> [1]
</...>

```

The XML Instance Representation table above shows the schema component's content as an XML instance.

- The minimum and maximum occurrence of elements and attributes are provided in square brackets, e.g. [0..1].
- Model group information are shown in gray, e.g. Start Choice ... End Choice.
- For type derivations, the elements and attributes that have been added to or changed from the base type's content are shown in **bold**.
- If an element/attribute has a fixed value, the fixed value is shown in green, e.g. country="Australia".
- Otherwise, the type of the element/attribute is displayed.
  - If the element/attribute's type is in the schema, a link is provided to it.
  - For local simple type definitions, the constraints are displayed in angle brackets, e.g. <><pattern = "[1-9][0-9]{3}></>.

### Schema Component Representation

```

<complexType name="AusAddress">
  <complexContent>
    <extension base=" Address ">
      <sequence>
        <element name="state" type=" AusStates " />
        <element name="postcode">
          <simpleType>
            <restriction base=" string ">
              <pattern value=" [1-9][0-9]{3} " />
            </restriction>
          </simpleType>
        </element>
      </sequence>
    </extension>
  </complexContent>
</complexType>

```

```
</element>
<sequence>
<attribute name="country" type=" string " fixed="Australia"/>
</extension>
</complexContent>
</complexType>
```

The Schema Component Representation table above displays the underlying XML representation of the schema component. (Annotations are not shown.)

[top](#)

## Glossary

**Abstract** (Applies to complex type definitions and element declarations). An abstract element or complex type cannot be used to validate an element instance. If there is a reference to an abstract element, only element declarations that can substitute the abstract element can be used to validate the instance. For references to abstract type definitions, only derived types can be used.

**All Model Group** Child elements can be provided *in any order* in instances. See: <http://www.w3.org/TR/xmlschema-1/#element-all>.

**Choice Model Group** Only one from the list of child elements and model groups can be provided in instances. See: <http://www.w3.org/TR/xmlschema-1/#element-choice>.

**Collapse Whitespace Policy** Replace tab, line feed, and carriage return characters with space character (Unicode character 32). Then, collapse contiguous sequences of space characters into single space character, and remove leading and trailing space characters.

**Disallowed Substitutions** (Applies to element declarations). If *substitution* is specified, then *substitution group* members cannot be used in place of the given element declaration to validate element instances. If *derivation methods*, e.g. extension, restriction, are specified, then the given element declaration will not validate element instances that have types derived from the element declaration's type using the specified derivation methods. Normally, element instances can override their declaration's type by specifying an *xsi:type* attribute.

**Key Constraint** Like [Uniqueness Constraint](#), but additionally requires that the specified value(s) must be provided. See: [http://www.w3.org/TR/xmlschema-1/#cIdentity-constraint\\_Definitions](http://www.w3.org/TR/xmlschema-1/#cIdentity-constraint_Definitions).

**Key Reference Constraint** Ensures that the specified value(s) must match value(s) from a [Key Constraint](#) or [Uniqueness Constraint](#). See: [http://www.w3.org/TR/xmlschema-1/#cIdentity-constraint\\_Definitions](http://www.w3.org/TR/xmlschema-1/#cIdentity-constraint_Definitions).

**Model Group** Groups together element content, specifying the order in which the element content can occur and the number of times the group of element content may be repeated. See: [http://www.w3.org/TR/xmlschema-1/#Model\\_Groups](http://www.w3.org/TR/xmlschema-1/#Model_Groups).

**Nullable** (Applies to element declarations). If an element declaration is nullable, instances can use the *xsi:nil* attribute. The *xsi:nil* attribute is the boolean attribute, *nil*, from the <http://www.w3.org/2001/XMLSchema-instance> namespace. If an element instance has an *xsi:nil* attribute set to true, it can be left empty, even though its element declaration may have required content.

**Notation** A notation is used to identify the format of a piece of data. Values of elements and attributes that are of type, NOTATION, must come from the names of declared notations. See: [http://www.w3.org/TR/xmlschema-1/#cNotation\\_Declarations](http://www.w3.org/TR/xmlschema-1/#cNotation_Declarations).

**Preserve Whitespace Policy** Preserve whitespaces exactly as they appear in instances.

**Prohibited Derivations** (Applies to type definitions). Derivation methods that cannot be used to create sub-types from a given type definition.

**Prohibited Substitutions** (Applies to complex type definitions). Prevents sub-types that have been derived using the specified derivation methods from validating element instances in place of the given type definition.

**Replace Whitespace Policy** Replace tab, line feed, and carriage return characters with space character (Unicode character 32).

**Sequence Model Group** Child elements and model groups must be provided *in the specified order* in instances. See: <http://www.w3.org/TR/xmlschema-1/#element-sequence>.

**Substitution Group** Elements that are *members* of a substitution group can be used wherever the *head* element of the substitution group is referenced.

**Substitution Group Exclusions** (Applies to element declarations). Prohibits element declarations from nominating themselves as being able to substitute a given element declaration, if they have types that are derived from the original element's type using the specified derivation methods.

**Target Namespace** The target namespace identifies the namespace that components in this schema belongs to. If no target namespace is provided, then the schema components do not belong to any namespace.

**Uniqueness Constraint** Ensures uniqueness of an element/attribute value, or a combination of values, within a specified scope. See: [http://www.w3.org/TR/xmlschema-1/#cIdentity-constraint\\_Definitions](http://www.w3.org/TR/xmlschema-1/#cIdentity-constraint_Definitions).

[top](#)

---

Generated by [eXoYgen](#)/> XML Editor using a modified version of [xs3p](#) that adds schema diagrams and chunking support.

# XML Schema Documentation

## Table of Contents

- [Schema Document Properties](#)
- [Global Declarations](#)
  - Element: [dividendSwapTransactionSupplement](#)
- [Global Definitions](#)
  - Complex Type: [DividendLeg](#)
  - Complex Type: [DividendPeriodPayment](#)
  - Complex Type: [DividendSwapTransactionSupplement](#)
  - Complex Type: [FixedPaymentAmount](#)
  - Complex Type: [FixedPaymentLeg](#)
- [Legend](#)
- [Glossary](#)

[top](#)

## Schema Document Properties

<b>Target Namespace</b>	<a href="http://www.fpml.org/FpML-5/confirmation">http://www.fpml.org/FpML-5/confirmation</a>
<b>Version</b>	\$Revision: 2527 \$
<b>Element and Attribute Namespaces</b>	<ul style="list-style-type: none"> <li>• Global element and attribute declarations belong to this schema's target namespace.</li> <li>• By default, local element declarations belong to this schema's target namespace.</li> <li>• By default, local attribute declarations have no namespace.</li> </ul>
<b>Schema Composition</b>	<ul style="list-style-type: none"> <li>• This schema includes components from the following schema document(s):           <ul style="list-style-type: none"> <li>◦ <a href="#">fpml-eq-shared-5-0.xsd</a></li> <li>◦ <a href="#">fpml-shared-5-0.xsd</a></li> </ul> </li> </ul>

## Declared Namespaces

Prefix	Namespace
Default namespace	<a href="http://www.fpml.org/FpML-5/confirmation">http://www.fpml.org/FpML-5/confirmation</a>
xml	http://www.w3.org/XML/1998/namespace
fpml-annotation	http://www.fpml.org/annotation
xsd	http://www.w3.org/2001/XMLSchema
ecore	http://www.eclipse.org/emf/2002/Ecore

## Schema Component Representation

```

<xsd:schema targetNamespace="http://www.fpml.org/FpML-5/confirmation"
documentRoot="FpML" nsPrefix="conf" package="org.fpml.confirmation" version="$Revision: 2527 "
$" elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xsd:include schemaLocation="fpml-eq-shared-5-0.xsd"/>
  <xsd:include schemaLocation="fpml-shared-5-0.xsd"/>
  ...
</xsd:schema>

```

[top](#)

## Global Declarations

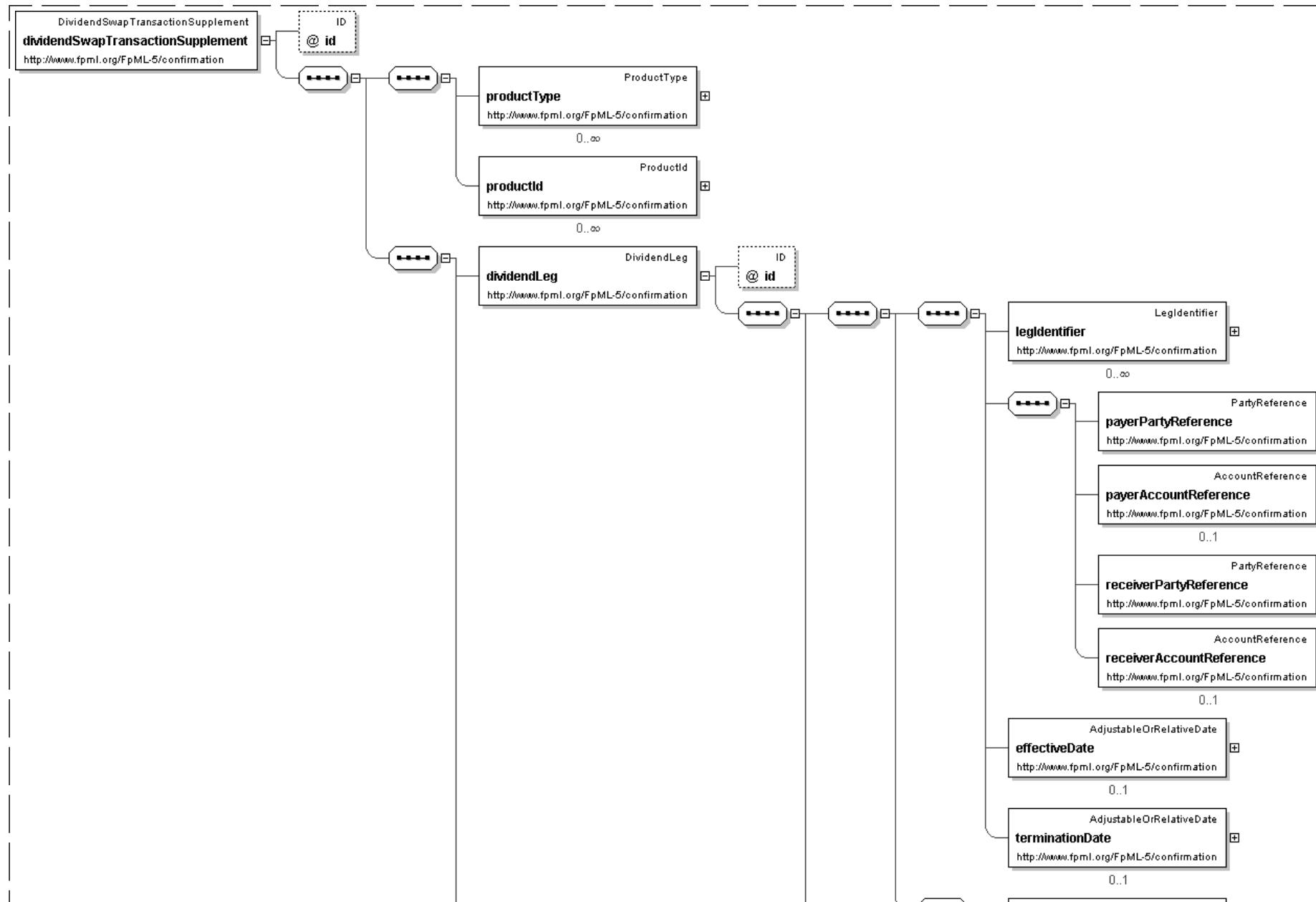
### Element: dividendSwapTransactionSupplement

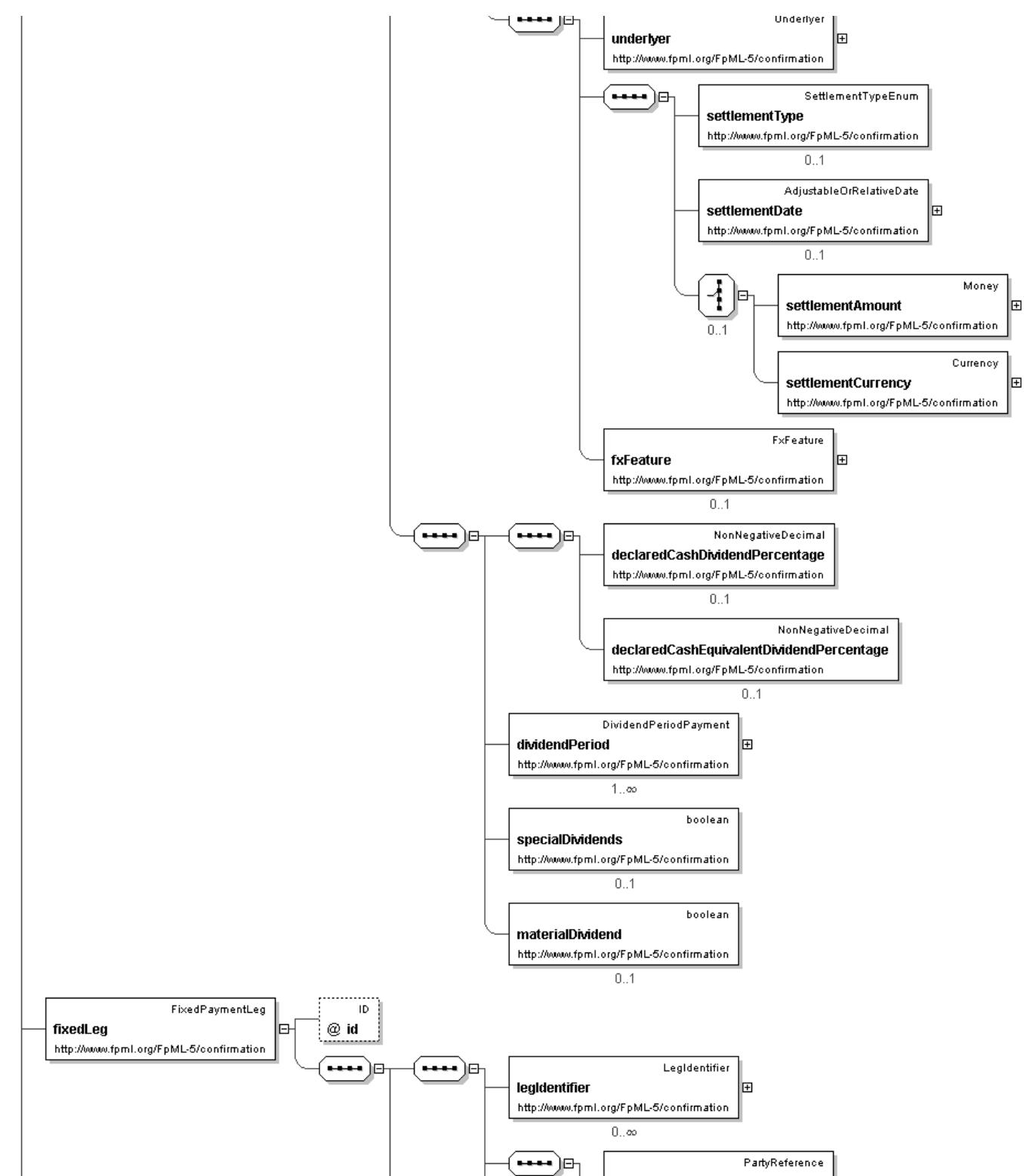
- This element can be used wherever the following element is referenced:

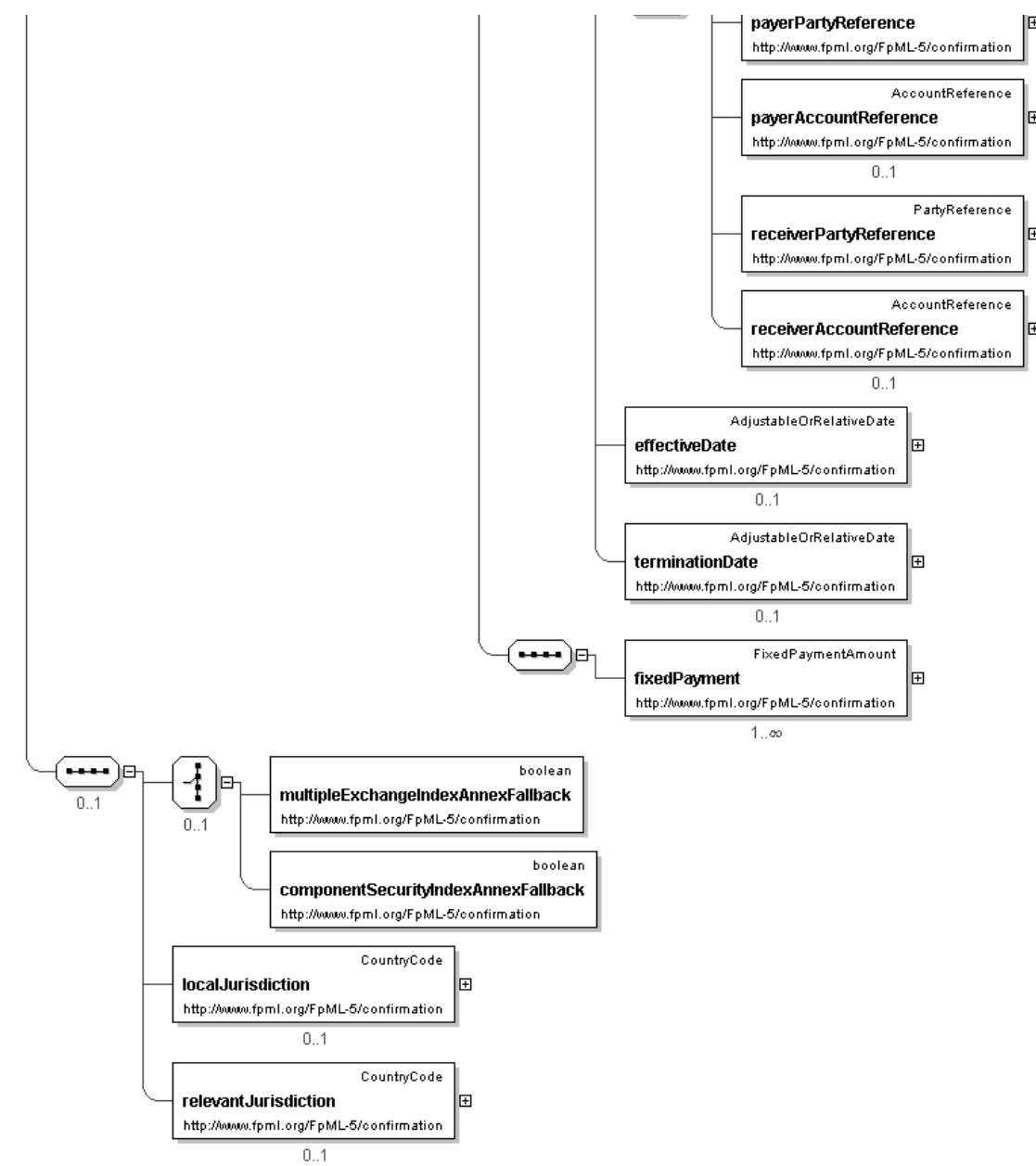
› [product](#)

<b>Name</b>	dividendSwapTransactionSupplement
<b>Type</b>	<a href="#">DividendSwapTransactionSupplement</a>
<b>Nillable</b>	no
<b>Abstract</b>	no
<b>Documentation</b>	Specifies the structure of the dividend swap transaction supplement.

#### Logical Diagram





**XML Instance Representation**

```

<dividendSwapTransactionSupplement
  id=" xsd:ID [0..1]">
  <productType> ProductType </productType> [0..*]
  'A classification of the type of product. FpML defines a simple product categorization using
  a coding scheme.'
  <productId> ProductId </productId> [0..*]
  'A product 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.'



<dividendLeg> DividendLeg </dividendLeg> [1]
'Dividend leg.'



<fixedLeg> FixedPaymentLeg </fixedLeg> [1]
'Fixed payment leg.'



Start Group: EquityUnderlyerProvisions.model [0..1]
Start Group: IndexAnnexFallback.model [0..1]
Start Choice [1]
    <multipleExchangeIndexAnnexFallback> xsd:boolean </multipleExchangeIndexAnnexFallback> [1]
    'For an index option transaction, a flag to indicate whether a relevant Multiple Exchange Index Annex is applicable to the transaction. This annex defines additional provisions which are applicable where an index is comprised of component securities that are traded on multiple exchanges.'



    <componentSecurityIndexAnnexFallback> xsd:boolean </componentSecurityIndexAnnexFallback> [1]
    'For an index option transaction, a flag to indicate whether a relevant Component Security Index Annex is applicable to the transaction.'



End Choice
End Group: IndexAnnexFallback.model
<localJurisdiction> CountryCode </localJurisdiction> [0..1]
'Local Jurisdiction is a term used in the AEJ Master Confirmation, which is used to determine local taxes, which shall mean taxes, duties, and similar charges imposed by the taxing authority of the Local Jurisdiction If this element is not present Local Jurisdiction is Not Applicable.'



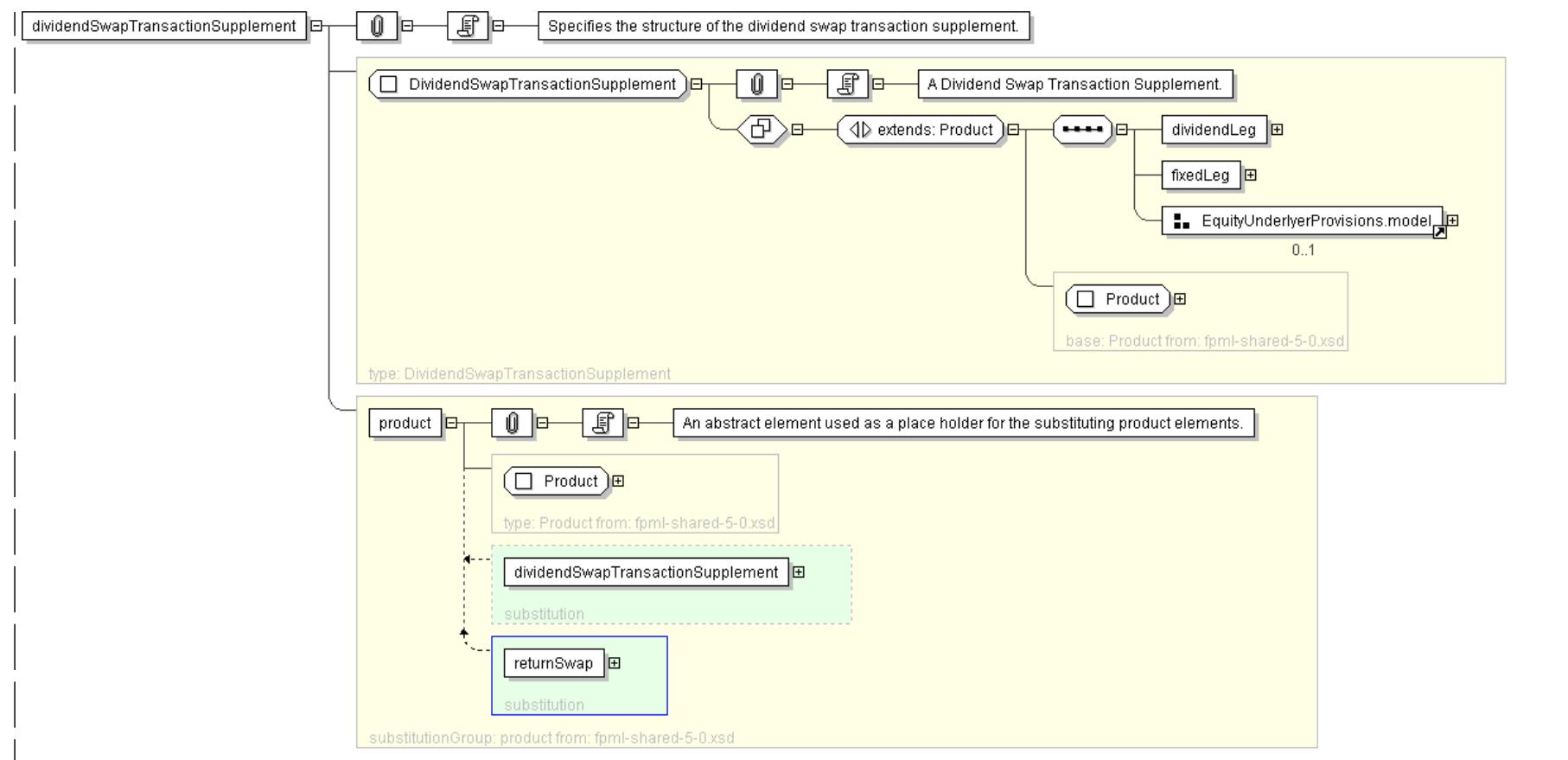
<relevantJurisdiction> CountryCode </relevantJurisdiction> [0..1]
'Relevant Jurisdiction is a term used in the AEJ Master Confirmation, which is used to determine local taxes, which shall mean taxes, duties and similar charges that would be imposed by the taxing authority of the Country of Underlyer on a Hypothetical Broker Dealer assuming the Applicable Hedge Positions are held by its office in the Relevant Jurisdiction. If this element is not present Relevant Jurisdiction is Not Applicable.'



End Group: EquityUnderlyerProvisions.model
</dividendSwapTransactionSupplement>

```

**Diagram**

**Schema Component Representation**

```
<xsd:element name="dividendSwapTransactionSupplement" type="DividendSwapTransactionSupplement" substitutionGroup="product"/>
```

top

**Global Definitions****Complex Type: DividendLeg**

Super-types:	<a href="#">DirectionalLegUnderlier</a> < <b>DividendLeg</b> (by extension)
Sub-types:	None
<b>Name</b>	DividendLeg
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">DividendSwapTransactionSupplement</a>
<b>Abstract</b>	no
<b>Documentation</b>	Floating Payment Leg of a Dividend Swap.

**XML Instance Representation**

```
<...>
  id="xsd:ID [0..1]">
    <legIdentifier> LegIdentifier </legIdentifier> [0..*]
```

'Version aware identification of this leg.'

<payerPartyReference> PartyReference </payerPartyReference> [1]  
'A reference to the party responsible for making the payments defined by this structure.'

<payerAccountReference> AccountReference </payerAccountReference> [0..1]  
'A reference to the account responsible for making the payments defined by this structure.'

<receiverPartyReference> PartyReference </receiverPartyReference> [1]  
'A reference to the party that receives the payments corresponding to this structure.'

<receiverAccountReference> AccountReference </receiverAccountReference> [0..1]  
'A reference to the account that receives the payments corresponding to this structure.'

<effectiveDate> AdjustableOrRelativeDate </effectiveDate> [0..1]  
'Specifies the effective date of this leg of the swap. When defined in relation to a date specified somewhere else in the document (through the relativeDate component), this element will typically point to the effective date of the other leg of the swap.'

<terminationDate> AdjustableOrRelativeDate </terminationDate> [0..1]  
'Specifies the termination date of this leg of the swap. When defined in relation to a date specified somewhere else in the document (through the relativeDate component), this element will typically point to the termination date of the other leg of the swap.'

<underlyer> Underlyer </underlyer> [1]  
'Specifies the underlyer of the leg.'

<settlementType> SettlementTypeEnum </settlementType> [0..1]  
<settlementDate> AdjustableOrRelativeDate </settlementDate> [0..1]

Start Group: SettlementAmountOrCurrency.model [0..1]

Start Choice [1]

- <settlementAmount> Money </settlementAmount> [1]  
'Settlement Amount'
- <settlementCurrency> Currency </settlementCurrency> [1]  
'Settlement Currency for use where the Settlement Amount cannot be known in advance'

End Choice

End Group: SettlementAmountOrCurrency.model

<fxFeature> FxFeature </fxFeature> [0..1]  
'Quanto, Composite, or Cross Currency FX features.'

<declaredCashDividendPercentage> NonNegativeDecimal </declaredCashDividendPercentage> [0..1]  
'Declared Cash Dividend Percentage.'

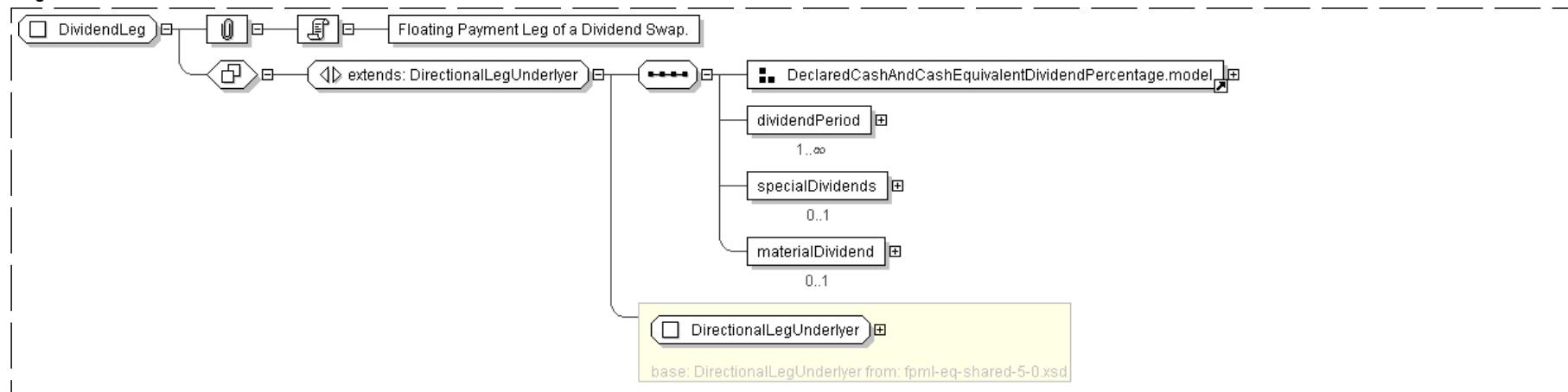
<declaredCashEquivalentDividendPercentage> NonNegativeDecimal  
</declaredCashEquivalentDividendPercentage> [0..1]  
'Declared Cash Equivalent Dividend Percentage.'

<dividendPeriod> DividendPeriodPayment </dividendPeriod> [1..\*]  
'One to many time bounded dividend payment periods, each with a fixed strike and dividend payment date per period.'

<specialDividends> xsd:boolean </specialDividends> [0..1]  
'If present and true, then special dividends and memorial dividends are applicable.'

<materialDividend> xsd:boolean </materialDividend> [0..1]  
'If present and true, then material non cash dividends are applicable.'

&lt; / ... &gt;

**Diagram****Schema Component Representation**

```

<xsd:complexType name="DividendLeg">
  <xsd:complexContent>
    <xsd:extension base="DirectionalLegUnderlyer">
      <xsd:sequence>
        <xsd:group ref="DeclaredCashAndCashEquivalentDividendPercentage.model"/>
        <xsd:element name="dividendPeriod" type="DividendPeriodPayment" maxOccurs="unbounded"/>
        <xsd:element name="specialDividends" type="xsd:boolean" minOccurs="0"/>
        <xsd:element name="materialDividend" type="xsd:boolean" minOccurs="0"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
  
```

top

**Complex Type: DividendPeriodPayment**

<b>Super-types:</b>	<a href="#">DividendPeriod</a> < <b>DividendPeriodPayment</b> (by extension)
<b>Sub-types:</b>	None
<b>Name</b>	DividendPeriodPayment
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">DividendLeg</a>
<b>Abstract</b>	no
<b>Documentation</b>	A time bounded dividend period, with fixed strike and a dividend payment date per period.

**XML Instance Representation**

```

<...
  id="xsd:ID [0..1]">
  <unadjustedStartDate> IdentifiedDate </unadjustedStartDate> [1]
  'Unadjusted inclusive dividend period start date.'

  <unadjustedEndDate> IdentifiedDate </unadjustedEndDate> [1]
  'Unadjusted inclusive dividend period end date.'

  <dateAdjustments> BusinessDayAdjustments </dateAdjustments> [1]
  
```

'Date adjustments for all unadjusted dates in this dividend period.'

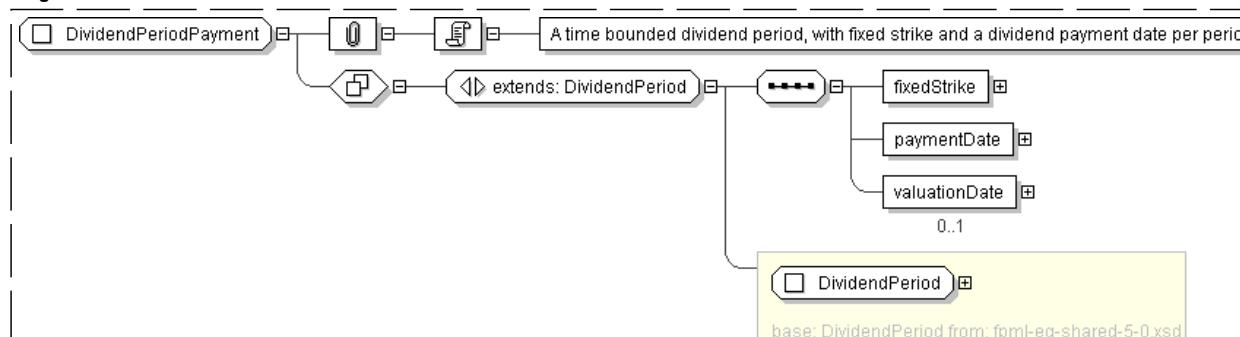
<underlyerReference> AssetReference </underlyerReference> [0..1]  
 Reference to the underlyer which is paying dividends. This should be used in all cases, and must be used where there are multiple underlying assets, to avoid any ambiguity about which asset the dividend period relates to.'

<fixedStrike> PositiveDecimal </fixedStrike> [1]  
 'Fixed strike.'

<paymentDate> AdjustableOrRelativeDate </paymentDate> [1]  
 'Dividend period amount payment date.'

<valuationDate> AdjustableOrRelativeDate </valuationDate> [0..1]  
 'Dividend period amount valuation date.'

<...>

**Diagram****Schema Component Representation**

```

<xsd:complexType name="DividendPeriodPayment">
  <xsd:complexContent>
    <xsd:extension base=" DividendPeriod ">
      <xsd:sequence>
        <xsd:element name="fixedStrike" type=" PositiveDecimal "/>
        <xsd:element name="paymentDate" type=" AdjustableOrRelativeDate "/>
        <xsd:element name="valuationDate" type=" AdjustableOrRelativeDate " minOccurs="0"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
  
```

[top](#)**Complex Type: DividendSwapTransactionSupplement**

**Super-types:** [Product](#) < **DividendSwapTransactionSupplement** (by extension)

**Sub-types:** None

<b>Name</b>	DividendSwap TransactionSupplement
<b>Used by (from the same schema document)</b>	Element <a href="#">dividendSwapTransactionSupplement</a>
<b>Abstract</b>	no
<b>Documentation</b>	A Dividend Swap Transaction Supplement.

**XML Instance Representation**

```
<...>
<id= "xsd:ID [0..1]">
<productType> ProductType </productType> [0..*]
'A classification of the type of product. FpML defines a simple product categorization using
a coding scheme.'

<productId> ProductId </productId> [0..*]
'A product 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.'

<dividendLeg> DividendLeg </dividendLeg> [1]
'Dividend leg.'

<fixedLeg> FixedPaymentLeg </fixedLeg> [1]
'Fixed payment leg.'
```

Start Group: EquityUnderlyerProvisions.model [0..1]

Start Group: IndexAnnexFallback.model [0..1]

Start Choice [1]

- <multipleExchangeIndexAnnexFallback> xsd:boolean </multipleExchangeIndexAnnexFallback> [1]
 'For an index option transaction, a flag to indicate whether a relevant Multiple Exchange
 Index Annex is applicable to the transaction. This annex defines additional provisions
 which are applicable where an index is comprised of component securities that are traded
 on multiple exchanges.'
- <componentSecurityIndexAnnexFallback> xsd:boolean </componentSecurityIndexAnnexFallback> [1]
 'For an index option transaction, a flag to indicate whether a relevant Component
 Security Index Annex is applicable to the transaction.'

End Choice

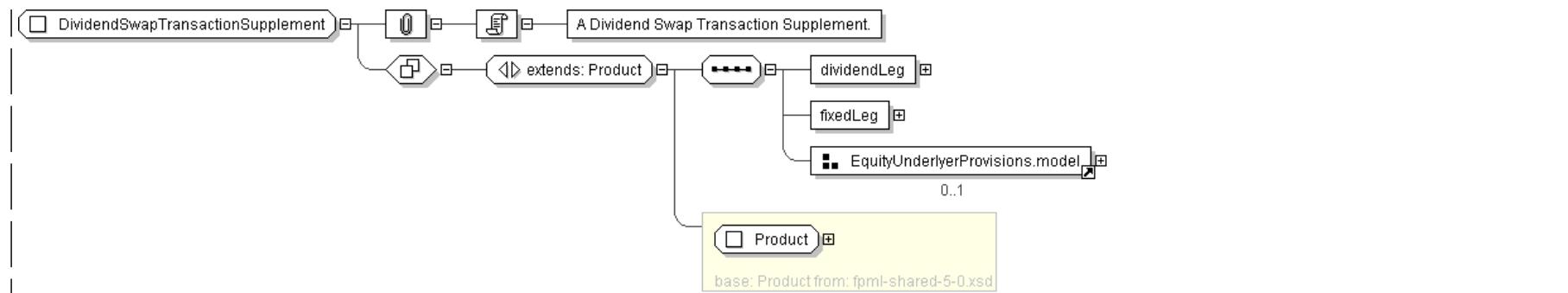
End Group: IndexAnnexFallback.model

<localJurisdiction> CountryCode </localJurisdiction> [0..1]
 'Local Jurisdiction is a term used in the AEJ Master Confirmation, which is used to
 determine local taxes, which shall mean taxes, duties, and similar charges imposed by
 the taxing authority of the Local Jurisdiction If this element is not present
 Local Jurisdiction is Not Applicable.'

<relevantJurisdiction> CountryCode </relevantJurisdiction> [0..1]
 'Relevant Jurisdiction is a term used in the AEJ Master Confirmation, which is used
 to determine local taxes, which shall mean taxes, duties and similar charges that would
 be imposed by the taxing authority of the Country of Underlyer on a Hypothetical Broker
 Dealer assuming the Applicable Hedge Positions are held by its office in the
 Relevant Jurisdiction. If this element is not present Relevant Jurisdiction is Not Applicable.'

End Group: EquityUnderlyerProvisions.model

**Diagram**



### Schema Component Representation

```

<xsd:complexType name="DividendSwapTransactionSupplement">
  <xsd:complexContent>
    <xsd:extension base=" Product ">
      <xsd:sequence>
        <xsd:element name="dividendLeg" type=" DividendLeg " />
        <xsd:element name="fixedLeg" type=" FixedPaymentLeg " />
        <xsd:group ref=" EquityUnderlyerProvisions.model " minOccurs="0" />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
  
```

[top](#)

### Complex Type: [FixedPaymentAmount](#)

<b>Super-types:</b>	<a href="#">PaymentBase</a> < <b>FixedPaymentAmount</b> (by extension)
<b>Sub-types:</b>	None

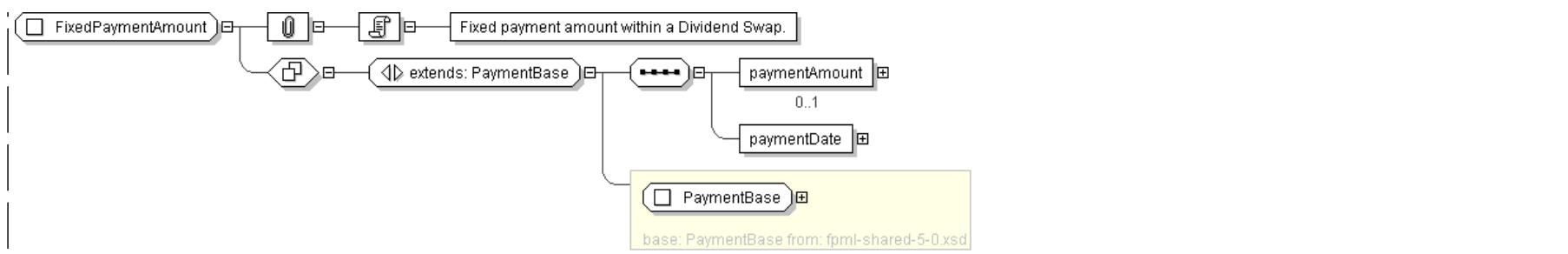
<b>Name</b>	FixedPaymentAmount
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">FixedPaymentLeg</a>
<b>Abstract</b>	no
<b>Documentation</b>	Fixed payment amount within a Dividend Swap.

### XML Instance Representation

```

<...
  id=" xsd:ID [0..1]">
  <paymentAmount> NonNegativeMoney </paymentAmount> [0..1]
  'Payment amount, which is optional since the payment amount may be calculated using
  fixed strike and number of open units.'
  <paymentDate> RelativeDateOffset </paymentDate> [1]
  'Payment date relative to another date.'
</...>
  
```

### Diagram



### Schema Component Representation

```

<xsd:complexType name="FixedPaymentAmount">
  <xsd:complexContent>
    <xsd:extension base=" PaymentBase ">
      <xsd:sequence>
        <xsd:element name="paymentAmount" type=" NonNegativeMoney " minOccurs="0" />
        <xsd:element name="paymentDate" type=" RelativeDateOffset " />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
  
```

top

### Complex Type: FixedPaymentLeg

Super-types:	<a href="#">DirectionalLeg</a> < <b>FixedPaymentLeg</b> (by extension)
Sub-types:	None

Name	FixedPaymentLeg
Used by (from the same schema document)	Complex Type <a href="#">DividendSwapTransactionSupplement</a>
Abstract	no
Documentation	Fixed Payment Leg of a Dividend Swap.

### XML Instance Representation

```

<...
id=" xsd:ID [0..1]">
<legIdentifier> LegIdentifier </legIdentifier> [0...*]
'Version aware identification of this leg.'

<payerPartyReference> PartyReference </payerPartyReference> [1]
'A reference to the party responsible for making the payments defined by this structure.'

<payerAccountReference> AccountReference </payerAccountReference> [0..1]
'A reference to the account responsible for making the payments defined by this structure.'

<receiverPartyReference> PartyReference </receiverPartyReference> [1]
'A reference to the party that receives the payments corresponding to this structure.'

<receiverAccountReference> AccountReference </receiverAccountReference> [0..1]
'A reference to the account that receives the payments corresponding to this structure.'

<effectiveDate> AdjustableOrRelativeDate </effectiveDate> [0..1]
'Specifies the effective date of this leg of the swap. When defined in relation to a
date specified somewhere else in the document (through the relativeDate component),
  
```

this element will typically point to the effective date of the other leg of the swap.'

<terminationDate> AdjustableOrRelativeDate </terminationDate> [0..1]

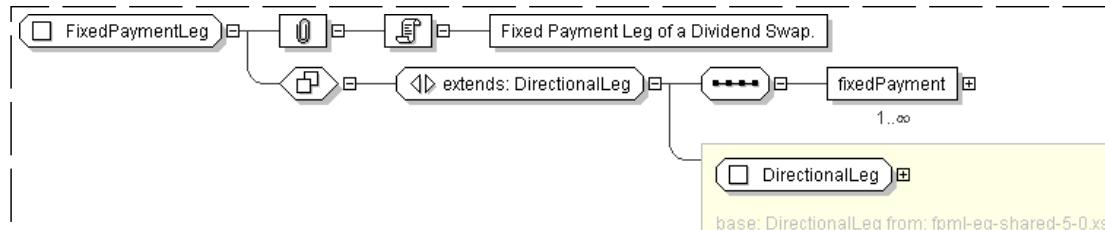
'Specifies the termination date of this leg of the swap. When defined in relation to a date specified somewhere else in the document (through the relativeDate component), this element will typically point to the termination date of the other leg of the swap.'

<fixedPayment> FixedPaymentAmount </fixedPayment> [1..\*]

'Fixed payment of a dividend swap, payment date is relative to a dividend period payment date. Commonly the dividend leg and the fixed payment leg will pay out on the same date, and the payments will be netted.'

</...>

#### Diagram



#### Schema Component Representation

```

<xsd:complexType name="FixedPaymentLeg">
  <xsd:complexContent>
    <xsd:extension base=" DirectionalLeg ">
      <xsd:sequence>
        <xsd:element name="fixedPayment" type=" FixedPaymentAmount " maxOccurs="unbounded" />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
  
```

[top](#)

## Legend

### Complex Type:

Schema Component Type

### AusAddress

Schema Component Name

Super-types:

[Address](#) < AusAddress (by extension)

• [OLDAddress](#) (by restriction)

If this schema component is a type definition, its type hierarchy is shown in a gray-bordered box.

Name

AusAddress

Abstract

no

The table above displays the properties of this schema component.

#### XML Instance Representation

```

<... country="Australia" >
<unitNo> string </unitNo> [0..1]
<houseNo> string </houseNo> [1]
  
```

```

<street> string </street> [1]
Start Choice [1]
<city> string </city> [1]
<town> string </town> [1]
End Choice
<state> AusStates </state> [1]
<postcode> string <><pattern = [1-9][0-9]{3}>> </postcode> [1]
</...>

```

The XML Instance Representation table above shows the schema component's content as an XML instance.

- The minimum and maximum occurrence of elements and attributes are provided in square brackets, e.g. [0..1].
- Model group information are shown in gray, e.g. Start Choice ... End Choice.
- For type derivations, the elements and attributes that have been added to or changed from the base type's content are shown in **bold**.
- If an element/attribute has a fixed value, the fixed value is shown in green, e.g. country="Australia".
- Otherwise, the type of the element/attribute is displayed.
  - If the element/attribute's type is in the schema, a link is provided to it.
  - For local simple type definitions, the constraints are displayed in angle brackets, e.g. <><pattern = [1-9][0-9]{3}>>.

### Schema Component Representation

```

<complexType name="AusAddress">
<complexContent>
<extension base=" Address ">
<sequence>
<element name="state" type=" AusStates " />
<element name="postcode">
<simpleType>
<restriction base=" string ">
<pattern value="[1-9][0-9]{3}" />
</restriction>
</simpleType>
</element>
</sequence>
<attribute name="country" type=" string " fixed="Australia" />
</extension>
</complexContent>
</complexType>

```

The Schema Component Representation table above displays the underlying XML representation of the schema component. (Annotations are not shown.)

[top](#)

## Glossary

**Abstract** (Applies to complex type definitions and element declarations). An abstract element or complex type cannot be used to validate an element instance. If there is a reference to an abstract element, only element declarations that can substitute the abstract element can be used to validate the instance. For references to abstract type definitions, only derived types can be used.

**All Model Group** Child elements can be provided *in any order* in instances. See: <http://www.w3.org/TR/xmlschema-1/#element-all>.

**Choice Model Group** Only one from the list of child elements and model groups can be provided in instances. See: <http://www.w3.org/TR/xmlschema-1/#element-choice>.

**Collapse Whitespace Policy** Replace tab, line feed, and carriage return characters with space character (Unicode character 32). Then, collapse contiguous sequences of space characters into single space character, and remove leading and trailing space characters.

**Disallowed Substitutions** (Applies to element declarations). If *substitution* is specified, then *substitution group* members cannot be used in place of the given element declaration to validate element instances. If *derivation methods*, e.g. extension, restriction, are specified, then the given element declaration will not validate element instances that have types derived from the element declaration's type using the specified derivation methods. Normally, element instances can override their declaration's type by specifying an *xsi:type* attribute.

**Key Constraint** Like [Uniqueness Constraint](#), but additionally requires that the specified value(s) must be provided. See: [http://www.w3.org/TR/xmlschema-1/#cIdentity-constraint\\_Definitions](http://www.w3.org/TR/xmlschema-1/#cIdentity-constraint_Definitions).

**Key Reference Constraint** Ensures that the specified value(s) must match value(s) from a [Key Constraint](#) or [Uniqueness Constraint](#). See: [http://www.w3.org/TR/xmlschema-1/#cIdentity-constraint\\_Definitions](http://www.w3.org/TR/xmlschema-1/#cIdentity-constraint_Definitions).

**Model Group** Groups together element content, specifying the order in which the element content can occur and the number of times the group of element content may be repeated. See: [http://www.w3.org/TR/xmlschema-1/#Model\\_Groups](http://www.w3.org/TR/xmlschema-1/#Model_Groups).

**Nillable** (Applies to element declarations). If an element declaration is nillable, instances can use the `xsi:nil` attribute. The `xsi:nil` attribute is the boolean attribute, `nil`, from the <http://www.w3.org/2001/XMLSchema-instance> namespace. If an element instance has an `xsi:nil` attribute set to true, it can be left empty, even though its element declaration may have required content.

**Notation** A notation is used to identify the format of a piece of data. Values of elements and attributes that are of type, NOTATION, must come from the names of declared notations. See: [http://www.w3.org/TR/xmlschema-1/#cNotation\\_Declarations](http://www.w3.org/TR/xmlschema-1/#cNotation_Declarations).

**Preserve Whitespace Policy** Preserve whitespaces exactly as they appear in instances.

**Prohibited Derivations** (Applies to type definitions). Derivation methods that cannot be used to create sub-types from a given type definition.

**Prohibited Substitutions** (Applies to complex type definitions). Prevents sub-types that have been derived using the specified derivation methods from validating element instances in place of the given type definition.

**Replace Whitespace Policy** Replace tab, line feed, and carriage return characters with space character (Unicode character 32).

**Sequence Model Group** Child elements and model groups must be provided *in the specified order* in instances. See: <http://www.w3.org/TR/xmlschema-1/#element-sequence>.

**Substitution Group** Elements that are *members* of a substitution group can be used wherever the `head` element of the substitution group is referenced.

**Substitution Group Exclusions** (Applies to element declarations). Prohibits element declarations from nominating themselves as being able to substitute a given element declaration, if they have types that are derived from the original element's type using the specified derivation methods.

**Target Namespace** The target namespace identifies the namespace that components in this schema belongs to. If no target namespace is provided, then the schema components do not belong to any namespace.

**Uniqueness Constraint** Ensures uniqueness of an element/attribute value, or a combination of values, within a specified scope. See: [http://www.w3.org/TR/xmlschema-1/#cIdentity-constraint\\_Definitions](http://www.w3.org/TR/xmlschema-1/#cIdentity-constraint_Definitions).

[top](#)

---

Generated by [oXygen XML Editor](#) using a modified version of `xs3p` that adds schema diagrams and chunking support.

# XML Schema Documentation

## Table of Contents

- [Schema Document Properties](#)
- [Global Declarations](#)
  - [Element: strategy](#)
- [Global Definitions](#)
  - [Attribute Group: VersionAttributes.atts](#)
  - [Complex Type: Allocation](#)
  - [Complex Type: Allocations](#)
  - [Complex Type: Approval](#)
  - [Complex Type: Approvals](#)
  - [Complex Type: BestFitTrade](#)
  - [Complex Type: Collateral](#)
  - [Complex Type: ContractId](#)
  - [Complex Type: ContractIdentifier](#)
  - [Complex Type: CreditDerivativesNotices](#)
  - [Complex Type: DataDocument](#)
  - [Complex Type: Difference](#)
  - [Complex Type: Document](#)
  - [Complex Type: ExecutionDateTime](#)
  - [Complex Type: FirstPeriodStartDate](#)
  - [Complex Type: IndependentAmount](#)
  - [Complex Type: LinkId](#)
  - [Complex Type: PartyPortfolioName](#)
  - [Complex Type: PartyTradeIdentifier](#)
  - [Complex Type: PartyTradeIdentifiers](#)
  - [Complex Type: PartyTradeInformation](#)
  - [Complex Type: PaymentDetail](#)
  - [Complex Type: PaymentRule](#)
  - [Complex Type: PercentageRule](#)
  - [Complex Type: Portfolio](#)
  - [Complex Type: PortfolioName](#)
  - [Complex Type: QueryParameter](#)
  - [Complex Type: QueryParameterId](#)
  - [Complex Type: QueryParameterOperator](#)
  - [Complex Type: QueryPortfolio](#)
  - [Complex Type: ReportingRole](#)
  - [Complex Type: Strategy](#)
  - [Complex Type: Trade](#)
  - [Complex Type: TradeCategory](#)
  - [Complex Type: TradeHeader](#)
  - [Complex Type: Tradeld](#)
  - [Complex Type: TradelIdentifier](#)
  - [Complex Type: Trader](#)
  - [Complex Type: Unit](#)
  - [Complex Type: Validation](#)
  - [Complex Type: VersionedContractId](#)
  - [Complex Type: VersionedTradeld](#)
  - [Model Group: AccountReferenceOrPartyReference.model](#)
  - [Model Group: AllocationContent.model](#)
  - [Model Group: AmendmentDetails.model](#)
  - [Model Group: CalculationAgent.model](#)
  - [Model Group: TradeOrTradeReference.model](#)
  - [Model Group: Validation.model](#)
  - [Simple Type: QueryParameterValue](#)
- [Legend](#)

## Schema Document Properties

<b>Target Namespace</b>	<a href="http://www.fpml.org/FpML-5/confirmation">http://www.fpml.org/FpML-5/confirmation</a>
<b>Version</b>	\$Revision: 2674 \$
<b>Element and Attribute Namespaces</b>	<ul style="list-style-type: none"> <li>Global element and attribute declarations belong to this schema's target namespace.</li> <li>By default, local element declarations belong to this schema's target namespace.</li> <li>By default, local attribute declarations have no namespace.</li> </ul>
<b>Schema Composition</b>	<ul style="list-style-type: none"> <li>This schema includes components from the following schema document(s):           <ul style="list-style-type: none"> <li><a href="#">fpml-shared-5-0.xsd</a></li> </ul> </li> </ul>

## Declared Namespaces

Prefix	Namespace
Default namespace	<a href="http://www.fpml.org/FpML-5/confirmation">http://www.fpml.org/FpML-5/confirmation</a>
xml	http://www.w3.org/XML/1998/namespace
fpml-annotation	http://www.fpml.org/annotation
xsd	http://www.w3.org/2001/XMLSchema
ecore	http://www.eclipse.org/emf/2002/Ecore

## Schema Component Representation

```
<xsd:schema targetNamespace="http://www.fpml.org/FpML-5/confirmation"
documentRoot="FpML" nsPrefix="conf" package="org.fpml.confirmation" version="$Revision: 2674 "
$ elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xsd:include schemaLocation="fpml-shared-5-0.xsd"/>
  ...
</xsd:schema>
```

## Global Declarations

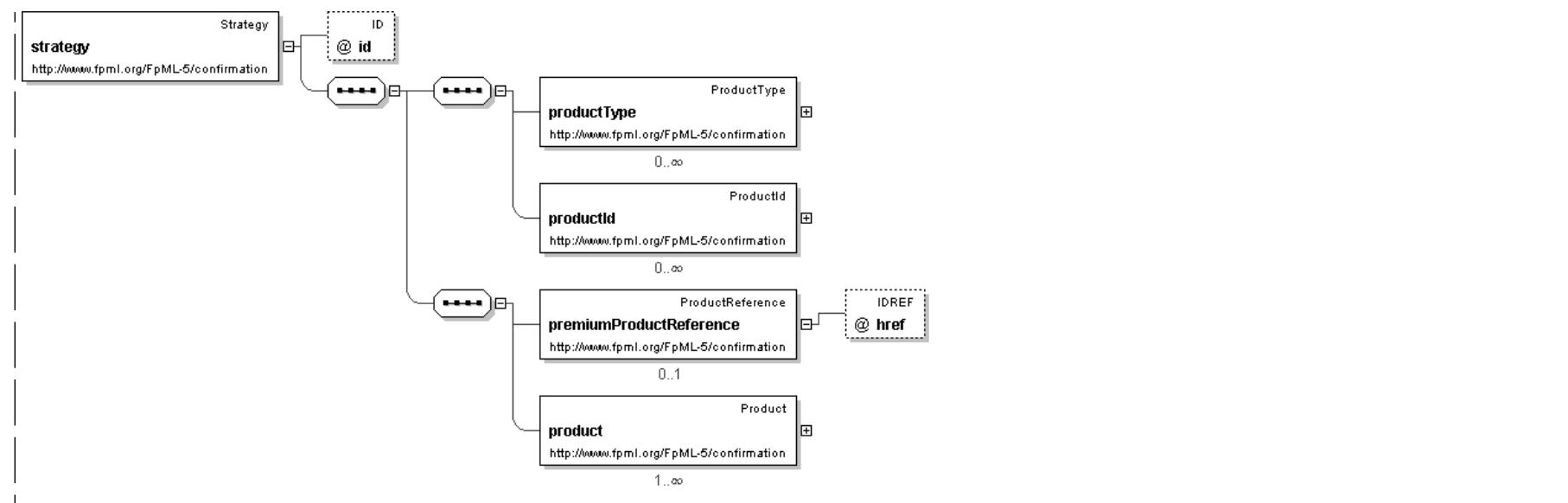
### Element: **strategy**

- This element can be used wherever the following element is referenced:
  - [product](#)

<b>Name</b>	strategy
<b>Type</b>	<a href="#">Strategy</a>
<b>Nillable</b>	no
<b>Abstract</b>	no
<b>Documentation</b>	A strategy product.

## Logical Diagram

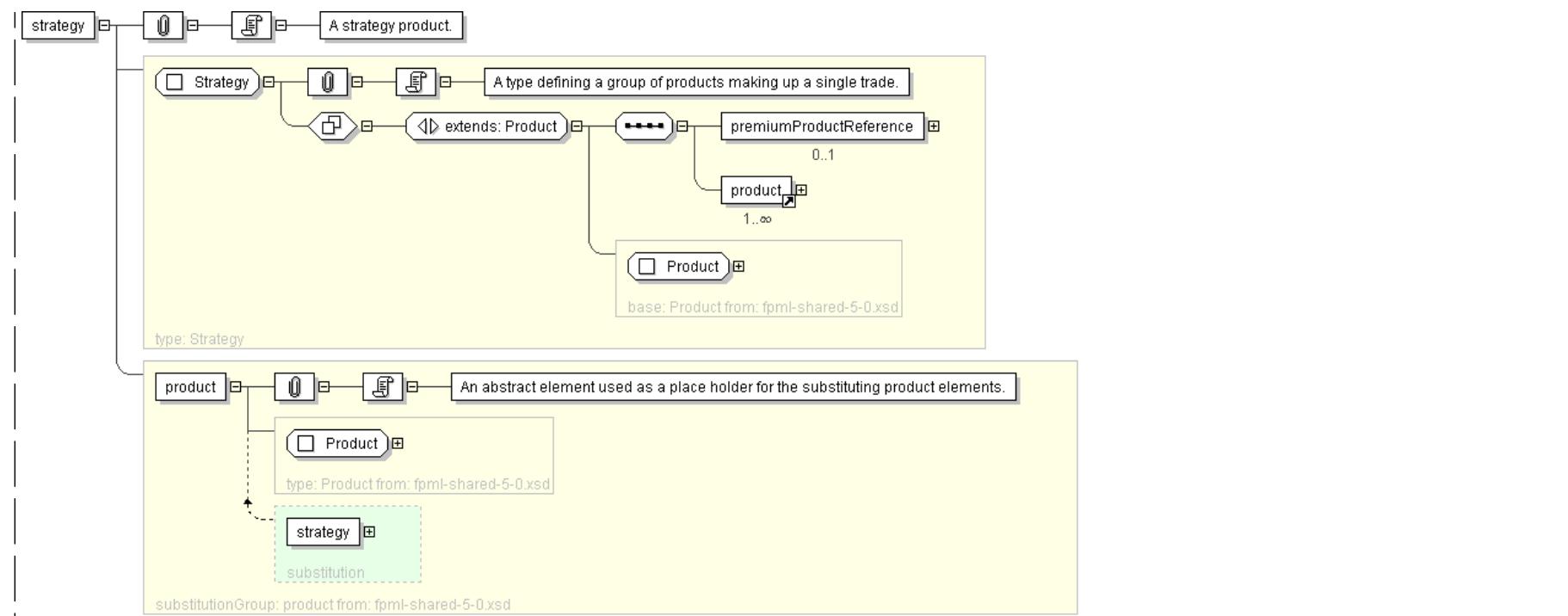


**XML Instance Representation**

```

<strategy
  id=" xsd:ID [0..1]">
  <productType> ProductType </productType> [0..*]
  'A classification of the type of product. FpML defines a simple product categorization using
  a coding scheme.'
  <productId> ProductId </productId> [0..*]
  'A product 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.'
  <premiumProductReference> ProductReference </premiumProductReference> [0..1]
  'Indicates which product within a strategy represents the premium payment.'
  <product> ... </product> [1..*]
</strategy>
  
```

**Diagram**



#### Schema Component Representation

```
<xsd:element name="strategy" type="#Strategy" substitutionGroup="product"/>
```

top

## Global Definitions

### Attribute Group: VersionAttributes.atts

Name	VersionAttributes.atts
Used by (from the same schema document)	Complex Type <a href="#">Document</a>
Documentation	Set of attributes that define versioning information.

#### XML Instance Representation

```

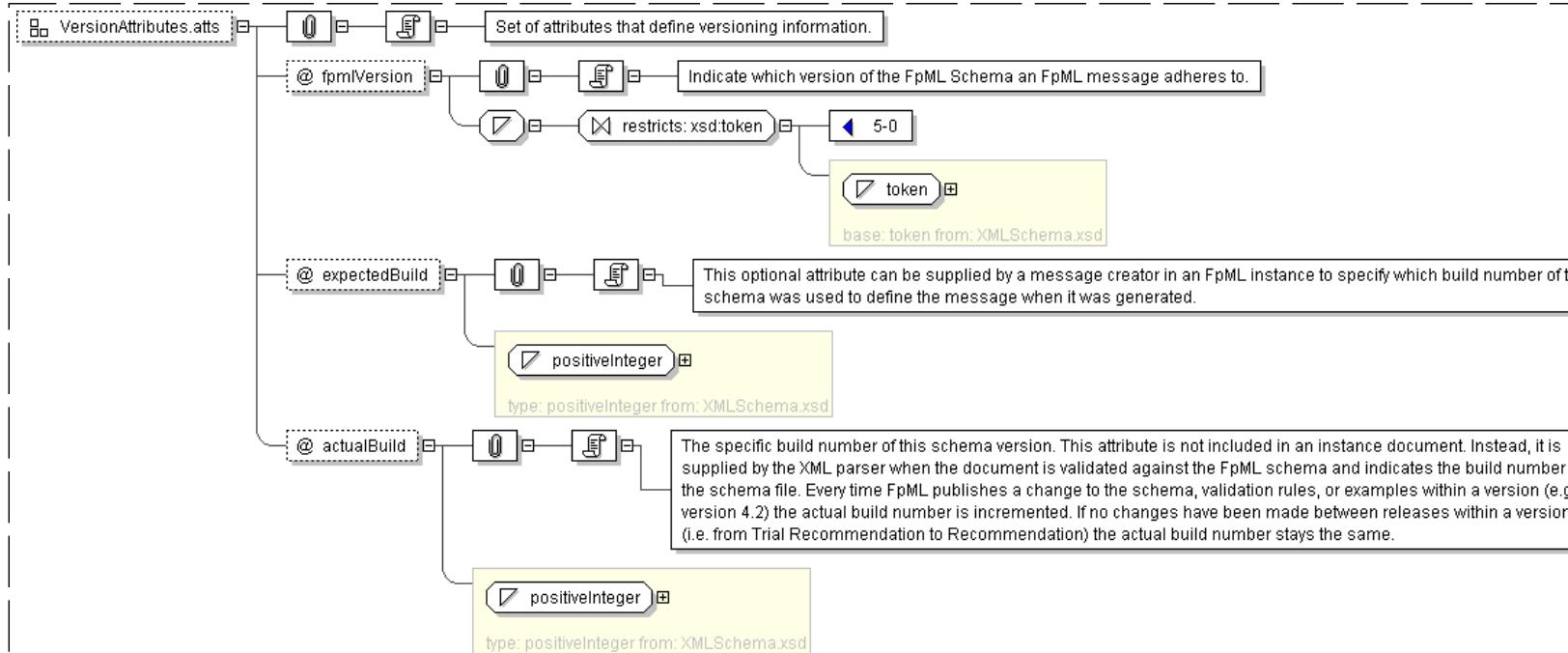
fpmlVersion=" xsd:token (value comes from list: {'5-0'}) [1]
'Indicate which version of the FpML Schema an FpML message adheres to.'

"
expectedBuild=" xsd:positiveInteger [0..1]
'This optional attribute can be supplied by a message creator in an FpML instance to
specify which build number of the schema was used to define the message when it was generated.'

"
actualBuild="8 [0..1]
'The specific build number of this schema version. This attribute is not included in
an instance document. Instead, it is supplied by the XML parser when the document is
validated against the FpML schema and indicates the build number of the schema file. Every
time FpML publishes a change to the schema, validation rules, or examples within a version
(e.g., version 4.2) the actual build number is incremented. If no changes have been
'

```

made between releases within a version (i.e. from Trial Recommendation to Recommendation)  
the actual build number stays the same.'

**Diagram****Schema Component Representation**

```

<xsd:attributeGroup name="VersionAttributes.atts">
  <xsd:attribute name="fpmVersion" use="required">
    <xsd:simpleType>
      <xsd:restriction base=" xsd:token ">
        <xsd:enumeration value="5-0"/>
      </xsd:restriction>
    </xsd:simpleType>
  </xsd:attribute>
  <xsd:attribute name="expectedBuild" type=" xsd:positiveInteger "/>
  <xsd:attribute name="actualBuild" type=" xsd:positiveInteger " fixed="8"/>
</xsd:attributeGroup>
  
```

top

**Complex Type: Allocation**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None

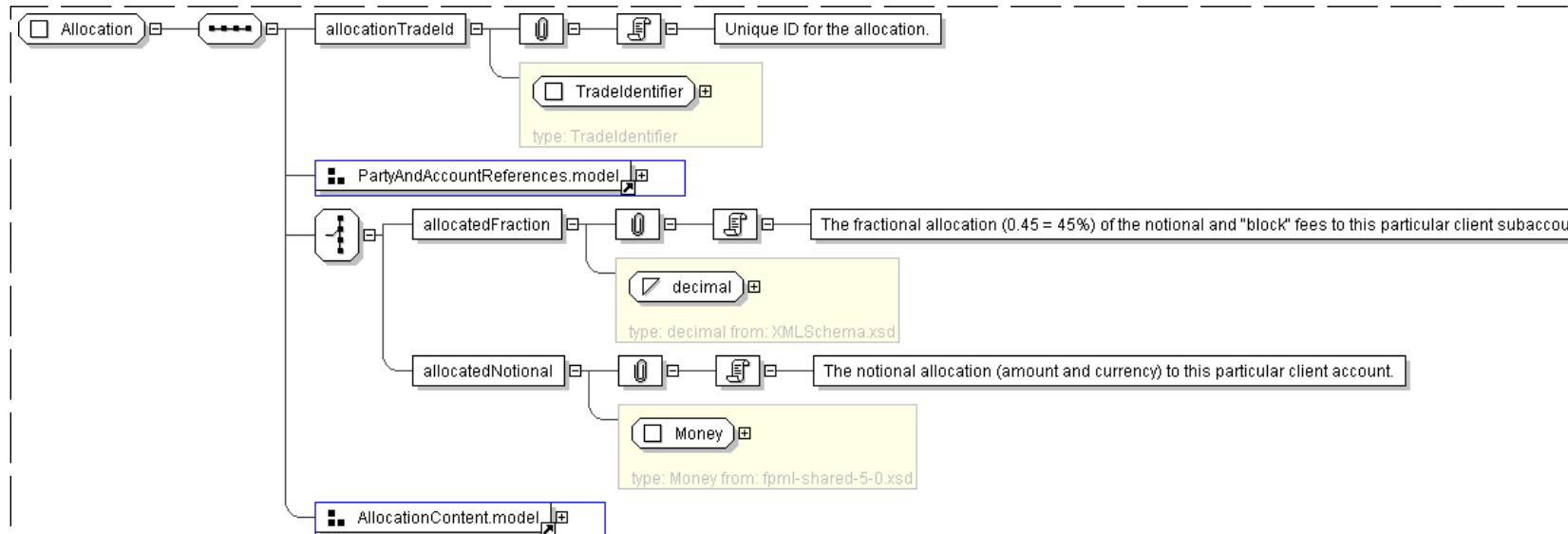
<b>Name</b>	Allocation
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">Allocations</a>
<b>Abstract</b>	no

**XML Instance Representation**

```

| <...>
|   <allocationTradeId> TradeIdentifier </allocationTradeId> [1]
|     'Unique ID for the allocation.'
|
|   <partyReference> PartyReference </partyReference> [1]
|     'Reference to a party.'
|
|   <accountReference> AccountReference </accountReference> [0..1]
|     'Reference to an account.'
|
Start Choice [1]
  <allocatedFraction> xsd:decimal </allocatedFraction> [1]
    'The fractional allocation (0.45 = 45%) of the notional and "block" fees to this
    particular client subaccount.'
  |
  <allocatedNotional> Money </allocatedNotional> [1]
    'The notional allocation (amount and currency) to this particular client account.'
|
End Choice
<collateral> Collateral </collateral> [0..1]
  'The sum that must be posted upfront to collateralize against counterparty credit risk.'
|
<creditChargeAmount> Money </creditChargeAmount> [0..1]
  'Special credit fee assessed to certain institutions.'
|
<approvals> Approvals </approvals> [0..1]
  'A container for approval states in the workflow.'
|
<masterConfirmationDate> xsd:date </masterConfirmationDate> [0..1]
  'The date of the confirmation executed between the parties and intended to govern the
  allocated trade between those parties.'
|
</...>

```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="Allocation">
  <xsd:sequence>
    <xsd:element name="allocationTradeId" type=" TradeIdentifier " />
    <xsd:group ref=" PartyAndAccountReferences.model " />
    <xsd:choice>
      <xsd:element name="allocatedFraction" type=" xsd:decimal " />
      <xsd:element name="allocatedNotional" type=" Money " />
    </xsd:choice>
    <xsd:group ref=" AllocationContent.model " />
  </xsd:sequence>
</xsd:complexType>

```

[top](#)

## Complex Type: Allocations

Super-types:	None
Sub-types:	None

Name	Allocations
Used by (from the same schema document)	Complex Type <a href="#">Trade</a>
Abstract	no

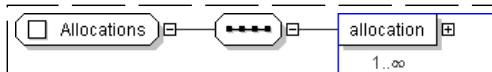
### XML Instance Representation

```

<...>
  <allocation> Allocation </allocation> [1...*]
</...>

```

### Diagram



### Schema Component Representation

```

<xsd:complexType name="Allocations">
  <xsd:sequence>
    <xsd:element name="allocation" type=" Allocation " maxOccurs="unbounded" />
  </xsd:sequence>
</xsd:complexType>

```

[top](#)

## Complex Type: Approval

Super-types:	None
Sub-types:	None

Name	Approval
Used by (from the same schema document)	Complex Type <a href="#">Approvals</a>
Abstract	no
Documentation	A specific approval state in the workflow.

### XML Instance Representation

```

<...>
  <type> xsd:normalizedString </type> [1]
  'The type of approval (e.g. \"Credit\").'

```

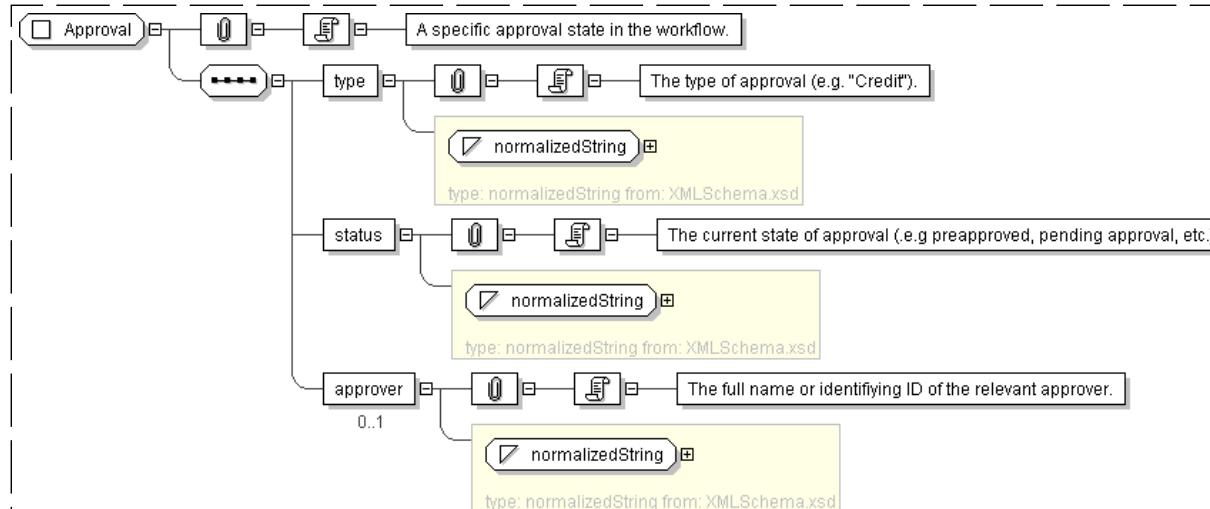
```

<status> xsd:normalizedString </status> [1]
'The current state of approval (.e.g preapproved, pending approval, etc.)'

<approver> xsd:normalizedString </approver> [0..1]
'The full name or identifying ID of the relevant approver.'

</...>

```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="Approval">
  <xsd:sequence>
    <xsd:element name="type" type="xsd:normalizedString" />
    <xsd:element name="status" type="xsd:normalizedString" />
    <xsd:element name="approver" type="xsd:normalizedString" minOccurs="0" />
  </xsd:sequence>
</xsd:complexType>

```

[top](#)**Complex Type: Approvals**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None

<b>Name</b>	Approvals
<b>Used by (from the same schema document)</b>	Model Group <a href="#">AllocationContent.model</a>
<b>Abstract</b>	no

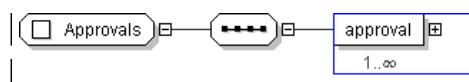
**XML Instance Representation**

```

<...>
  <approval> Approval </approval> [1..*]
</...>

```

**Diagram**

**Schema Component Representation**

```

<xsd:complexType name="Approvals">
  <xsd:sequence>
    <xsd:element name="approval" type="Approval" maxOccurs="unbounded" />
  </xsd:sequence>
</xsd:complexType>

```

[top](#)**Complex Type: BestFitTrade**

**Super-types:** None  
**Sub-types:** None

<b>Name</b>	BestFitTrade
<b>Abstract</b>	no
<b>Documentation</b>	A type used to record the differences between the current trade and another indicated trade.

**XML Instance Representation**

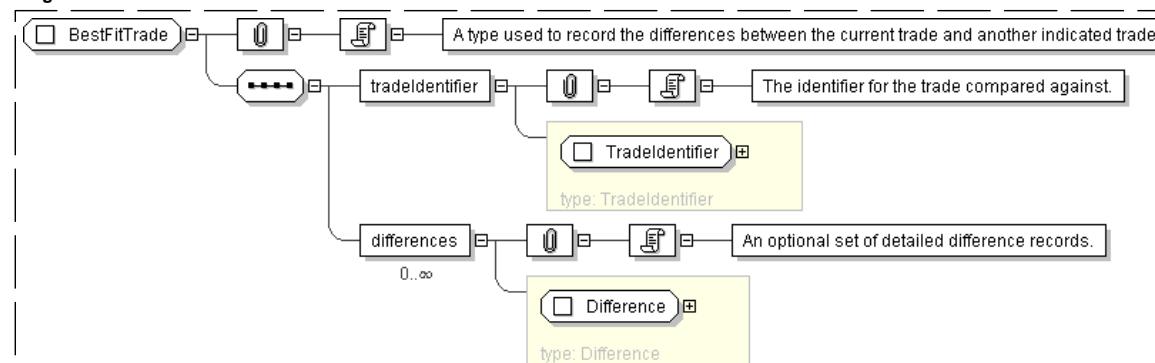
```

<...>
  <tradeIdentifier> TradeIdentifier </tradeIdentifier> [1]
  'The identifier for the trade compared against.'

  <differences> Difference </differences> [0..*]
  'An optional set of detailed difference records.'

</...>

```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="BestFitTrade">
  <xsd:sequence>
    <xsd:element name="tradeIdentifier" type="TradeIdentifier" />
    <xsd:element name="differences" type="Difference" minOccurs="0" maxOccurs="unbounded" />
  </xsd:sequence>
</xsd:complexType>

```

[top](#)

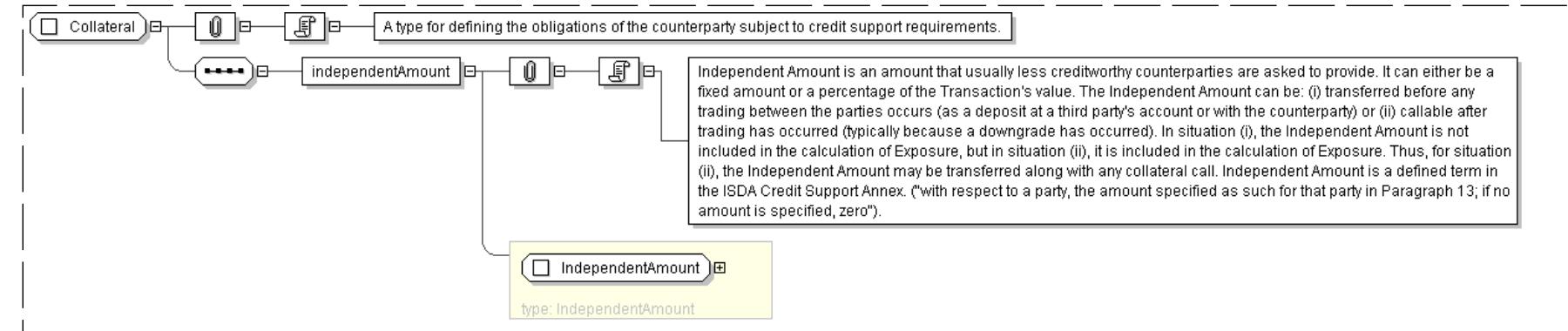
**Complex Type: Collateral**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None

<b>Name</b>	Collateral
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">Trade</a> , Model Group <a href="#">AllocationContent.model</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type for defining the obligations of the counterparty subject to credit support requirements.

**XML Instance Representation**

<...>	<pre>&lt;independentAmount&gt; <a href="#">IndependentAmount</a> &lt;/independentAmount&gt; [1]</pre> <p>'Independent Amount is an amount that usually less creditworthy counterparties are asked to provide. It can either be a fixed amount or a percentage of the Transaction's value. The Independent Amount can be: (i) transferred before any trading between the parties occurs (as a deposit at a third party's account or with the counterparty) or (ii) callable after trading has occurred (typically because a downgrade has occurred). In situation (i), the Independent Amount is not included in the calculation of Exposure, but in situation (ii), it is included in the calculation of Exposure. Thus, for situation (ii), the Independent Amount may be transferred along with any collateral call. Independent Amount is a defined term in the ISDA Credit Support Annex. ("with respect to a party, the amount specified as such for that party in Paragraph 13; if no amount is specified, zero").'</p>
</...>	

**Diagram****Schema Component Representation**

```
<xsd:complexType name="Collateral">
  <xsd:sequence>
    <xsd:element name="independentAmount" type=" IndependentAmount " />
  </xsd:sequence>
</xsd:complexType>
```

top

**Complex Type: ContractId**

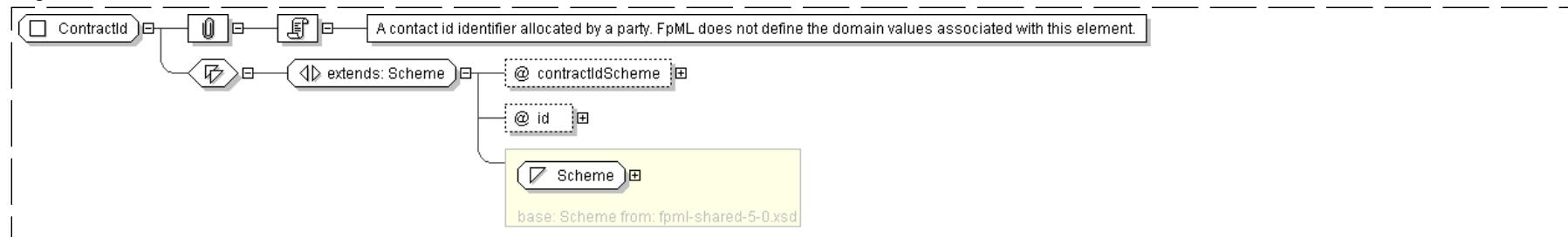
<b>Super-types:</b>	<a href="#">Scheme</a> < <a href="#">ContractId</a> (by extension)
<b>Sub-types:</b>	None

<b>Name</b>	ContractId
-------------	------------

<b>Used by (from the same schema document)</b>	Complex Type <a href="#">ContractIdentifier</a> , Complex Type <a href="#">VersionedContractId</a>
<b>Abstract</b>	no
<b>Documentation</b>	A contact id identifier allocated by a party. FpML does not define the domain values associated with this element.

**XML Instance Representation**

```
<...  
contractIdScheme=" xsd:anyURI [1]"  
id=" xsd:ID [0..1]">  
Scheme  
</...>
```

**Diagram****Schema Component Representation**

```
<xsd:complexType name="ContractId">  
  <xsd:simpleContent>  
    <xsd:extension base=" Scheme ">  
      <xsd:attribute name="contractIdScheme" type=" xsd:anyURI " use="required"/>  
      <xsd:attribute name="id" type=" xsd:ID "/>  
    </xsd:extension>  
  </xsd:simpleContent>  
</xsd:complexType>
```

[top](#)**Complex Type: ContractIdentifier**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None

<b>Name</b>	ContractIdentifier
<b>Abstract</b>	no
<b>Documentation</b>	A type defining a contract identifier issued by the indicated party.

**XML Instance Representation**

```
<...  
id=" xsd:ID [0..1]">  
<partyReference> PartyReference </partyReference> [1]  
'A pointer style reference to a party identifier defined elsewhere in the document. The  
party referenced has allocated the contract identifier.'
```

**Start Choice [1]**

'Where the legal activity is to agree a contract of variation then the business process  
should be to modify a contract. This is a contract in its own right and not a version of  
a previous contract. Where the business process is to replace and supersede a contract then  
you have a new contract and a contract version should not be used.'

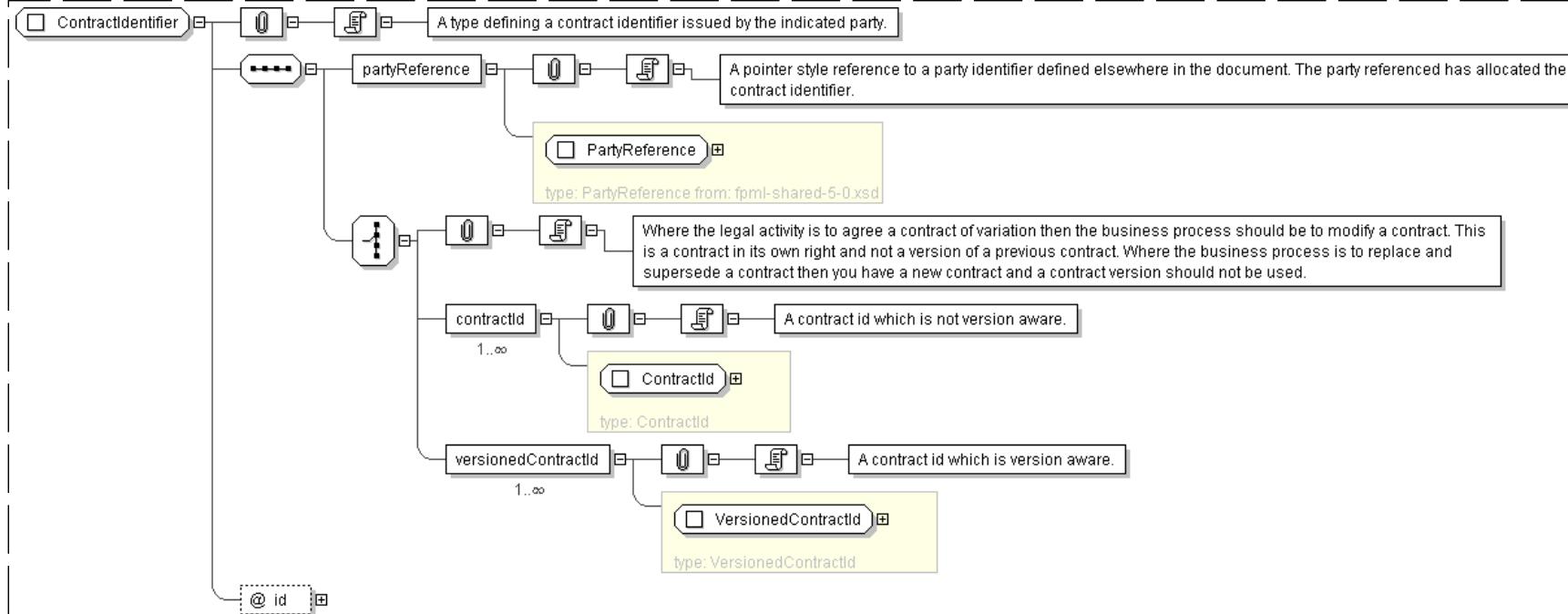
```

<contractId> ContractId </contractId> [1..*]
'A contract id which is not version aware.'

<versionedContractId> VersionedContractId </versionedContractId> [1..*]
'A contract id which is version aware.'

```

End Choice  
/>

**Diagram****Schema Component Representation**

```

<xsd:complexType name="ContractIdentifier">
  <xsd:sequence>
    <xsd:element name="partyReference" type=" PartyReference " />
    <xsd:choice>
      <xsd:element name="contractId" type=" ContractId " maxOccurs="unbounded"/>
      <xsd:element name="versionedContractId" type=" VersionedContractId " maxOccurs="unbounded"/>
    </xsd:choice>
  </xsd:sequence>
  <xsd:attribute name="id" type=" xsd:ID " />
</xsd:complexType>

```

top

**Complex Type: CreditDerivativesNotices**

Super-types:	None
Sub-types:	None

Name	CreditDerivativesNotices
Abstract	no

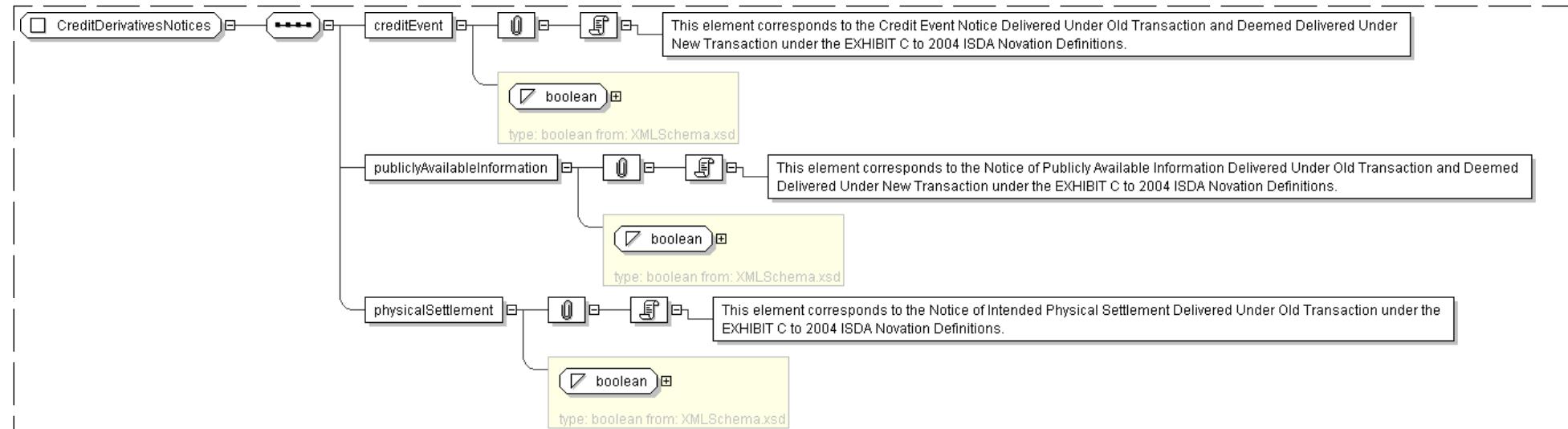
**XML Instance Representation**

```
<...>
<creditEvent> xsd:boolean </creditEvent> [1]
'This element corresponds to the Credit Event Notice Delivered Under Old Transaction and Deemed Delivered Under New Transaction under the EXHIBIT C to 2004 ISDA Novation Definitions.'

<publiclyAvailableInformation> xsd:boolean </publiclyAvailableInformation> [1]
'This element corresponds to the Notice of Publicly Available Information Delivered Under Old Transaction and Deemed Delivered Under New Transaction under the EXHIBIT C to 2004 ISDA Novation Definitions.'

<physicalSettlement> xsd:boolean </physicalSettlement> [1]
'This element corresponds to the Notice of Intended Physical Settlement Delivered Under Old Transaction under the EXHIBIT C to 2004 ISDA Novation Definitions.

</...>
```

**Diagram****Schema Component Representation**

```
<xsd:complexType name="CreditDerivativesNotices">
  <xsd:sequence>
    <xsd:element name="creditEvent" type="xsd:boolean" />
    <xsd:element name="publiclyAvailableInformation" type="xsd:boolean" />
    <xsd:element name="physicalSettlement" type="xsd:boolean" />
  </xsd:sequence>
</xsd:complexType>
```

top

**Complex Type: DataDocument**

Super-types:	<a href="#">Document</a> < <b>DataDocument</b> (by extension)
Sub-types:	None

Name	DataDocument
Abstract	no

**Documentation**

A type defining a content model that is backwards compatible with older FpML releases and which can be used to contain sets of data without expressing any processing intention.

**XML Instance Representation**

```
<...>
  fpmlVersion=" xsd:token (value comes from list: {'5-0'}) [1]
  'Indicate which version of the FpML Schema an FpML message adheres to.'

  "
  expectedBuild=" xsd:positiveInteger [0..1]
  'This optional attribute can be supplied by a message creator in an FpML instance to
  specify which build number of the schema was used to define the message when it was generated.'

  "
  actualBuild="8 [0..1]
  'The specific build number of this schema version. This attribute is not included in
  an instance document. Instead, it is supplied by the XML parser when the document is
  validated against the FpML schema and indicates the build number of the schema file. Every
  time FpML publishes a change to the schema, validation rules, or examples within a version
  (e.g., version 4.2) the actual build number is incremented. If no changes have been
  made between releases within a version (i.e. from Trial Recommendation to Recommendation)
  the actual build number stays the same.'

  ">
    <validation> Validation </validation> [0..*]
  Start Choice [1]
    <trade> Trade </trade> [0..*]
    'The root element in an FpML trade document.'

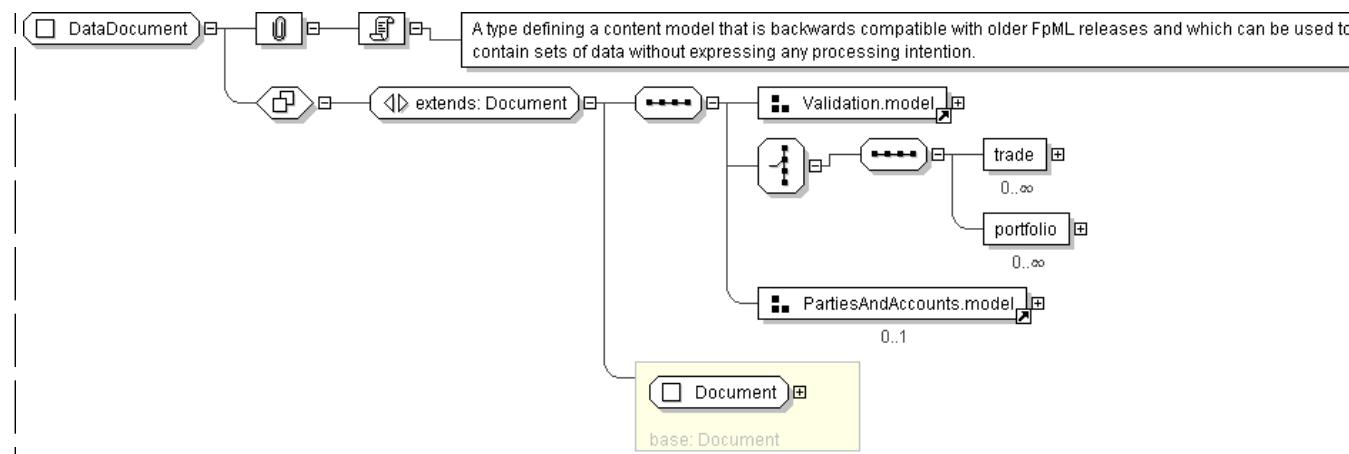
    <portfolio> Portfolio </portfolio> [0..*]
    'An arbitrary grouping of trade references (and possibly other portfolios).'

  End Choice
  Start Group: PartiesAndAccounts.model [0..1]
    <party> Party </party> [1..*]
    'A legal entity or a subdivision of a legal entity.', 'Parties can perform multiple roles in
    a trade lifecycle. For example, the principal parties obligated to make payments from time
    to time during the term of the trade, but may include other parties involved in, or
    incidental to, the trade, such as parties acting in the role of novation transferor/
    transferee, broker, calculation agent, etc. If FpML roles are defined in multiple places
    within a document.'

    <account> Account </account> [0..*]
    'Optional account information used to precisely define the origination and destination
    of financial instruments.'

  End Group: PartiesAndAccounts.model
</...>
```

**Diagram**

**Schema Component Representation**

```

<xsd:complexType name="DataDocument">
  <xsd:complexContent>
    <xsd:extension base=" Document " >
      <xsd:sequence>
        <xsd:group ref=" Validation.model " />
        <xsd:choice>
          <xsd:sequence>
            <xsd:element name="trade" type=" Trade " minOccurs="0" maxOccurs="unbounded" />
            <xsd:element name="portfolio" type=" Portfolio " minOccurs="0" maxOccurs="unbounded" />
          </xsd:sequence>
        </xsd:choice>
        <xsd:group ref=" PartiesAndAccounts.model " minOccurs="0" />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
  
```

[top](#)**Complex Type: Difference**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None
<b>Name</b>	Difference
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">BestFitTrade</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type used to record the details of a difference between two business objects/

**XML Instance Representation**

```

<...>
<differenceType> DifferenceTypeEnum </differenceType> [1]
  'The type of difference that exists.'

<differenceSeverity> DifferenceSeverityEnum </differenceSeverity> [1]
  'An indication of the severity of the difference.'

<element> xsd:string </element> [1]
  'The name of the element affected.'
  
```

```

<basePath> xsd:string </basePath> [0..1]
'XPath to the element in the base object.'

<baseValue> xsd:string </baseValue> [0..1]
'The value of the element in the base object.'

<otherPath> xsd:string </otherPath> [0..1]
'XPath to the element in the other object.'

<otherValue> xsd:string </otherValue> [0..1]
'Value of the element in the other trade.'

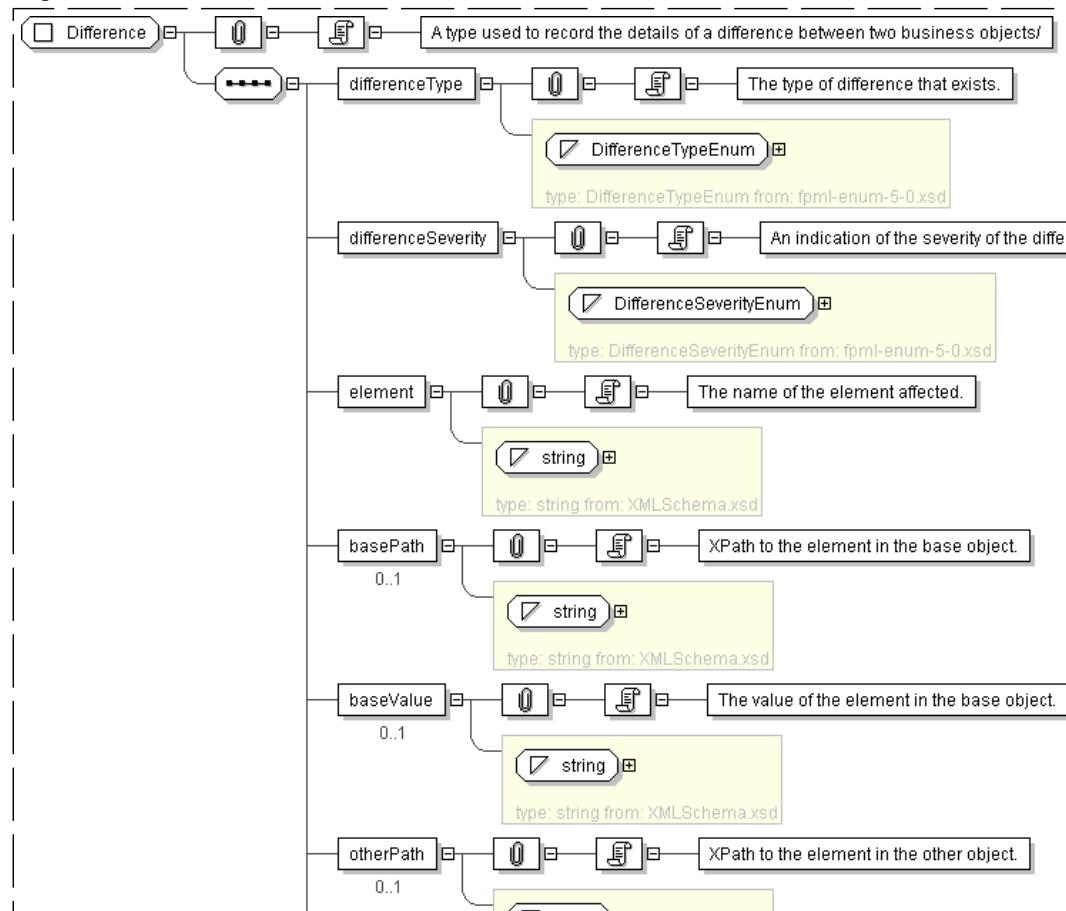
<missingElement> xsd:string </missingElement> [0..*]
'Element(s) that are missing in the other trade.'

<extraElement> xsd:string </extraElement> [0..*]
'Element(s) that are extraneous in the other object.'

<message> xsd:string </message> [1]
'A human readable description of the problem.'

</...>

```

**Diagram**



#### Schema Component Representation

```

<xsd:complexType name="Difference">
  <xsd:sequence>
    <xsd:element name="differenceType" type=" DifferenceTypeEnum " />
    <xsd:element name="differenceSeverity" type=" DifferenceSeverityEnum " />
    <xsd:element name="element" type=" xsd:string " />
    <xsd:element name="basePath" type=" xsd:string " minOccurs="0" />
    <xsd:element name="baseValue" type=" xsd:string " minOccurs="0" />
    <xsd:element name="otherPath" type=" xsd:string " minOccurs="0" />
    <xsd:element name="otherValue" type=" xsd:string " minOccurs="0" />
    <xsd:element name="missingElement" type=" xsd:string " minOccurs="0" maxOccurs="unbounded" />
    <xsd:element name="extraElement" type=" xsd:string " minOccurs="0" maxOccurs="unbounded" />
    <xsd:element name="message" type=" xsd:string " />
  </xsd:sequence>
</xsd:complexType>

```

top

#### Complex Type: Document

Super-types:	None
Sub-types:	<ul style="list-style-type: none"> <li>• <a href="#">DataDocument</a> (by extension)</li> </ul>

Name	Document
Abstract	yes
Documentation	The abstract base type from which all FpML compliant messages and documents must be derived.

#### XML Instance Representation

```

<...
  fpmlVersion=" xsd:token (value comes from list: {'5-0'}) [1]

```

'Indicate which version of the FpML Schema an FpML message adheres to.'

"

**expectedBuild=** "xsd:positiveInteger [0..1]

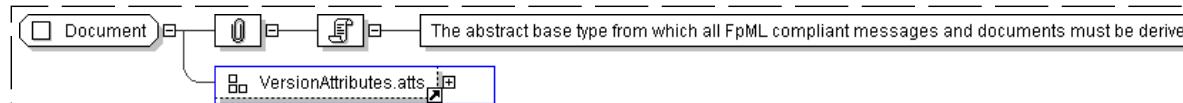
'This optional attribute can be supplied by a message creator in an FpML instance to specify which build number of the schema was used to define the message when it was generated.'

"

**actualBuild=** "8 [0..1]

'The specific build number of this schema version. This attribute is not included in an instance document. Instead, it is supplied by the XML parser when the document is validated against the FpML schema and indicates the build number of the schema file. Every time FpML publishes a change to the schema, validation rules, or examples within a version (e.g., version 4.2) the actual build number is incremented. If no changes have been made between releases within a version (i.e. from Trial Recommendation to Recommendation) the actual build number stays the same.'

"/>

**Diagram****Schema Component Representation**

```

<xsd:complexType name="Document" abstract="true">
  <xsd:attributeGroup ref=" VersionAttributes.atts " />
</xsd:complexType>
  
```

[top](#)**Complex Type: ExecutionDateTime**

**Super-types:** [xsd:dateTime](#) < **ExecutionDateTime** (by extension)

**Sub-types:** None

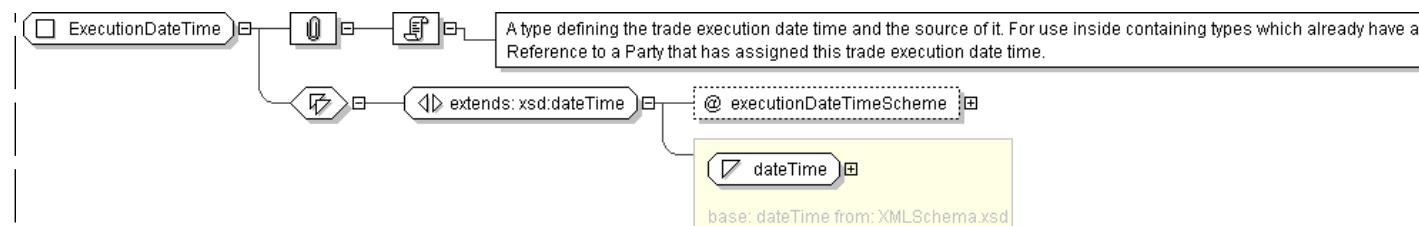
<b>Name</b>	ExecutionDateTime
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">PartyTradeInformation</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type defining the trade execution date time and the source of it. For use inside containing types which already have a Reference to a Party that has assigned this trade execution date time.

**XML Instance Representation**

```

<...
  executionDateTimeScheme=" xsd:anyURI [0..1]
  'Identification of the source (e.g. clock id) generating the execution date time.'
  ">
  xsd:dateTime
</...>
  
```

**Diagram**

**Schema Component Representation**

```

<xsd:complexType name="ExecutionDateTime">
  <xsd:simpleContent>
    <xsd:extension base="xsd:dateTime">
      <xsd:attribute name="executionDateTimeScheme" type="xsd:anyURI"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
  
```

[top](#)**Complex Type: FirstPeriodStartDate**

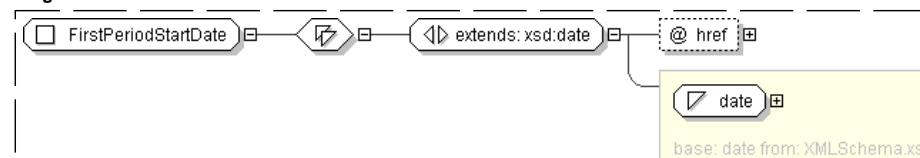
<b>Super-types:</b>	<a href="#">xsd:date</a> < <b>FirstPeriodStartDate</b> (by extension)
<b>Sub-types:</b>	None

<b>Name</b>	FirstPeriodStartDate
<b>Abstract</b>	no

**XML Instance Representation**

```

<...
  href="#>
  xsd:date
<...>
  
```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="FirstPeriodStartDate">
  <xsd:simpleContent>
    <xsd:extension base="xsd:date">
      <xsd:attribute name="href" type="xsd:IDREF" use="required" reference="Party"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
  
```

[top](#)**Complex Type: IndependentAmount**

<b>Super-types:</b>	None
---------------------	------

Sub-types:

None

**Name**IndependentAmount  
Used by (from the same schema document) Complex Type [Collateral](#)**Abstract**

no

**XML Instance Representation**

```
<...>
<payerPartyReference> PartyReference </payerPartyReference> [1]
'A reference to the party responsible for making the payments defined by this structure.'

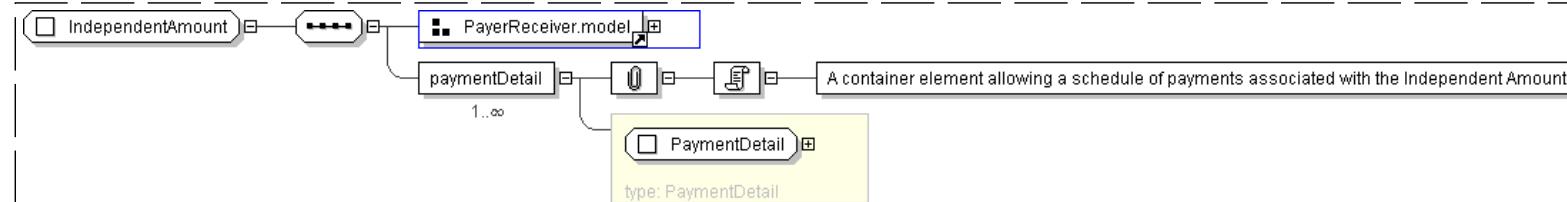
<payerAccountReference> AccountReference </payerAccountReference> [0..1]
'A reference to the account responsible for making the payments defined by this structure.'

<receiverPartyReference> PartyReference </receiverPartyReference> [1]
'A reference to the party that receives the payments corresponding to this structure.'

<receiverAccountReference> AccountReference </receiverAccountReference> [0..1]
'A reference to the account that receives the payments corresponding to this structure.'

<paymentDetail> PaymentDetail </paymentDetail> [1..*]
'A container element allowing a schedule of payments associated with the Independent Amount.'
```

&lt;/...&gt;

**Diagram****Schema Component Representation**

```
<xsd:complexType name="IndependentAmount">
  <xsd:sequence>
    <xsd:group ref="PayerReceiver.model" />
    <xsd:element name="paymentDetail" type="PaymentDetail" maxOccurs="unbounded" />
  </xsd:sequence>
</xsd:complexType>
```

top

**Complex Type: LinkId**

Super-types:

[Scheme](#) < [LinkId](#) (by extension)

Sub-types:

None

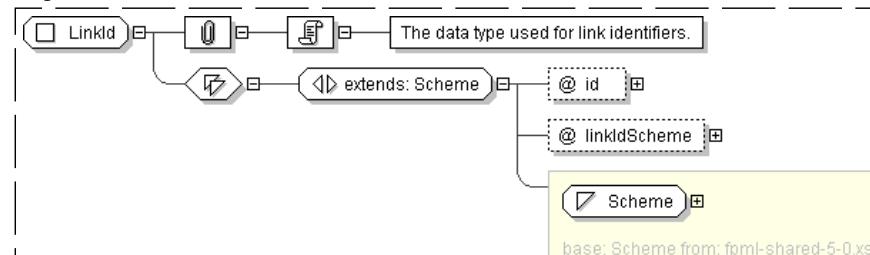
**Name**LinkId  
Used by (from the same schema document) Complex Type [PartyTradeIdentifier](#)**Abstract**

no

**Documentation**  
The data type used for link identifiers.**XML Instance Representation**

```
<...>
```

```
| id=" xsd:ID [0..1]"
| linkIdsScheme=" xsd:anyURI [1]">
| Scheme
| </...>
```

**Diagram****Schema Component Representation**

```
<xsd:complexType name="LinkId">
  <xsd:simpleContent>
    <xsd:extension base=" Scheme ">
      <xsd:attribute name="id" type=" xsd:ID " />
      <xsd:attribute name="linkIdsScheme" type=" xsd:anyURI " use="required" />
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
```

[top](#)**Complex Type: PartyPortfolioName**

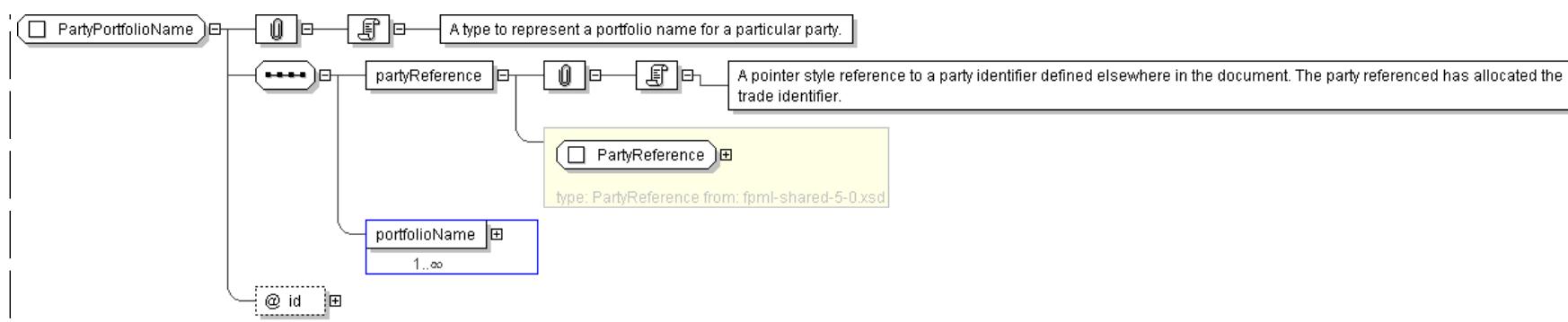
<b>Super-types:</b>	None
<b>Sub-types:</b>	None

<b>Name</b>	PartyPortfolioName
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">Portfolio</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type to represent a portfolio name for a particular party.

**XML Instance Representation**

```
<...
id=" xsd:ID [0..1]">
<partyReference> PartyReference </partyReference> [1]
' A pointer style reference to a party identifier defined elsewhere in the document. The
party referenced has allocated the trade identifier.'
<portfolioName> PortfolioName </portfolioName> [1..*]
</...>
```

**Diagram**

**Schema Component Representation**

```

<xsd:complexType name="PartyPortfolioName">
  <xsd:sequence>
    <xsd:element name="partyReference" type="PartyReference" />
    <xsd:element name="portfolioName" type="PortfolioName" maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:attribute name="id" type="xsd:ID" />
</xsd:complexType>
  
```

[top](#)**Complex Type: PartyTradeIdentifier**

Super-types:	<a href="#">TradeIdentifier</a> < <b>PartyTradeIdentifier</b> (by extension)
Sub-types:	None

Name	PartyTradeIdentifier
Used by (from the same schema document)	Complex Type <a href="#">PartyTradeIdentifiers</a> , Complex Type <a href="#">TradeHeader</a>
Abstract	no
Documentation	A type defining one or more trade 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 trade there will be only one tradeld sent in the document per party.

**XML Instance Representation**

```

<...
  id="xsd:ID [0..1]">
  <partyReference> PartyReference </partyReference> [1]
  'Reference to a party.'

  <accountReference> AccountReference </accountReference> [0..1]
  'Reference to an account.'

Start Choice [1..*]
  <tradeId> TradeId </tradeId> [1]
  <versionedTradeId> VersionedTradeId </versionedTradeId> [1]
End Choice
  <linkId> LinkId </linkId> [0..*]

'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.'

<allocationTradeId> TradeIdentifier </allocationTradeId> [0..*]
  'The trade id of the allocated trade. This is used by the block trade to reference
  '
  
```

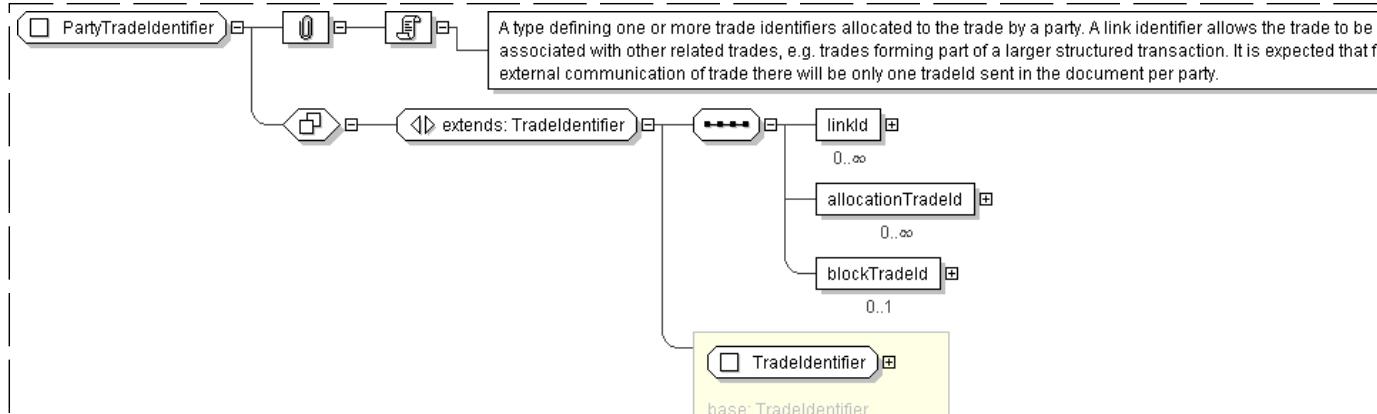
*the allocated trade.'*

<blockTradeId> TradeIdentifier </blockTradeId> [0..1]

'The trade id of the block trade. This is used by each one of the allocated trades to reference the block trade. This element can also represent the trade id of the parent trade for N-level allocations. In the case, this element is only used to model N-level allocations in which the trade acts as block and allocated trade at the same time. This basically means the ability to allocate a block trade to multiple allocation trades, and then allocate these in turn to other allocation trades (and so on if desired).'

<...>

#### Diagram



#### Schema Component Representation

```

<xsd:complexType name="PartyTradeIdentifier">
  <xsd:complexContent>
    <xsd:extension base=" TradeIdentifier ">
      <xsd:sequence>
        <xsd:element name="linkID" type=" LinkId " minOccurs="0" maxOccurs="unbounded" />
        <xsd:element name="allocationTradeID" type=" TradeIdentifier "
          minOccurs="0" maxOccurs="unbounded" />
        <xsd:element name="blockTradeID" type=" TradeIdentifier " minOccurs="0" />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
  
```

top

#### Complex Type: PartyTradeIdentifiers

Super-types:	None
Sub-types:	None

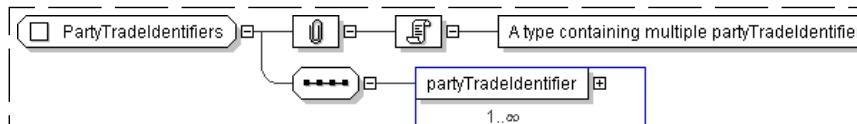
Name	PartyTradeIdentifiers
Used by (from the same schema document)	Model Group <a href="#">TradeOrTradeReference.model</a>
Abstract	no
Documentation	A type containing multiple partyTradelid.

#### XML Instance Representation

```

<...>
<partyTradeIdentifier> PartyTradeIdentifier </partyTradeIdentifier> [1..*]
  
```

&lt; / ... &gt;

**Diagram****Schema Component Representation**

```

<xsd:complexType name="PartyTradeIdentifiers">
  <xsd:sequence>
    <xsd:element name="partyTradeIdentifier" type="PartyTradeIdentifier" maxOccurs="unbounded" />
  </xsd:sequence>
</xsd:complexType>
  
```

[top](#)**Complex Type: PartyTradeInformation**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None

<b>Name</b>	PartyTradeInformation
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">TradeHeader</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type defining additional information that may be recorded against a trade.

**XML Instance Representation**

```

<...>
<partyReference> PartyReference </partyReference> [1]
'Reference to a party.'

<accountReference> AccountReference </accountReference> [0..1]
'Reference to an account.'

<relatedParty> RelatedParty </relatedParty> [0..*]
'Identifies a related party performing a role within the transaction.'

<unit> Unit </unit> [0..*]
'Identifies the unit/division/desk etc. that executed or supports this trade'

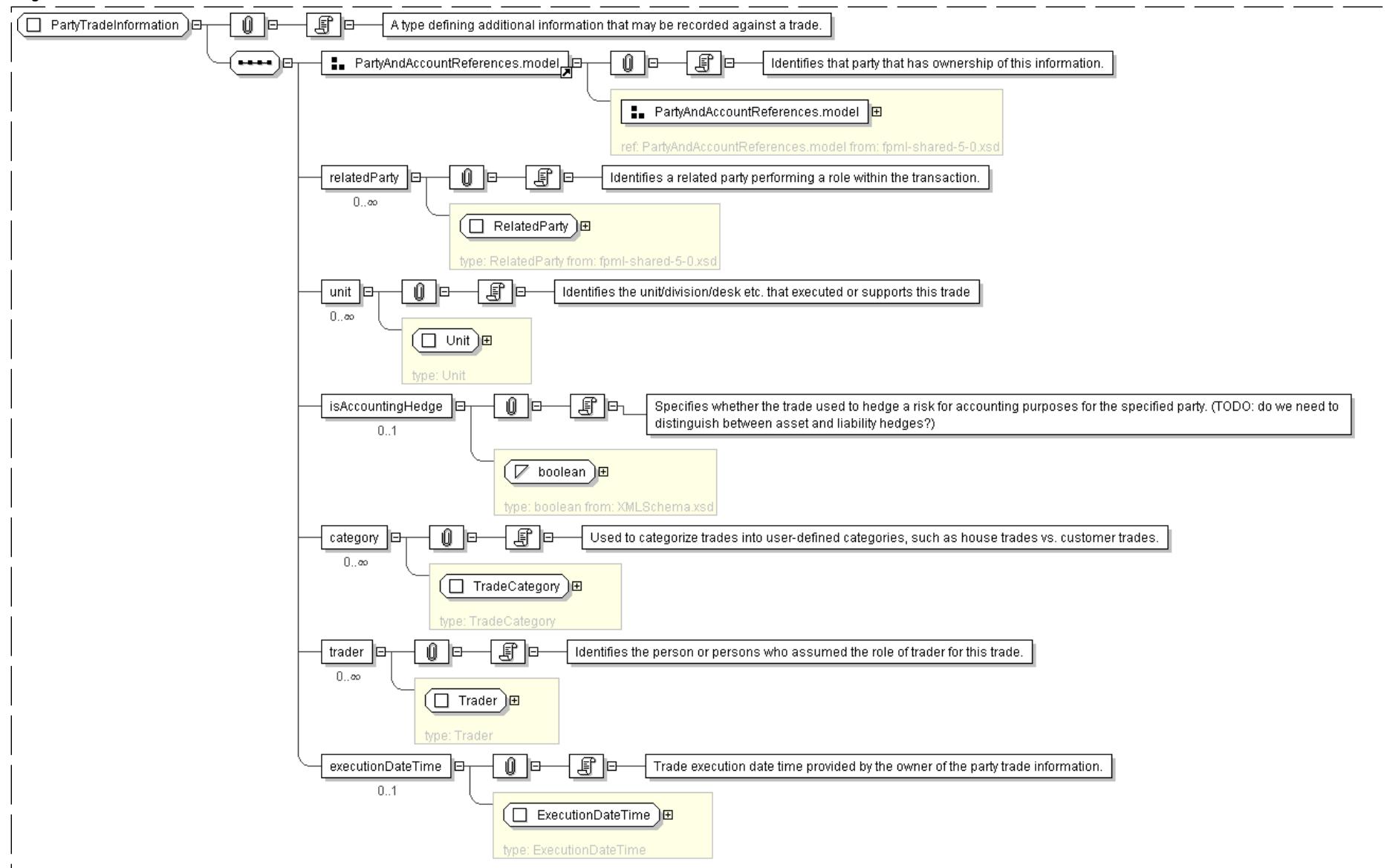
<isAccountingHedge> xsd:boolean </isAccountingHedge> [0..1]
'Specifies whether the trade used to hedge a risk for accounting purposes for the
specified party. (TODO: do we need to distinguish between asset and liability hedges?)'

<category> TradeCategory </category> [0..*]
'Used to categorize trades into user-defined categories, such as house trades vs.
customer trades.'

<trader> Trader </trader> [0..*]
'Identifies the person or persons who assumed the role of trader for this trade.'

<executionDateTime> ExecutionDateTime </executionDateTime> [0..1]
'Trade execution date time provided by the owner of the party trade information.'

</...>
  
```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="PartyTradeInformation">
  <xsd:sequence>
    <xsd:group ref=" PartyAndAccountReferences.model "/>
    <xsd:element name="relatedParty" type=" RelatedParty " minOccurs="0" maxOccurs="unbounded ">
      <!-- RPTWG 2009-09-04 begin-->
    <xsd:element name="unit" type=" Unit " minOccurs="0" maxOccurs="unbounded ">
    <xsd:element name="isAccountingHedge" type=" xsd:boolean " minOccurs="0 ">
    <xsd:element name="category" type=" TradeCategory " minOccurs="0" maxOccurs="unbounded ">
      <!-- RPTWG 2009-09-04 end -->
    <xsd:element name="trader" type=" Trader " minOccurs="0" maxOccurs="unbounded ">
    <xsd:element name="executionDateTime" type=" ExecutionDateTime " minOccurs="0 ">
  </xsd:sequence>

```

**Complex Type: PaymentDetail**

**Super-types:** [PaymentBase](#) < **PaymentDetail** (by extension)  
**Sub-types:** None

<b>Name</b>	PaymentDetail
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">IndependentAmount</a>
<b>Abstract</b>	no

**XML Instance Representation**

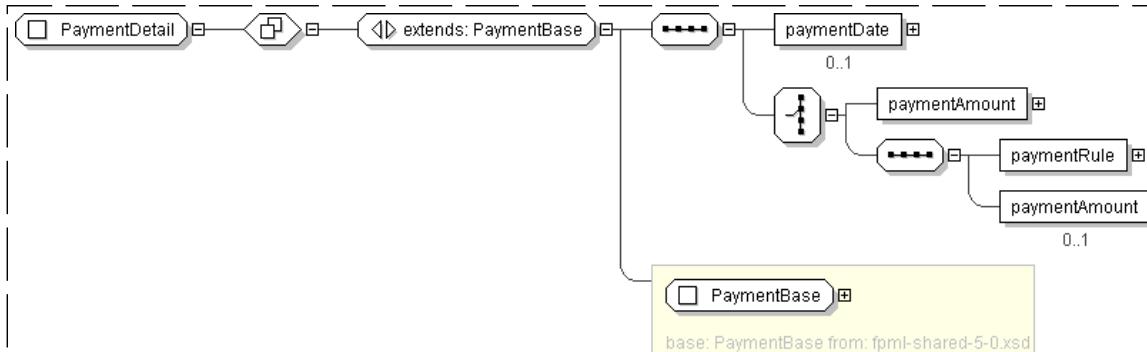
```
<...>
  id=" xsd:ID [0..1]">
    <paymentDate> AdjustableOrRelativeDate </paymentDate> [0..1]
    'Payment date.'

Start Choice [1]
  <paymentAmount> Money </paymentAmount> [1]
  'A fixed payment amount.'

  <paymentRule> PaymentRule </paymentRule> [1]
  'A type defining the calculation rule.'

  <paymentAmount> Money </paymentAmount> [0..1]
  'A fixed payment amount.'

End Choice
</...>
```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="PaymentDetail">
  <xsd:complexContent>
    <xsd:extension base=" PaymentBase ">
      <xsd:sequence>
        <xsd:element name="paymentDate" type=" AdjustableOrRelativeDate " minOccurs="0" />
        <xsd:choice>
          <xsd:element name="paymentAmount" type=" Money " />
          <xsd:sequence>
            <xsd:element name="paymentRule" type=" PaymentRule " />
          </xsd:sequence>
        </xsd:choice>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
  
```

```

<xsd:element name="paymentAmount" type="Money" minOccurs="0" />
</xsd:sequence>
</xsd:choice>
</xsd:sequence>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>

```

[top](#)

## Complex Type: PaymentRule

Super-types:

None

Sub-types:

- [PercentageRule](#) (by extension)

**Name**

PaymentRule

**Used by (from the same schema document)**Complex Type [PaymentDetail](#)**Abstract**

yes

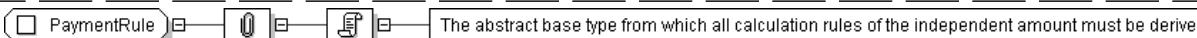
**Documentation**

The abstract base type from which all calculation rules of the independent amount must be derived.

### XML Instance Representation

&lt;.../&gt;

### Diagram



### Schema Component Representation

```
<xsd:complexType name="PaymentRule" abstract="true"/>
```

[top](#)

## Complex Type: PercentageRule

Super-types:

[PaymentRule](#) < [PercentageRule](#) (by extension)

Sub-types:

None

**Name**

PercentageRule

**Abstract**

no

**Documentation**

A type defining a content model for a calculation rule defined as percentage of the notional amount.

### XML Instance Representation

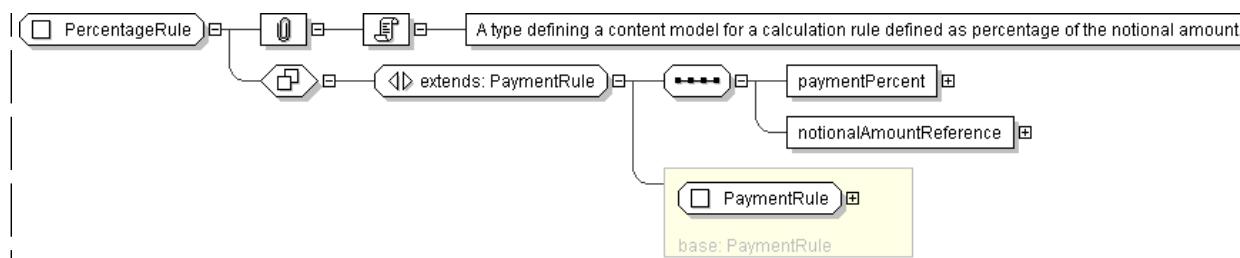
```

<...>
<paymentPercent> xsd:decimal </paymentPercent> [1]
  'A percentage of the notional amount.'
<notionalAmountReference> NotionalAmountReference </notionalAmountReference> [1]
  'A reference to the notional amount.'
</...>

```

### Diagram



**Schema Component Representation**

```

<xsd:complexType name="PercentageRule">
  <xsd:complexContent>
    <xsd:extension base=" PaymentRule " >
      <xsd:sequence>
        <xsd:element name="paymentPercent" type=" xsd:decimal " />
        <xsd:element name="notionalAmountReference" type=" NotionalAmountReference " />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
  
```

top

**Complex Type: Portfolio**

Super-types:

None

Sub-types:

- [QueryPortfolio](#) (by extension)

**Name**

Portfolio

**Used by (from the same schema document)**Complex Type [DataDocument](#), Complex Type [Portfolio](#)**Abstract**

no

**Documentation**

A type representing an arbitrary grouping of trade references.

**XML Instance Representation**

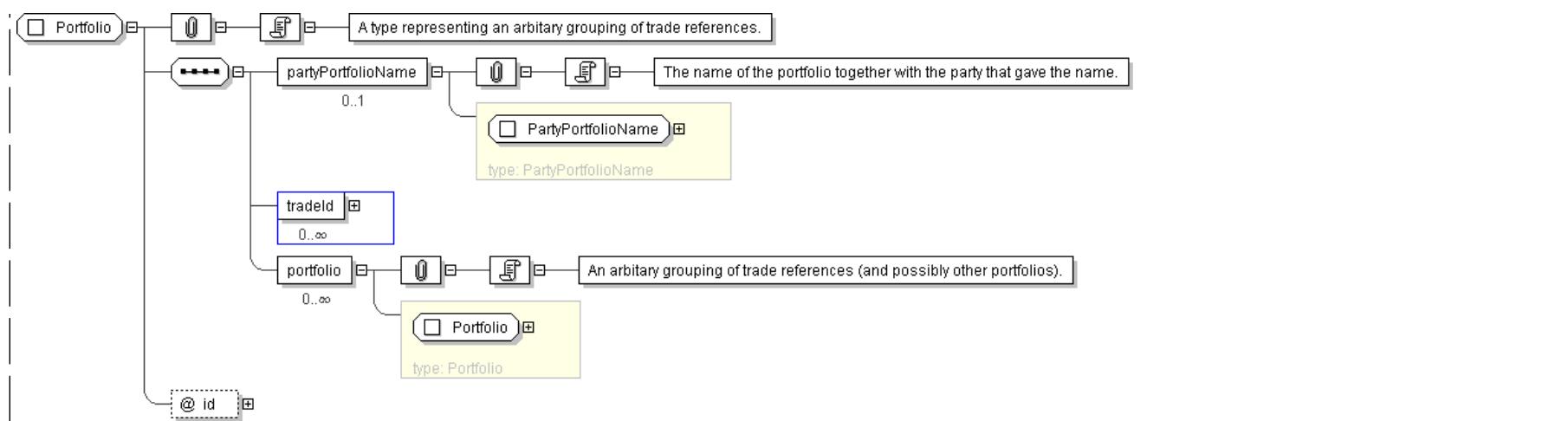
```

<...
  id=" xsd:ID [0..1]">
  <partyPortfolioName> PartyPortfolioName </partyPortfolioName> [0..1]
  'The name of the portfolio together with the party that gave the name.'

  <tradeId> TradeId </tradeId> [0..*]
  <portfolio> Portfolio </portfolio> [0..*]
  'An arbitrary grouping of trade references (and possibly other portfolios).'

</...>
  
```

**Diagram**



#### Schema Component Representation

```

<xsd:complexType name="Portfolio">
  <xsd:sequence>
    <xsd:element name="partyPortfolioName" type="PartyPortfolioName" minOccurs="0"/>
    <xsd:element name="tradeId" type="TradeId" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element name="portfolio" type="Portfolio" minOccurs="0" maxOccurs="unbounded"/>
  </xsd:sequence>
  <xsd:attribute name="id" type="xsd:ID" />
</xsd:complexType>

```

top

#### Complex Type: `PortfolioName`

Super-types:	<a href="#">Scheme</a> < <b>PortfolioName</b> (by extension)
Sub-types:	None
Name	PortfolioName
Used by (from the same schema document)	Complex Type <a href="#">PartyPortfolioName</a>
Abstract	no
Documentation	The data type used for portfolio names.

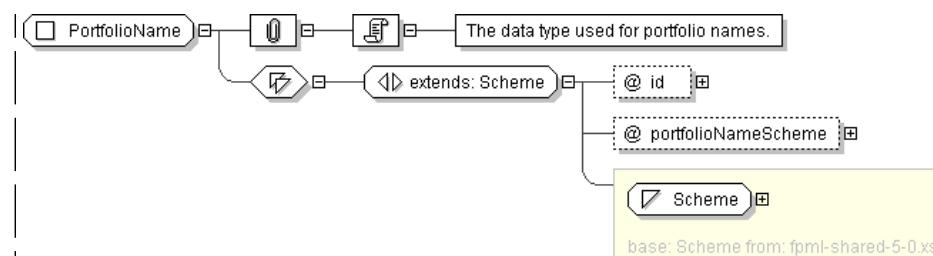
#### XML Instance Representation

```

<...
  id="xsd:ID [0..1]"
  portfolioNameScheme="xsd:anyURI [0..1]">
  Scheme
</...>

```

#### Diagram

**Schema Component Representation**

```

<xsd:complexType name="PortfolioName">
  <xsd:simpleContent>
    <xsd:extension base=" Scheme ">
      <xsd:attribute name="id" type=" xsd:ID ">
      <xsd:attribute name="portfolioNameScheme" type=" xsd:anyURI ">
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
  
```

[top](#)**Complex Type: QueryParameter**

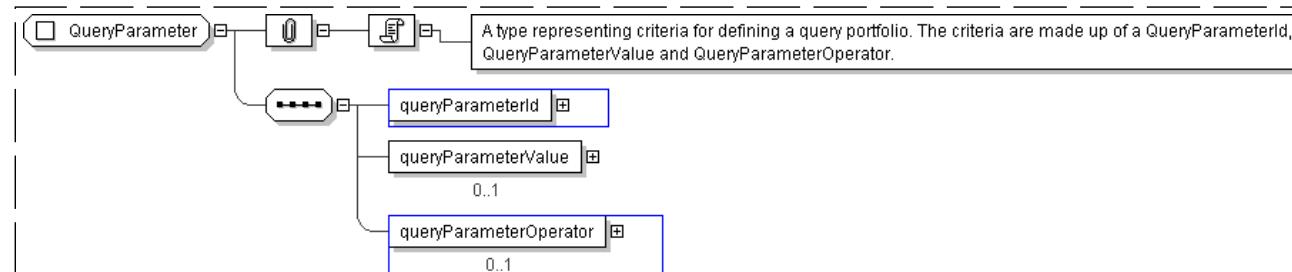
<b>Super-types:</b>	None
<b>Sub-types:</b>	None

<b>Name</b>	QueryParameter
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">QueryPortfolio</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type representing criteria for defining a query portfolio. The criteria are made up of a <code>QueryParameterId</code> , <code>QueryParameterValue</code> and <code>QueryParameterOperator</code> .

**XML Instance Representation**

```

<...>
  <queryParameterId> QueryParameterId </queryParameterId> [1]
  <queryParameterValue> xsd:normalizedString </queryParameterValue> [0..1]
  <queryParameterOperator> QueryParameterOperator </queryParameterOperator> [0..1]
</...>
  
```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="QueryParameter">
  <xsd:sequence>
    <xsd;element name="queryParameterId" type=" QueryParameterId ">
  </xsd:sequence>
</xsd:complexType>
  
```

```

<xsd:element name="queryParameterValue" type="xsd:normalizedString" minOccurs="0"/>
<xsd:element name="queryParameterOperator" type="QueryParameterOperator" minOccurs="0"/>
</xsd:sequence>
</xsd:complexType>

```

## Complex Type: QueryParameterId

Super-types:	<a href="#">Scheme</a> < <b>QueryParameterId</b> (by extension)
Sub-types:	None

Name	QueryParameterId
Used by (from the same schema document)	Complex Type <a href="#">QueryParameter</a>
Abstract	no
Documentation	A type representing an identifier for a parameter describing a query portfolio. An identifier can be anything from a product name like swap to a termination date.

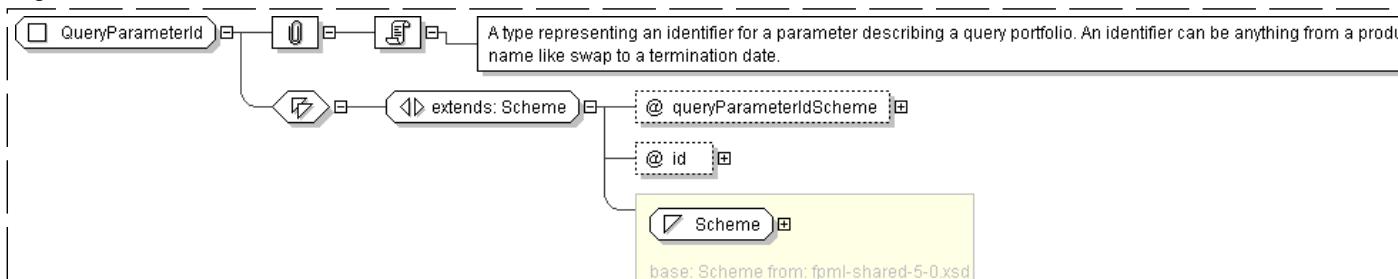
### XML Instance Representation

```

<...
queryParameterIdScheme="xsd:anyURI [1]"
id="xsd:ID [0..1]">
Scheme
</...>

```

### Diagram



### Schema Component Representation

```

<xsd:complexType name="QueryParameterId">
  <xsd:simpleContent>
    <xsd:extension base="Scheme">
      <xsd:attribute name="queryParameterIdScheme" type="xsd:anyURI" use="required"/>
      <xsd:attribute name="id" type="xsd:ID" />
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>

```

## Complex Type: QueryParameterOperator

Super-types:	<a href="#">Scheme</a> < <b>QueryParameterOperator</b> (by extension)
Sub-types:	None

Name	QueryParameterOperator
Used by (from the same schema document)	Complex Type <a href="#">QueryParameter</a>

**Abstract**

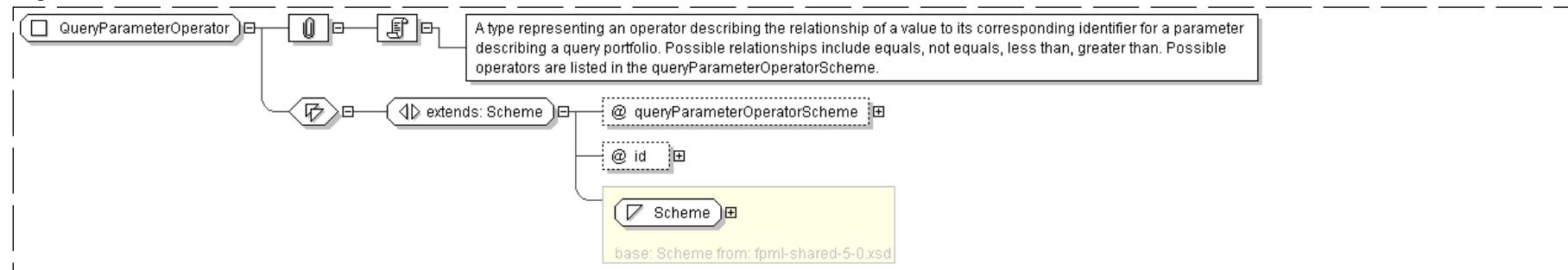
no

**Documentation**

A type representing an operator describing the relationship of a value to its corresponding identifier for a parameter describing a query portfolio. Possible relationships include equals, not equals, less than, greater than. Possible operators are listed in the queryParameterOperatorScheme.

**XML Instance Representation**

```
<...
queryParameterOperatorScheme=" xsd:anyURI [0..1]"
id=" xsd:ID [0..1]">
Scheme
</...>
```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="QueryParameterOperator">
  <xsd:simpleContent>
    <xsd:extension base=" Scheme ">
      <xsd:attribute name="queryParameterOperatorScheme" type=" xsd:anyURI " default="http://www.
        fpmml.org/coding-scheme/query-parameter-operator"/>
      <xsd:attribute name="id" type=" xsd:ID "/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>

```

top

**Complex Type: QueryPortfolio**

Super-types:

[Portfolio](#) < **QueryPortfolio** (by extension)

Sub-types:

None

**Name**

QueryPortfolio

**Abstract**

no

**Documentation**

A type representing a portfolio obtained by querying the set of trades held in a repository. It contains trades matching the intersection of all criteria specified using one or more queryParameters or trades matching the union of two or more child queryPortfolios.

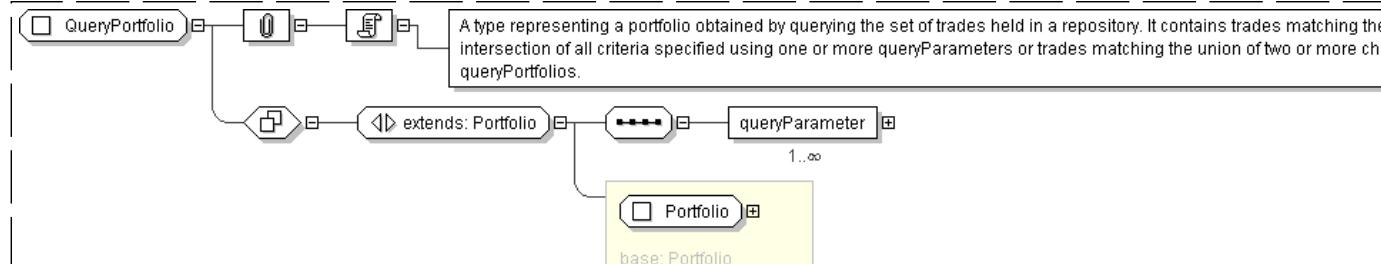
**XML Instance Representation**

```

<...
id=" xsd:ID [0..1]">
<partyPortfolioName> PartyPortfolioName </partyPortfolioName> [0..1]
'The name of the portfolio together with the party that gave the name.'
<tradeId> TradeId </tradeId> [0..*]
<portfolio> Portfolio </portfolio> [0..*]
'An arbitrary grouping of trade references (and possibly other portfolios).'

```

```
<queryParameter> QueryParameter </queryParameter> [1..*]
</...>
```

**Diagram****Schema Component Representation**

```
<xsd:complexType name="QueryPortfolio">
  <xsd:complexContent>
    <xsd:extension base="#Portfolio">
      <xsd:sequence>
        <xsd:element name="queryParameter" type="QueryParameter" maxOccurs="unbounded"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

top

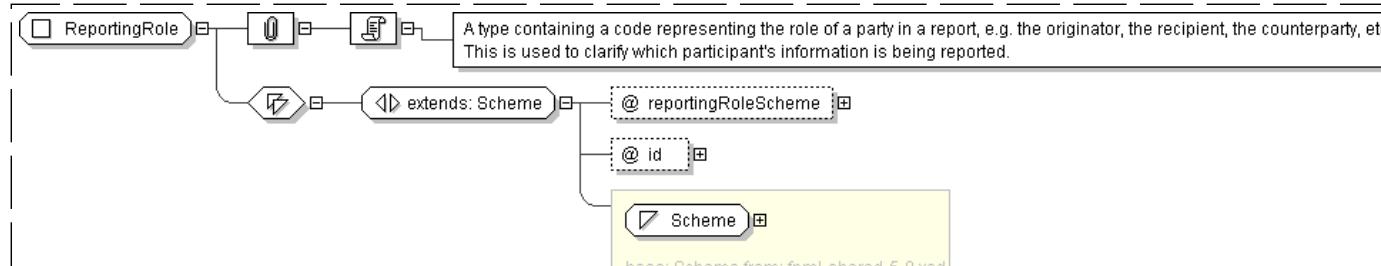
**Complex Type: ReportingRole**

Super-types:	<a href="#">Scheme</a> < <b>ReportingRole</b> (by extension)
Sub-types:	None

Name	ReportingRole
Abstract	no
Documentation	A type containing a code representing the role of a party in a report, e.g. the originator, the recipient, the counterparty, etc. This is used to clarify which participant's information is being reported.

**XML Instance Representation**

```
<...
  reportingRoleScheme="xsd:anyURI [0..1]"
  id="#ID [0..1]">
  Scheme
</...>
```

**Diagram**

**Schema Component Representation**

```
<xsd:complexType name="ReportingRole">
  <xsd:simpleContent>
    <xsd:extension base="Scheme">
      <xsd:attribute name="reportingRoleScheme" type="xsd:anyURI" use="optional"/>
      <xsd:attribute name="id" type="xsd:ID"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
```

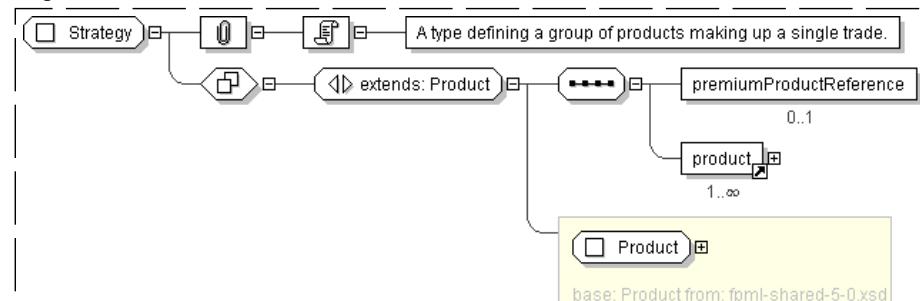
[top](#)**Complex Type: Strategy**

<b>Super-types:</b>	<a href="#">Product</a> < <b>Strategy</b> (by extension)
<b>Sub-types:</b>	None

<b>Name</b>	Strategy
<b>Used by (from the same schema document)</b>	Element <a href="#">strategy</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type defining a group of products making up a single trade.

**XML Instance Representation**

```
<...>
  id=" xsd:ID [0..1]">
  <productType> ProductType </productType> [0..*]
  'A classification of the type of product. FpML defines a simple product categorization using
  a coding scheme.'
  <productId> ProductId </productId> [0..*]
  'A product 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.'
  <premiumProductReference> ProductReference </premiumProductReference> [0..1]
  'Indicates which product within a strategy represents the premium payment.'
  <product> ... </product> [1..*]
</...>
```

**Diagram****Schema Component Representation**

```
<xsd:complexType name="Strategy">
  <xsd:complexContent>
    <xsd:extension base="Product">
```

```

<xsd:sequence>
  <xsd:element name="premiumProductReference" type=" ProductReference " minOccurs="0" />
  <xsd:element ref=" product " maxOccurs="unbounded" />
</xsd:sequence>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>

```

[top](#)

## Complex Type: [Trade](#)

<b>Super-types:</b>	None
<b>Sub-types:</b>	None

<b>Name</b>	Trade
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">DataDocument</a> , Model Group <a href="#">TradeOrTradeReference.model</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type defining an FpML trade.

### XML Instance Representation

```

<...>
  <id=" xsd:ID [0..1]">
    <tradeHeader> TradeHeader </tradeHeader> [1]
    'The information on the trade which is not product specific, e.g. trade date.'

    <product> ... </product> [1]
    <otherPartyPayment> Payment </otherPartyPayment> [0..*]
    '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.'

    <brokerPartyReference> PartyReference </brokerPartyReference> [0..*]
    'Identifies that party (or parties) that brokered this trade.'

    <calculationAgent> CalculationAgent </calculationAgent> [0..1]
    'The ISDA calculation agent responsible for performing duties as defined in the
    applicable product definitions.'

    <calculationAgentBusinessCenter> BusinessCenter </calculationAgentBusinessCenter> [0..1]
    'The city in which the office through which ISDA Calculation Agent is acting for purposes
    of the transaction is located. The short-form confirm for a trade that is executed under
    a Sovereign or Asia Pacific Master Confirmation Agreement ( MCA ), does not need to specify
    the Calculation Agent. However, the confirm does need to specify the Calculation Agent
    City. This is due to the fact that the MCA sets the value for Calculation Agent but does
    not set the value for Calculation Agent City.'

    <determiningParty> PartyReference </determiningParty> [0..2]
    'The party referenced is the ISDA Determination Party that specified in the
    related Confirmation as Determination Party.'

    <hedgingParty> PartyReference </hedgingParty> [0..2]
    'The party referenced is the ISDA Hedging Party that specified in the related Confirmation
    as Hedging, or if no Hedging Party is specified, either party to the Transaction.'

    <collateral> Collateral </collateral> [0..1]
    'Defines collateral obligations of a Party'

    <documentation> Documentation </documentation> [0..1]
    'Defines the definitions that govern the document and should include the year and type
    of definitions referenced, along with any relevant documentation (such as master agreement)

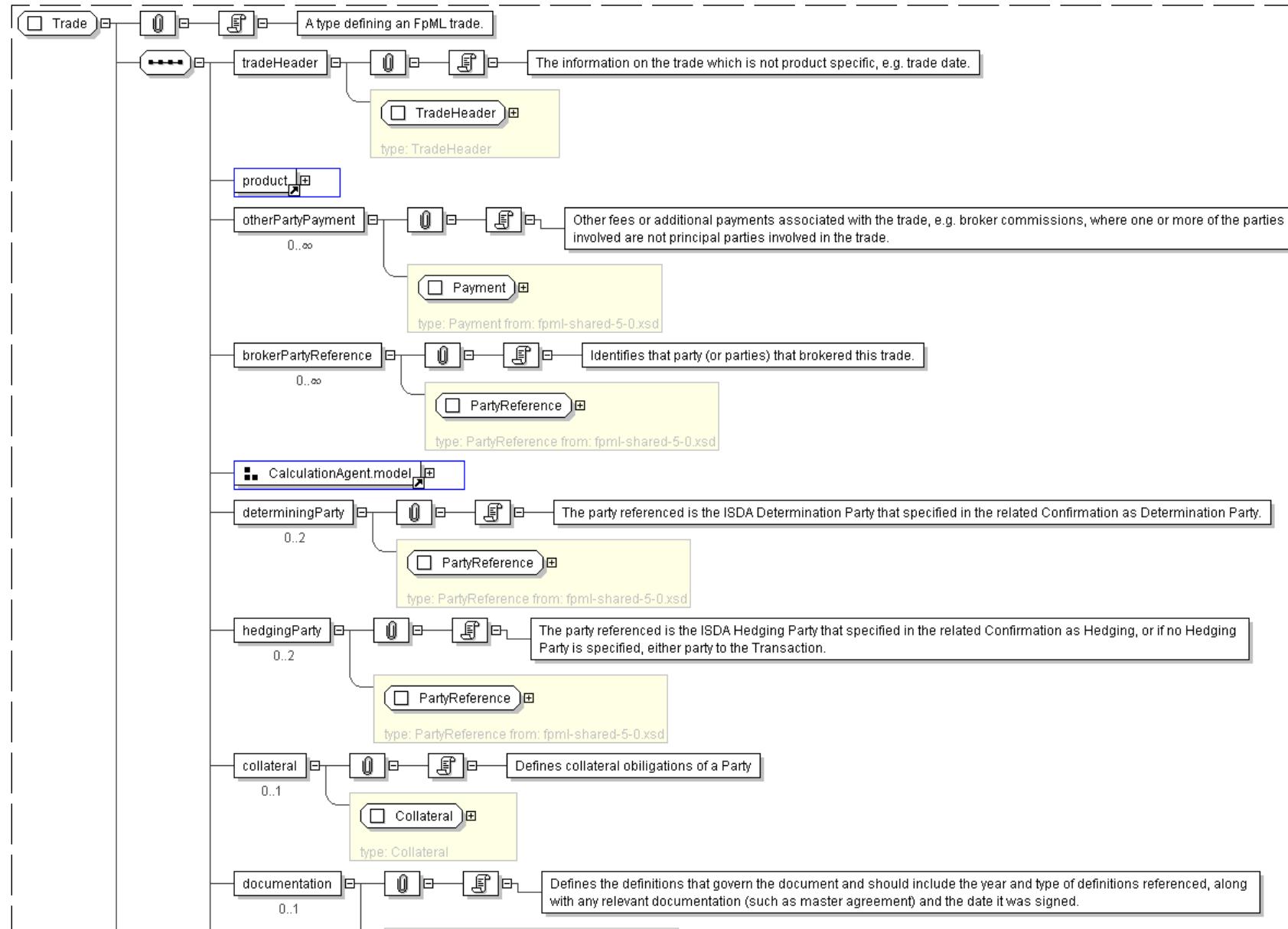
```

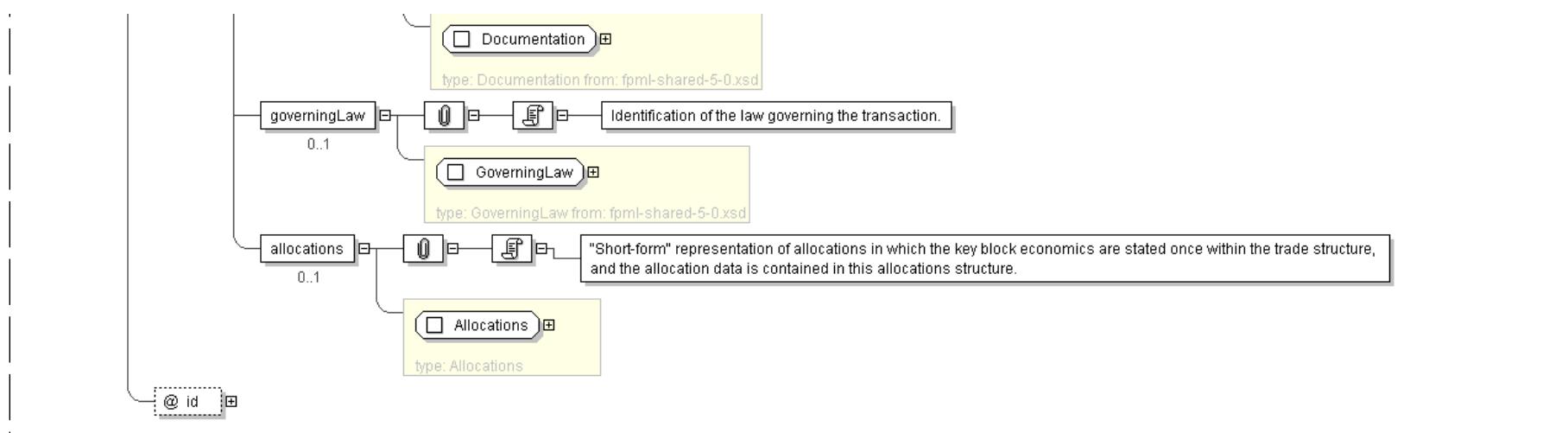
*and the date it was signed.'*

<governingLaw> GoverningLaw </governingLaw> [0..1]  
*'Identification of the law governing the transaction.'*

<allocations> Allocations </allocations> [0..1]  
*"Short-form" representation of allocations in which the key block economics are stated once within the trade structure, and the allocation data is contained in this allocations structure.'*

</...>

**Diagram**



### Schema Component Representation

```

<xsd:complexType name="Trade">
  <xsd:sequence>
    <xsd;element name="tradeHeader" type=" TradeHeader " />
    <xsd;element ref=" product "/>
    <xsd;element name="otherPartyPayment" type=" Payment " minOccurs="0" maxOccurs="unbounded"/>
    <xsd;element name="brokerPartyReference" type=" PartyReference "
      minOccurs="0" maxOccurs="unbounded"/>
    <xsd:group ref=" CalculationAgent.model " />
    <xsd;element name="determiningParty" type=" PartyReference " minOccurs="0" maxOccurs="2"/>
    <xsd;element name="hedgingParty" type=" PartyReference " minOccurs="0" maxOccurs="2"/>
    <xsd;element name="collateral" type=" Collateral " minOccurs="0" />
    <xsd;element name="documentation" type=" Documentation " minOccurs="0"/>
    <xsd;element name="governingLaw" type=" GoverningLaw " minOccurs="0"/>
    <xsd;element name="allocations" type=" Allocations " minOccurs="0" />
  </xsd:sequence>
  <xsd:attribute name="id" type=" xsd:ID "/>
</xsd:complexType>

```

top

### Complex Type: TradeCategory

Super-types:	<a href="#">Scheme</a> < <b>TradeCategory</b> (by extension)
Sub-types:	None

Name	TradeCategory
Used by (from the same schema document)	Complex Type <a href="#">PartyTradeInformation</a>
Abstract	no
Documentation	A scheme used to categorize positions.

### XML Instance Representation

```

<...
  categoryScheme=" xsd:anyURI [0..1]">
  Scheme
</...>

```

### Diagram

**Schema Component Representation**

```

<xsd:complexType name="TradeCategory">
  <xsd:simpleContent>
    <xsd:extension base=" Scheme ">
      <xsd:attribute name="categoryScheme" type=" xsd:anyURI " />
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>

```

top

**Complex Type: TradeHeader**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None

<b>Name</b>	TradeHeader
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">Trade</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type defining trade related information which is not product specific.

**XML Instance Representation**

```

<...>
<partyTradeIdentifier> PartyTradeIdentifier </partyTradeIdentifier> [1..*]
'The trade reference identifier(s) allocated to the trade by the parties involved.'

<partyTradeInformation> PartyTradeInformation </partyTradeInformation> [0..*]
'Additional trade information that may be provided by each involved party.'

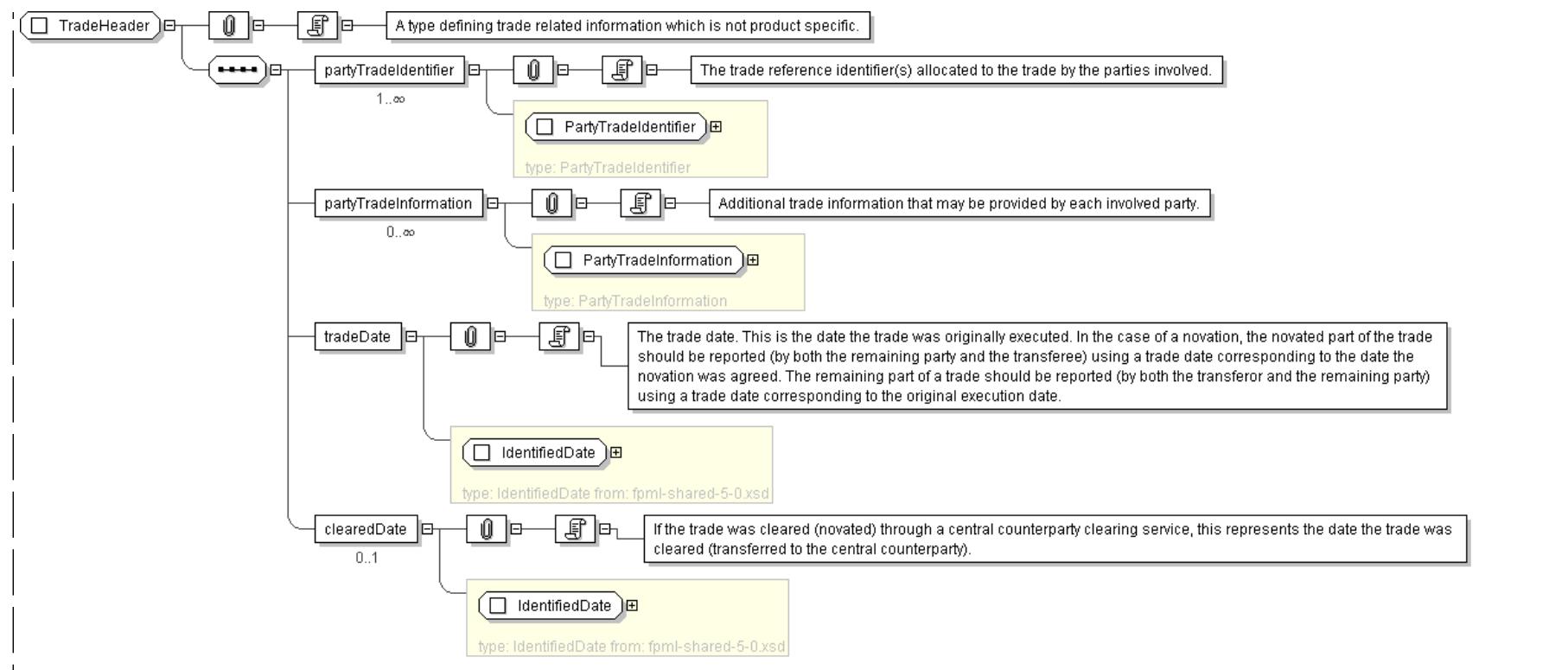
<tradeDate> IdentifiedDate </tradeDate> [1]
'The trade date. This is the date the trade was originally executed. In the case of a novation, the novated part of the trade should be reported (by both the remaining party and the transferee) using a trade date corresponding to the date the novation was agreed. The remaining part of a trade should be reported (by both the transferor and the remaining party) using a trade date corresponding to the original execution date.'

<clearedDate> IdentifiedDate </clearedDate> [0..1]
'If the trade was cleared (novated) through a central counterparty clearing service, this represents the date the trade was cleared (transferred to the central counterparty).'

</...>

```

**Diagram**

**Schema Component Representation**

```

<xsd:complexType name="TradeHeader">
  <xsd:sequence>
    <xsd:element name="partyTradeIdentifier" type=" PartyTradeIdentifier " maxOccurs="unbounded" />
    <xsd:element name="partyTradeInformation" type=" PartyTradeInformation "
      minOccurs="0" maxOccurs="unbounded" />
    <xsd:element name="tradeDate" type=" IdentifiedDate " />
    <!-- RPTWG 2009-11-12 begin-->
    <xsd:element name="clearedDate" type=" IdentifiedDate " minOccurs="0" />
    <!-- RPTWG 2009-11-12 end -->
  </xsd:sequence>
</xsd:complexType>

```

top

**Complex Type: TradId**

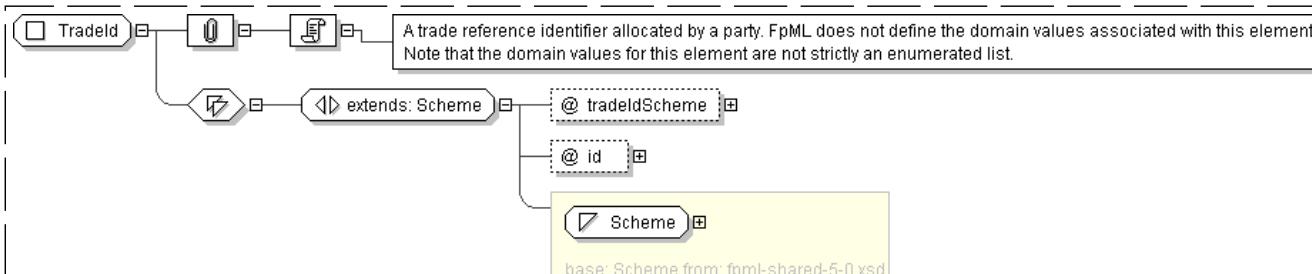
<b>Super-types:</b>	<a href="#">Scheme</a> < <b>TradId</b> (by extension)
<b>Sub-types:</b>	None
<b>Name</b>	<b>TradId</b>
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">Portfolio</a> , Complex Type <a href="#">TradId</a> , Complex Type <a href="#">VersionedTradId</a>
<b>Abstract</b>	no
<b>Documentation</b>	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.

**XML Instance Representation**

```

<...
|
```

```
| tradeIdScheme=" xsd:anyURI [1]"
| id=" xsd:ID [0..1]">
| Scheme
| </...>
```

**Diagram****Schema Component Representation**

```
<xsd:complexType name="TradeId">
  <xsd:simpleContent>
    <xsd:extension base=" Scheme ">
      <xsd:attribute name="tradeIdScheme" type=" xsd:anyURI " use="required"/>
      <xsd:attribute name="id" type=" xsd:ID "/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
```

top

**Complex Type: TradeIdentifier**

Super-types: None

Sub-types:

- [PartyTradeIdentifier](#) (by extension)

Name	TradeIdentifier
Used by (from the same schema document)	Complex Type <a href="#">Allocation</a> , Complex Type <a href="#">BestFitTrade</a> , Complex Type <a href="#">PartyTradeIdentifier</a> , Complex Type <a href="#">PartyTradeIdentifier</a>
Abstract	no
Documentation	A type defining a trade identifier issued by the indicated party.

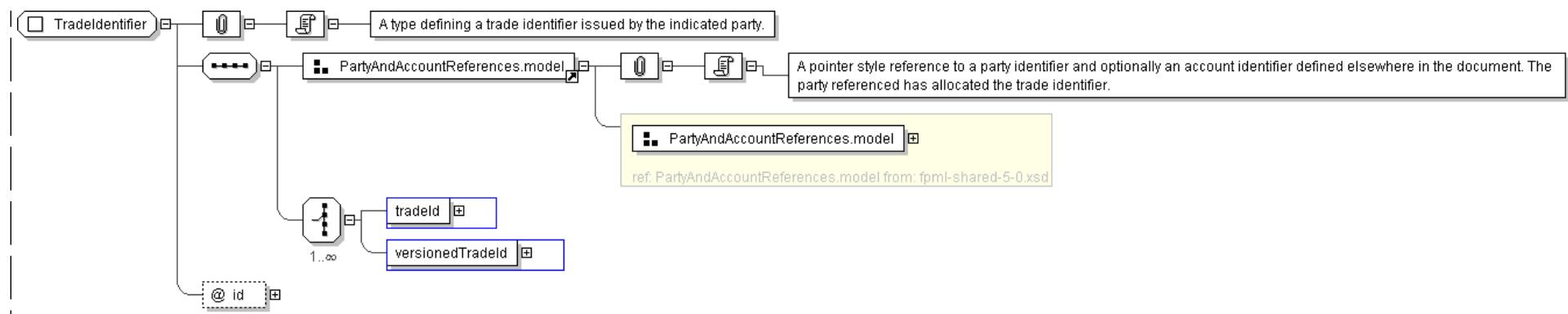
**XML Instance Representation**

```
<...
  id=" xsd:ID [0..1]">
  <partyReference> PartyReference </partyReference> [1]
  'Reference to a party.'

  <accountReference> AccountReference </accountReference> [0..1]
  'Reference to an account.'

  Start Choice [1..*]
    <tradeId> TradeId </tradeId> [1]
    <versionedTradeId> VersionedTradeId </versionedTradeId> [1]
  End Choice
</...>
```

**Diagram**

**Schema Component Representation**

```
<xsd:complexType name="TradeIdentifier">
  <xsd:sequence>
    <xsd:group ref=" PartyAndAccountReferences.model " />
    <xsd:choice maxOccurs="unbounded">
      <xsd:element name="tradelId" type=" TradeId " />
      <xsd:element name="versionedTradelId" type=" VersionedTradeId " />
    </xsd:choice>
  </xsd:sequence>
  <xsd:attribute name="id" type=" xsd:ID " />
</xsd:complexType>
```

top

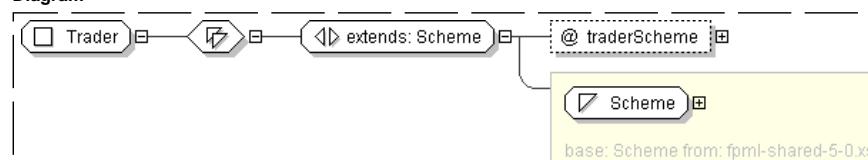
**Complex Type: Trader**

<b>Super-types:</b>	<a href="#">Scheme</a> < <b>Trader</b> (by extension)
<b>Sub-types:</b>	None

<b>Name</b>	Trader
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">PartyTradeInformation</a>
<b>Abstract</b>	no

**XML Instance Representation**

```
<...
  traderScheme=" xsd:anyURI [0..1]">
  Scheme
</...>
```

**Diagram****Schema Component Representation**

```
<xsd:complexType name="Trader">
  <xsd:simpleContent>
    <xsd:extension base=" Scheme ">
      <xsd:attribute name="traderScheme" type=" xsd:anyURI " use="optional " />
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
```

```

</xsd:extension>
</xsd:simpleContent>
</xsd:complexType>

```

## Complex Type: Unit

**Super-types:** [Scheme](#) < **Unit** (by extension)

**Sub-types:** None

**Name** **Unit**

**Used by (from the same schema document)** Complex Type [PartyTradeInformation](#)

**Abstract** no

**Documentation** A type used to record information about a unit, subdivision, desk, or other similar business entity.

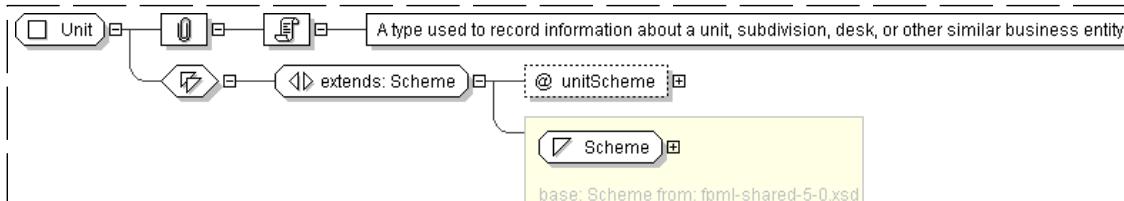
### XML Instance Representation

```

<...
  unitscheme=" xsd:anyURI [0..1]">
  Scheme
</...>

```

### Diagram



### Schema Component Representation

```

<xsd:complexType name="Unit">
  <xsd:simpleContent>
    <xsd:extension base=" Scheme ">
      <xsd:attribute name="unitScheme" type=" xsd:anyURI " use="optional"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>

```

## Complex Type: Validation

**Super-types:** [Scheme](#) < **Validation** (by extension)

**Sub-types:** None

**Name** **Validation**

**Used by (from the same schema document)** Model Group [Validation.model](#)

**Abstract** no

**Documentation** A reference identifying a rule within a validation scheme.

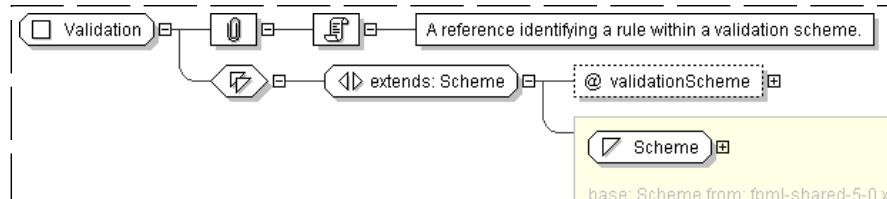
### XML Instance Representation

```

<...
  validationScheme=" xsd:anyURI [0..1]">

```

[Scheme](#)  
 </...>

**Diagram****Schema Component Representation**

```

<xsd:complexType name="Validation">
  <xsd:simpleContent>
    <xsd:extension base=" Scheme ">
      <xsd:attribute name="validationScheme" type=" xsd:anyURI ">
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
  
```

top

**Complex Type: VersionedContractId**

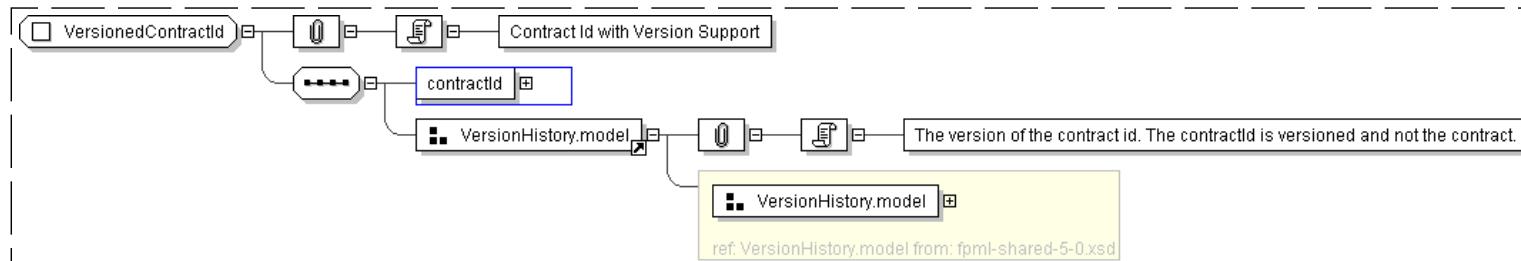
Super-types:	None
Sub-types:	None

Name	VersionedContractId
Used by (from the same schema document)	Complex Type <a href="#">ContractIdentifier</a>
Abstract	no
Documentation	Contract Id with Version Support

**XML Instance Representation**

```

<...>
<contractId> ContractId </contractId> [1]
<version> xsd:nonNegativeInteger </version> [1]
  'The version number'
<effectiveDate> IdentifiedDate </effectiveDate> [0..1]
  'Optionally it is possible to specify a version effective date when a versionId is supplied.'
</...>
  
```

**Diagram**

**Schema Component Representation**

```
<xsd:complexType name="VersionedContractId">
  <xsd:sequence>
    <xsd:element name="contractId" type=" ContractId " />
    <xsd:group ref=" VersionHistory.model " />
  </xsd:sequence>
</xsd:complexType>
```

[top](#)**Complex Type: VersionedTradeId**

Super-types:	None
Sub-types:	None

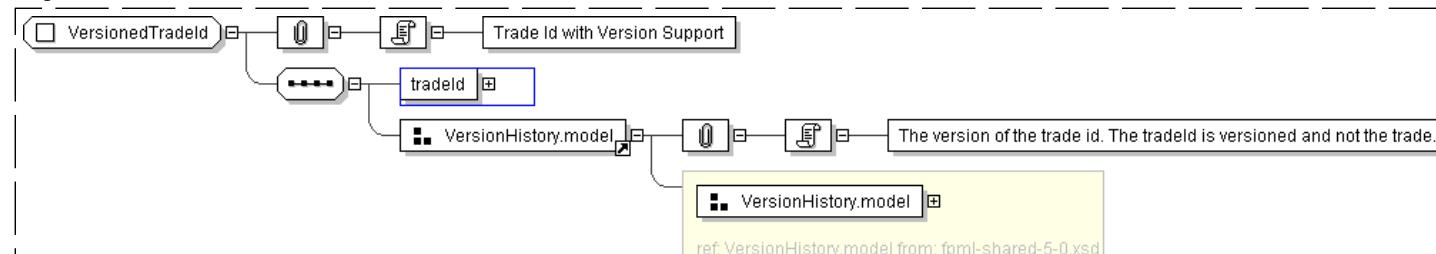
Name	VersionedTradeId
Used by (from the same schema document)	Complex Type <a href="#">TradeIdIdentifier</a>
Abstract	no
Documentation	Trade Id with Version Support

**XML Instance Representation**

```
<...>
  <tradeId> TradeId </tradeId> [1]
  <version> xsd:nonNegativeInteger </version> [1]
  'The version number'

  <effectiveDate> IdentifiedDate </effectiveDate> [0..1]
  'Optionally it is possible to specify a version effective date when a versionId is supplied.'

</...>
```

**Diagram****Schema Component Representation**

```
<xsd:complexType name="VersionedTradeId">
  <xsd:sequence>
    <xsd:element name="tradeId" type=" TradeId " />
    <xsd:group ref=" VersionHistory.model " />
  </xsd:sequence>
</xsd:complexType>
```

[top](#)**Model Group: AccountReferenceOrPartyReference.model**

Name	AccountReferenceOrPartyReference.model
------	--

**XML Instance Representation**

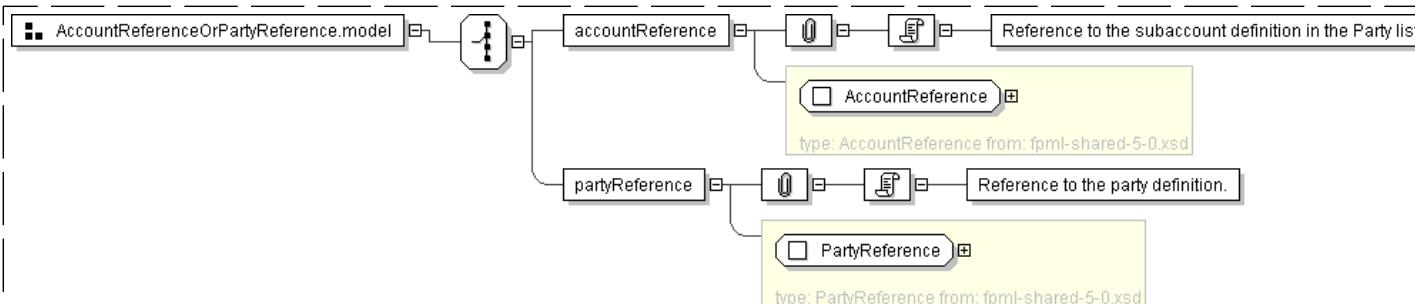
```

Start Choice [1]
<accountReference> AccountReference </accountReference> [1]
  'Reference to the subaccount definition in the Party list.'

<partyReference> PartyReference </partyReference> [1]
  'Reference to the party definition.'

End Choice

```

**Diagram****Schema Component Representation**

```

<xsd:group name="AccountReferenceOrPartyReference.model">
  <xsd:choice>
    <xsd:element name="accountReference" type=" AccountReference " />
    <xsd:element name="partyReference" type=" PartyReference " />
  </xsd:choice>
</xsd:group>

```

[top](#)**Model Group: AllocationContent.model**

Name	AllocationContent.model
Used by (from the same schema document)	Complex Type <a href="#">Allocation</a>

**XML Instance Representation**

```

<collateral> Collateral </collateral> [0..1]
  'The sum that must be posted upfront to collateralize against counterparty credit risk.'

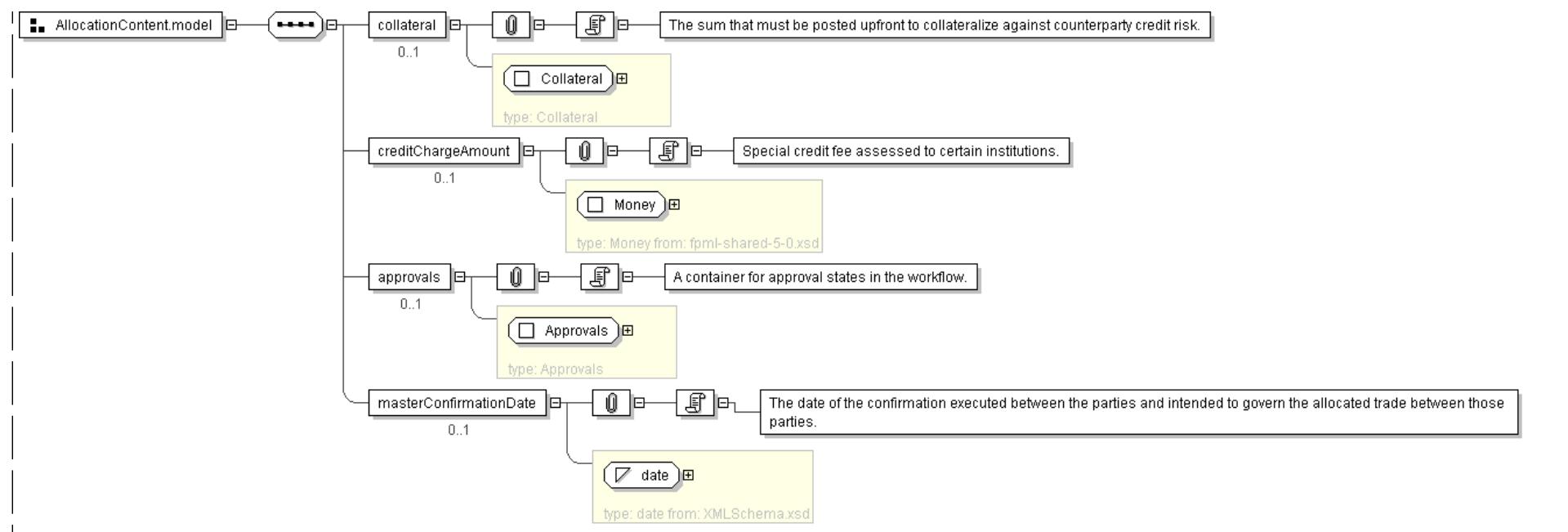
<creditChargeAmount> Money </creditChargeAmount> [0..1]
  'Special credit fee assessed to certain institutions.'

<approvals> Approvals </approvals> [0..1]
  'A container for approval states in the workflow.'

<masterConfirmationDate> xsd:date </masterConfirmationDate> [0..1]
  'The date of the confirmation executed between the parties and intended to govern the
  allocated trade between those parties.'

```

**Diagram**



#### Schema Component Representation

```

<xsd:group name="AllocationContent.model">
  <xsd:sequence>
    <xsd:element name="collateral" type="Collateral" minOccurs="0"/>
    <xsd:element name="creditChargeAmount" type="Money" minOccurs="0"/>
    <xsd:element name="approvals" type="Approvals" minOccurs="0"/>
    <xsd:element name="masterConfirmationDate" type="xsd:date" minOccurs="0"/>
  </xsd:sequence>
</xsd:group>
  
```

[top](#)

#### Model Group: AmendmentDetails.model

##### Name

AmendmentDetails.model

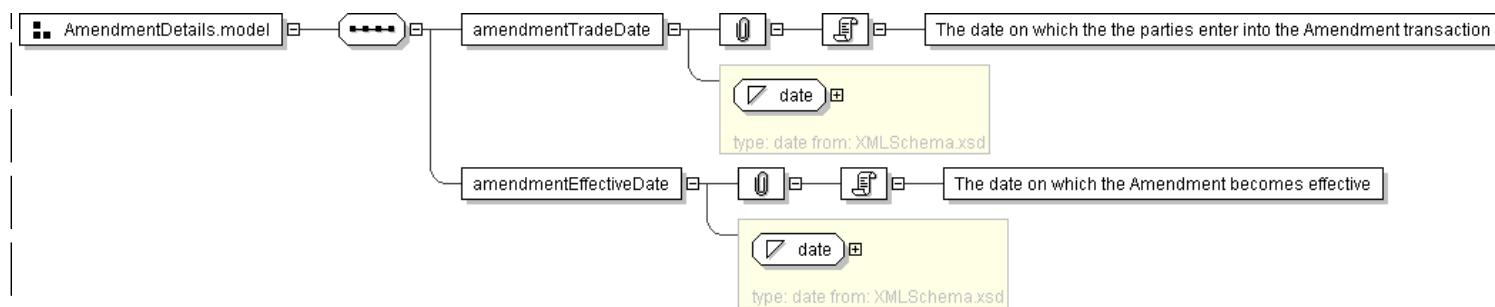
##### XML Instance Representation

```

<amendmentTradeDate> xsd:date </amendmentTradeDate> [1]
'The date on which the parties enter into the Amendment transaction'

<amendmentEffectiveDate> xsd:date </amendmentEffectiveDate> [1]
'The date on which the Amendment becomes effective'
  
```

##### Diagram

**Schema Component Representation**

```

<xsd:group name="AmendmentDetails.model">
  <xsd:sequence>
    <xsd:element name="amendmentTradeDate" type="xsd:date" />
    <xsd:element name="amendmentEffectiveDate" type="xsd:date" />
  </xsd:sequence>
</xsd:group>
  
```

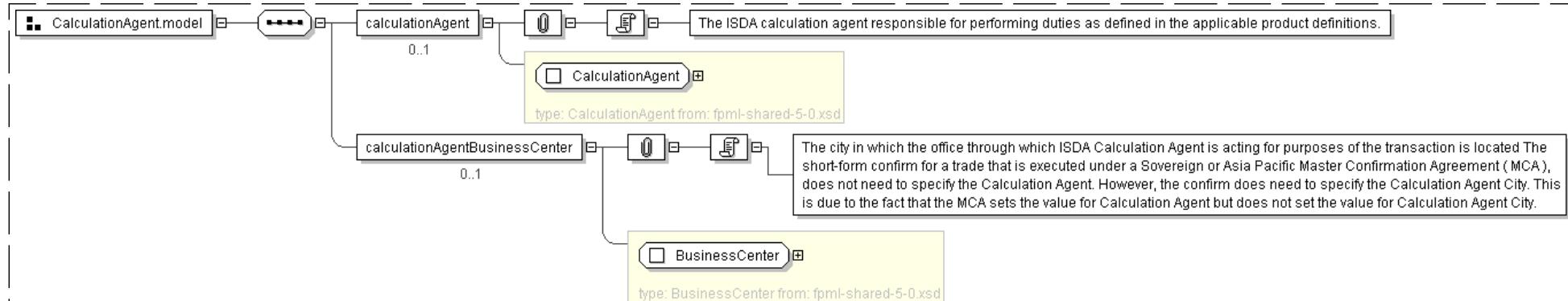
[top](#)**Model Group: CalculationAgent.model**

<b>Name</b>	CalculationAgent.model
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">Trade</a>

**XML Instance Representation**

```

<calculationAgent> CalculationAgent </calculationAgent> [0..1]
' The ISDA calculation agent responsible for performing duties as defined in the
applicable product definitions.'
<calculationAgentBusinessCenter> BusinessCenter </calculationAgentBusinessCenter> [0..1]
' The city in which the office through which ISDA Calculation Agent is acting for purposes
of the transaction is located. The short-form confirm for a trade that is executed under
a Sovereign or Asia Pacific Master Confirmation Agreement ( MCA ), does not need to specify
the Calculation Agent. However, the confirm does need to specify the Calculation Agent
City. This is due to the fact that the MCA sets the value for Calculation Agent but does
not set the value for Calculation Agent City.'
  
```

**Diagram****Schema Component Representation**

```

<xsd:group name="CalculationAgent.model">
  <xsd:sequence>
    <xsd:element name="calculationAgent" type=" CalculationAgent " minOccurs="0"/>
    <xsd:element name="calculationAgentBusinessCenter" type=" BusinessCenter " minOccurs="0"/>
  </xsd:sequence>
</xsd:group>

```

[top](#)

## Model Group: TradeOrTradeReference.model

Name	TradeOrTradeReference.model
Documentation	Choice between identification and representation of trade execution.

### XML Instance Representation

```

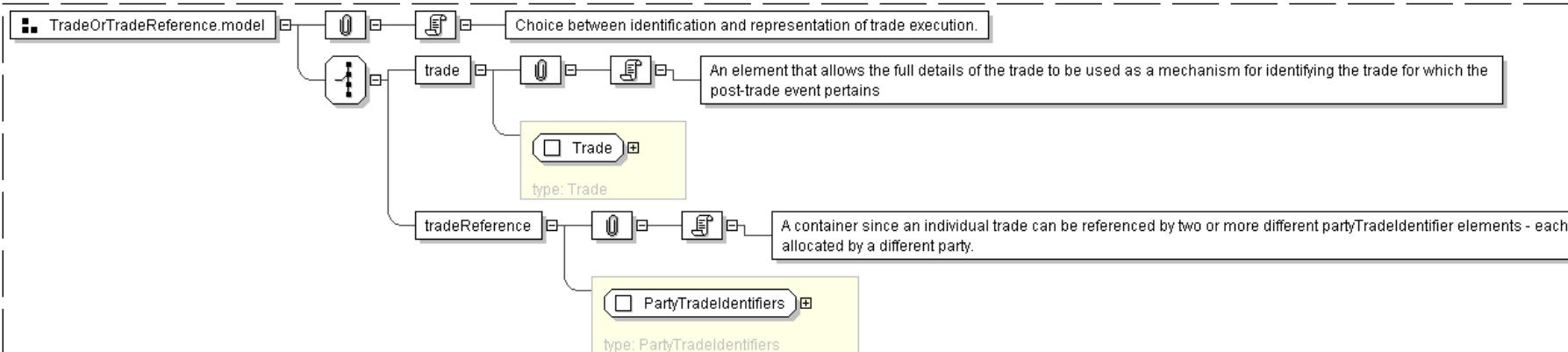
Start Choice [1]
<trade> Trade </trade> [1]
'An element that allows the full details of the trade to be used as a mechanism for identifying the trade for which the post-trade event pertains'

<tradeReference> PartyTradeIdentifiers </tradeReference> [1]
'A container since an individual trade can be referenced by two or more different partyTradeIdentifier elements - each allocated by a different party.'

```

End Choice

### Diagram



### Schema Component Representation

```

<xsd:group name="TradeOrTradeReference.model">
  <xsd:choice>
    <xsd:element name="trade" type=" Trade " />
    <xsd:element name="tradeReference" type=" PartyTradeIdentifiers " />
  </xsd:choice>
</xsd:group>

```

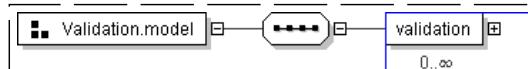
[top](#)

## Model Group: Validation.model

Name	Validation.model
------	------------------

**Used by (from the same schema document)**Complex Type [DataDocument](#)**XML Instance Representation**

```
<validation> Validation </validation> [0..*]
```

**Diagram****Schema Component Representation**

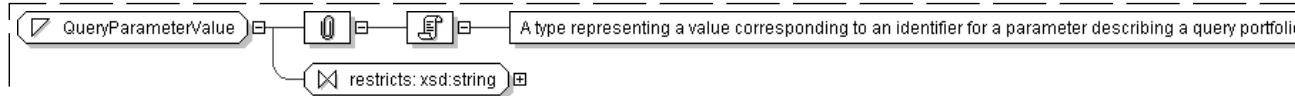
```

<xsd:group name="Validation.model">
  <xsd:sequence>
    <xsd:element name="validation" type="xsd:string" minOccurs="0" maxOccurs="unbounded" />
  </xsd:sequence>
</xsd:group>

```

[top](#)**Simple Type: QueryParameterValue**Super-types: [xsd:string](#) < **QueryParameterValue** (by restriction)

Sub-types: None

**Name** [QueryParameterValue](#)**Content** • Base XSD Type: string**Documentation** A type representing a value corresponding to an identifier for a parameter describing a query portfolio.**Diagram****Schema Component Representation**

```

<xsd:simpleType name="QueryParameterValue">
  <xsd:restriction base="xsd:string" />
</xsd:simpleType>

```

[top](#)**Legend****Complex Type:**

Schema Component Type

**AusAddress**

Schema Component Name

Super-types: [Address](#) < AusAddress (by extension)Sub-types: • [QLDAddress](#) (by restriction)

If this schema component is a type definition, its type hierarchy is shown in a gray-bordered box.

**Name****Abstract**

AusAddress

no

The table above displays the properties of this schema component.

### XML Instance Representation

```
<... country="Australia" >
<unitNo> string </unitNo> [0..1]
<houseNo> string </houseNo> [1]
<street> string </street> [1]
Start Choice [1]
<city> string </city> [1]
<town> string </town> [1]
End Choice
<state> AusStates </state> [1]
<postcode> string <><pattern = "[1-9][0-9]{3}"></postcode> [1]
</...>
```

The XML Instance Representation table above shows the schema component's content as an XML instance.

- The minimum and maximum occurrence of elements and attributes are provided in square brackets, e.g. [0..1].
- Model group information are shown in gray, e.g. Start Choice ... End Choice.
- For type derivations, the elements and attributes that have been added to or changed from the base type's content are shown in **bold**.
- If an element/attribute has a fixed value, the fixed value is shown in green, e.g. country="Australia".
- Otherwise, the type of the element/attribute is displayed.
  - If the element/attribute's type is in the schema, a link is provided to it.
  - For local simple type definitions, the constraints are displayed in angle brackets, e.g. <><pattern = "[1-9][0-9]{3}"></>.

### Schema Component Representation

```
<complexType name="AusAddress">
<complexContent>
<extension base=" Address ">
<sequence>
<element name="state" type=" AusStates " />
<element name="postcode">
<simpleType>
<restriction base=" string ">
<pattern value="[1-9][0-9]{3}">
</restriction>
</simpleType>
</element>
</sequence>
<attribute name="country" type=" string " fixed="Australia" />
</extension>
</complexContent>
</complexType>
```

The Schema Component Representation table above displays the underlying XML representation of the schema component. (Annotations are not shown.)

[top](#)

## Glossary

**Abstract** (Applies to complex type definitions and element declarations). An abstract element or complex type cannot be used to validate an element instance. If there is a reference to an abstract element, only element declarations that can substitute the abstract element can be used to validate the instance. For references to abstract type definitions, only derived types can be used.

**All Model Group** Child elements can be provided *in any order* in instances. See: <http://www.w3.org/TR/xmlschema-1/#element-all>.

**Choice Model Group** Only one from the list of child elements and model groups can be provided in instances. See: <http://www.w3.org/TR/xmlschema-1/#element-choice>.

**Collapse Whitespace Policy** Replace tab, line feed, and carriage return characters with space character (Unicode character 32). Then, collapse contiguous sequences of space characters into single space character, and remove leading and trailing space characters.

**Disallowed Substitutions** (Applies to element declarations). If *substitution* is specified, then *substitution group* members cannot be used in place

of the given element declaration to validate element instances. If **derivation methods**, e.g. extension, restriction, are specified, then the given element declaration will not validate element instances that have types derived from the element declaration's type using the specified derivation methods. Normally, element instances can override their declaration's type by specifying an `xsi:type` attribute.

**Key Constraint** Like [Uniqueness Constraint](#), but additionally requires that the specified value(s) must be provided. See: [http://www.w3.org/TR/xmlschema-1/#cIdentity-constraint\\_Definitions](http://www.w3.org/TR/xmlschema-1/#cIdentity-constraint_Definitions).

**Key Reference Constraint** Ensures that the specified value(s) must match value(s) from a [Key Constraint](#) or [Uniqueness Constraint](#). See: [http://www.w3.org/TR/xmlschema-1/#cIdentity-constraint\\_Definitions](http://www.w3.org/TR/xmlschema-1/#cIdentity-constraint_Definitions).

**Model Group** Groups together element content, specifying the order in which the element content can occur and the number of times the group of element content may be repeated. See: [http://www.w3.org/TR/xmlschema-1/#Model\\_Groups](http://www.w3.org/TR/xmlschema-1/#Model_Groups).

**Nillable** (Applies to element declarations). If an element declaration is nillable, instances can use the `xsi:nil` attribute. The `xsi:nil` attribute is the boolean attribute, `nil`, from the <http://www.w3.org/2001/XMLSchema-instance> namespace. If an element instance has an `xsi:nil` attribute set to true, it can be left empty, even though its element declaration may have required content.

**Notation** A notation is used to identify the format of a piece of data. Values of elements and attributes that are of type, NOTATION, must come from the names of declared notations. See: [http://www.w3.org/TR/xmlschema-1/#cNotation\\_Declarations](http://www.w3.org/TR/xmlschema-1/#cNotation_Declarations).

**Preserve Whitespace Policy** Preserve whitespaces exactly as they appear in instances.

**Prohibited Derivations** (Applies to type definitions). Derivation methods that cannot be used to create sub-types from a given type definition.

**Prohibited Substitutions** (Applies to complex type definitions). Prevents sub-types that have been derived using the specified derivation methods from validating element instances in place of the given type definition.

**Replace Whitespace Policy** Replace tab, line feed, and carriage return characters with space character (Unicode character 32).

**Sequence Model Group** Child elements and model groups must be provided *in the specified order* in instances. See: <http://www.w3.org/TR/xmlschema-1/#element-sequence>.

**Substitution Group** Elements that are *members* of a substitution group can be used wherever the `head` element of the substitution group is referenced.

**Substitution Group Exclusions** (Applies to element declarations). Prohibits element declarations from nominating themselves as being able to substitute a given element declaration, if they have types that are derived from the original element's type using the specified derivation methods.

**Target Namespace** The target namespace identifies the namespace that components in this schema belongs to. If no target namespace is provided, then the schema components do not belong to any namespace.

**Uniqueness Constraint** Ensures uniqueness of an element/attribute value, or a combination of values, within a specified scope. See: [http://www.w3.org/TR/xmlschema-1/#cIdentity-constraint\\_Definitions](http://www.w3.org/TR/xmlschema-1/#cIdentity-constraint_Definitions).

[top](#)

# XML Schema Documentation

---

## Table of Contents

- [Schema Document Properties](#)
- [Global Definitions](#)
  - Simple Type: [AveragingInOutEnum](#)
  - Simple Type: [AveragingMethodEnum](#)
  - Simple Type: [BreakageCostEnum](#)
  - Simple Type: [BullionTypeEnum](#)
  - Simple Type: [BusinessDayConventionEnum](#)
  - Simple Type: [CalculationAgentPartyEnum](#)
  - Simple Type: [CashPhysicalEnum](#)
  - Simple Type: [CommissionDenominationEnum](#)
  - Simple Type: [CommodityBullionSettlementDisruptionEnum](#)
  - Simple Type: [CommodityDayTypeEnum](#)
  - Simple Type: [CompoundingMethodEnum](#)
  - Simple Type: [DayOfWeekEnum](#)
  - Simple Type: [DayTypeEnum](#)
  - Simple Type: [DeliveryDatesEnum](#)
  - Simple Type: [DeliveryTypeEnum](#)
  - Simple Type: [DifferenceSeverityEnum](#)
  - Simple Type: [DifferenceTypeEnum](#)
  - Simple Type: [DiscountingTypeEnum](#)
  - Simple Type: [DisruptionFallbacksEnum](#)
  - Simple Type: [DividendAmountTypeEnum](#)
  - Simple Type: [DividendCompositionEnum](#)
  - Simple Type: [DividendDateReferenceEnum](#)
  - Simple Type: [DividendEntitlementEnum](#)
  - Simple Type: [DividendPeriodEnum](#)
  - Simple Type: [ElectricityProductTypeEnum](#)
  - Simple Type: [EquityOptionTypeEnum](#)
  - Simple Type: [ExerciseStyleEnum](#)
  - Simple Type: [FPVFinalPriceElectionFallbackEnum](#)
  - Simple Type: [FeeElectionEnum](#)
  - Simple Type: [FlatRateEnum](#)
  - Simple Type: [FraDiscountingEnum](#)
  - Simple Type: [FrequencyTypeEnum](#)
  - Simple Type: [FxBarrierTypeEnum](#)
  - Simple Type: [GasProductTypeEnum](#)
  - Simple Type: [IndexEventConsequenceEnum](#)
  - Simple Type: [InterestCalculationMethodEnum](#)
  - Simple Type: [InterestShortfallCapEnum](#)
  - Simple Type: [InterpolationPeriodEnum](#)
  - Simple Type: [LengthUnitEnum](#)
  - Simple Type: [MarketDisruptionEventsEnum](#)
  - Simple Type: [MethodOfAdjustmentEnum](#)
  - Simple Type: [NationalisationOrInsolvencyOrDelistingEventEnum](#)
  - Simple Type: [NegativeInterestRateTreatmentEnum](#)
  - Simple Type: [NonCashDividendTreatmentEnum](#)
  - Simple Type: [NotionalAdjustmentEnum](#)
  - Simple Type: [ObligationCategoryEnum](#)
  - Simple Type: [OptionTypeEnum](#)
  - Simple Type: [PayRelativeToEnum](#)
  - Simple Type: [PayerReceiverEnum](#)
  - Simple Type: [PayoutEnum](#)
  - Simple Type: [PeriodEnum](#)
  - Simple Type: [PeriodExtendedEnum](#)
  - Simple Type: [PremiumQuoteBasisEnum](#)
  - Simple Type: [PremiumTypeEnum](#)
  - Simple Type: [PriceExpressionEnum](#)
  - Simple Type: [PutCallEnum](#)
  - Simple Type: [QuotationRateTypeEnum](#)
  - Simple Type: [QuotationSideEnum](#)
  - Simple Type: [QuotationStyleEnum](#)
  - Simple Type: [QuoteBasisEnum](#)
  - Simple Type: [RateTreatmentEnum](#)
  - Simple Type: [RealisedVarianceMethodEnum](#)
  - Simple Type: [ResetRelativeToEnum](#)
  - Simple Type: [ReturnTypeEnum](#)
  - Simple Type: [RollConventionEnum](#)
  - Simple Type: [RoundingDirectionEnum](#)
  - Simple Type: [SettlementPeriodDurationEnum](#)
  - Simple Type: [SettlementTypeEnum](#)
  - Simple Type: [ShareExtraordinaryEventEnum](#)
  - Simple Type: [SideRateBasisEnum](#)
  - Simple Type: [SpecifiedPriceEnum](#)
  - Simple Type: [StandardSettlementStyleEnum](#)
  - Simple Type: [StepRelativeToEnum](#)
  - Simple Type: [StrikeQuoteBasisEnum](#)
  - Simple Type: [StubPeriodTypeEnum](#)

- Simple Type: [TimeTypeEnum](#)
- Simple Type: [TouchConditionEnum](#)
- Simple Type: [TriggerConditionEnum](#)
- Simple Type: [TriggerTimeTypeEnum](#)
- Simple Type: [TriggerTypeEnum](#)
- Simple Type: [ValuationMethodEnum](#)
- Simple Type: [WeeklyRollConventionEnum](#)

- [Legend](#)
- [Glossary](#)

## Schema Document Properties

Target Namespace	<a href="http://www.fpml.org/FpML-5/confirmation">http://www.fpml.org/FpML-5/confirmation</a>
Version	\$Revision: 2586 \$
Element and Attribute Namespaces	<ul style="list-style-type: none"> <li>Global element and attribute declarations belong to this schema's target namespace.</li> <li>By default, local element declarations belong to this schema's target namespace.</li> <li>By default, local attribute declarations have no namespace.</li> </ul>

## Declared Namespaces

Prefix	Namespace
Default namespace	<a href="http://www.fpml.org/FpML-5/confirmation">http://www.fpml.org/FpML-5/confirmation</a>
xml	<a href="http://www.w3.org/XML/1998/namespace">http://www.w3.org/XML/1998/namespace</a>
fpmi-annotation	<a href="http://www.fpml.org/annotation">http://www.fpml.org/annotation</a>
xsd	<a href="http://www.w3.org/2001/XMLSchema">http://www.w3.org/2001/XMLSchema</a>
ecore	<a href="http://www.eclipse.org/emf/2002/Ecore">http://www.eclipse.org/emf/2002/Ecore</a>

## Schema Component Representation

```
<xsd:schema targetNamespace="http://www.fpml.org/FpML-5/confirmation"
  documentRoot="FpML" nsPrefix="conf" package="org.fpml.confirmation" version="$Revision: 2586 "
  $" elementFormDefault="qualified" attributeFormDefault="unqualified">
  ...
</xsd:schema>
```

## Global Definitions

### Simple Type: AveragingInOutEnum

Super-types:	<a href="#">xsd:token</a> < <b>AveragingInOutEnum</b> (by restriction)
Sub-types:	None
Name	<b>AveragingInOutEnum</b>
Content	<ul style="list-style-type: none"> <li>Base XSD Type: token</li> <li>value comes from list: ('In' 'Out' 'Both')</li> </ul>
Documentation	The type of averaging used in an Asian option.
<b>Diagram</b>	
<pre> classDiagram     class AveragingInOutEnum {         &lt;&lt;The type of averaging used in an Asian option.&gt;&gt;     }     class In {         &lt;&lt;The average price is used to derive the strike price. Also known as "Asian strike" style option.&gt;&gt;     }     class Out {         &lt;&lt;The average price is used to derive the expiration price. Also known as "Asian price" style option.&gt;&gt;     }     class Both {         &lt;&lt;The average price is used to derive both the strike and the expiration price.&gt;&gt;     }     class token {         &lt;&lt;base: token from: XMLSchema.xsd&gt;&gt;     }     AveragingInOutEnum &lt; -- In     AveragingInOutEnum &lt; -- Out     AveragingInOutEnum &lt; -- Both   </pre>	

## Schema Component Representation

```
<xsd:simpleType name="AveragingInOutEnum">
  <xsd:restriction base="xsd:token">
    <xsd:enumeration value="In"/>
    <xsd:enumeration value="Out"/>
    <xsd:enumeration value="Both"/>
  </xsd:restriction>
</xsd:simpleType>
```

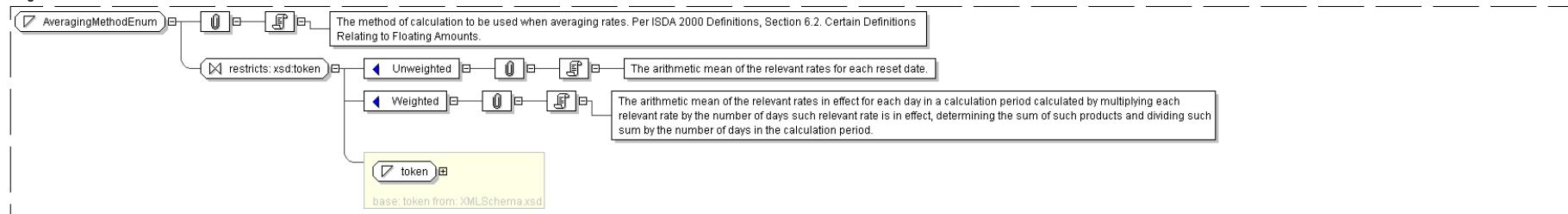
### Simple Type: AveragingMethodEnum

Super-types:	<a href="#">xsd:token</a> < <b>AveragingMethodEnum</b> (by restriction)
file:///C/Irina-Local/Subversion/branches/FpML-5-0/pdf/confirmation/fpmi-enum-5-0.xsd.html (2 of 46) [7/7/2010 11:05:54 AM]	

Sub-types: None

Name	AveragingMethodEnum
Content	<ul style="list-style-type: none"> <li>• Base XSD Type: token</li> <li>• value comes from list: ('Unweighted' 'Weighted')</li> </ul>
Documentation	The method of calculation to be used when averaging rates. Per ISDA 2000 Definitions, Section 6.2. Certain Definitions Relating to Floating Amounts.

## Diagram



## Schema Component Representation

```

<xsd:simpleType name="AveragingMethodEnum">
  <xsd:restriction base="xsd:token">
    <xsd:enumeration value="Unweighted"/>
    <xsd:enumeration value="Weighted"/>
  </xsd:restriction>
</xsd:simpleType>
  
```

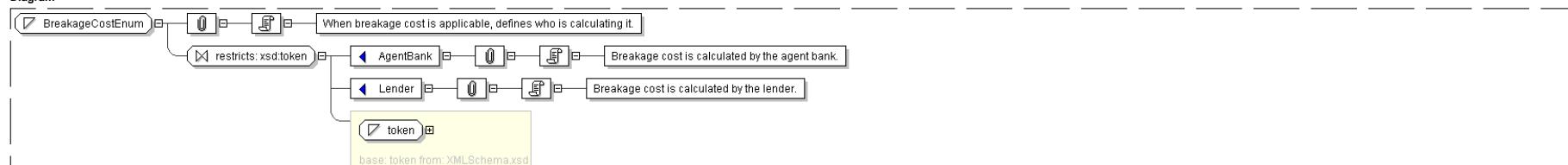
top

## Simple Type: BreakageCostEnum

Super-types: xsd:token < BreakageCostEnum (by restriction)  
Sub-types: None

Name	BreakageCostEnum
Content	<ul style="list-style-type: none"> <li>• Base XSD Type: token</li> <li>• value comes from list: ('AgentBank' 'Lender')</li> </ul>
Documentation	When breakage cost is applicable, defines who is calculating it.

## Diagram



## Schema Component Representation

```

<xsd:simpleType name="BreakageCostEnum">
  <xsd:restriction base="xsd:token">
    <xsd:enumeration value="AgentBank"/>
    <xsd:enumeration value="Lender"/>
  </xsd:restriction>
</xsd:simpleType>
  
```

top

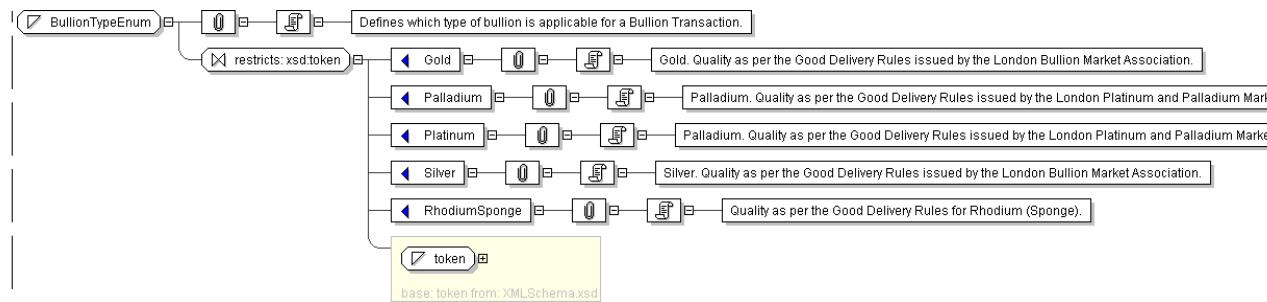
## Simple Type: BullionTypeEnum

Super-types: xsd:token < BullionTypeEnum (by restriction)  
Sub-types: None

Name	BullionTypeEnum
Content	<ul style="list-style-type: none"> <li>• Base XSD Type: token</li> <li>• value comes from list: ('Gold' 'Palladium' 'Platinum' 'Silver' 'RhodiumSponge')</li> </ul>
Documentation	Defines which type of bullion is applicable for a Bullion Transaction.

## Diagram





## Schema Component Representation

```

<xsd:simpleType name="BullionTypeEnum">
  <xsd:restriction base=" xsd:token ">
    <xsd:enumeration value="Gold"/>
    <xsd:enumeration value="Palladium"/>
    <xsd:enumeration value="Platinum"/>
    <xsd:enumeration value="Silver"/>
    <xsd:enumeration value="RhodiumSponge"/>
  </xsd:restriction>
</xsd:simpleType>

```

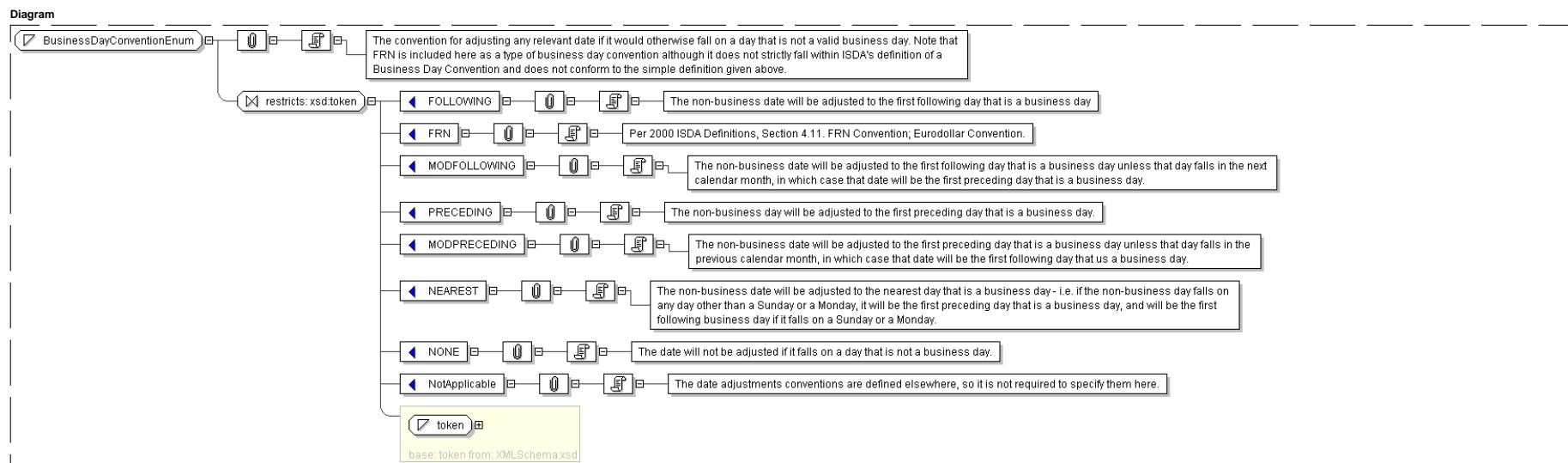
top

## Simple Type: BusinessDayConventionEnum

Super-types:	xsd:token < BusinessDayConventionEnum (by restriction)
Sub-types:	None

Name	BusinessDayConventionEnum
Content	<ul style="list-style-type: none"> <li>Base XSD Type: token</li> <li>value comes from list: ('FOLLOWING' 'FRN' 'MODFOLLOWING' 'PRECEDING' 'MODPRECEDING' 'NEAREST' 'NONE' 'NotApplicable')</li> </ul>
Documentation	

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.



## Schema Component Representation

```

<xsd:simpleType name="BusinessDayConventionEnum">
  <xsd:restriction base=" xsd:token ">
    <xsd:enumeration value="FOLLOWING"/>
    <xsd:enumeration value="FRN"/>
    <xsd:enumeration value="MODFOLLOWING"/>
    <xsd:enumeration value="PRECEDING"/>
    <xsd:enumeration value="MODPRECEDING"/>
    <xsd:enumeration value="NEAREST"/>
  </xsd:restriction>
</xsd:simpleType>

```

```

<xsd:enumeration value="NONE"/>
<xsd:enumeration value="NotApplicable"/>
</xsd:restriction>
</xsd:simpleType>

```

**Simple Type: CalculationAgentPartyEnum**

**Super-types:** xsd:token < **CalculationAgentPartyEnum** (by restriction)

**Sub-types:** None

<b>Name</b>	CalculationAgentPartyEnum
<b>Content</b>	<ul style="list-style-type: none"> <li>Base XSD Type: token</li> <li>value comes from list: ('ExercisingParty' 'NonExercisingParty' 'AsSpecifiedInMasterAgreement' 'AsSpecifiedInStandardTermsSupplement')</li> </ul>
<b>Documentation</b>	The specification of how a calculation agent will be determined.

**Diagram**

```

classDiagram
    class CalculationAgentPartyEnum {
        <<xsd:token>>
    }
    CalculationAgentPartyEnum --> xsd:token : restricts
    xsd:token --> ExercisingParty : ExercisingParty
    xsd:token --> NonExercisingParty : NonExercisingParty
    xsd:token --> AsSpecifiedInMasterAgreement : AsSpecifiedInMasterAgreement
    xsd:token --> AsSpecifiedInStandardTermsSupplement : AsSpecifiedInStandardTermsSupplement

```

base: token from: XMLSchema.xsd

**Schema Component Representation**

```

<xsd:simpleType name="CalculationAgentPartyEnum">
  <xsd:restriction base=" xsd:token ">
    <xsd:enumeration value="ExercisingParty"/>
    <xsd:enumeration value="NonExercisingParty"/>
    <xsd:enumeration value="AsSpecifiedInMasterAgreement"/>
    <xsd:enumeration value="AsSpecifiedInStandardTermsSupplement"/>
  </xsd:restriction>
</xsd:simpleType>

```

**Simple Type: CashPhysicalEnum**

**Super-types:** xsd:token < **CashPhysicalEnum** (by restriction)

**Sub-types:** None

<b>Name</b>	CashPhysicalEnum
<b>Content</b>	<ul style="list-style-type: none"> <li>Base XSD Type: token</li> <li>value comes from list: ('Cash' 'Physical')</li> </ul>
<b>Documentation</b>	Shows how the transaction is to be settled when it is exercised.

**Diagram**

```

classDiagram
    class CashPhysicalEnum {
        <<xsd:token>>
    }
    CashPhysicalEnum --> xsd:token : restricts
    xsd:token --> Cash : Cash
    xsd:token --> Physical : Physical

```

base: token from: XMLSchema.xsd

**Schema Component Representation**

```

<xsd:simpleType name="CashPhysicalEnum">
  <xsd:restriction base=" xsd:token ">
    <xsd:enumeration value="Cash"/>
    <xsd:enumeration value="Physical"/>
  </xsd:restriction>
</xsd:simpleType>

```

**Simple Type: CommissionDenominationEnum**

**Super-types:** xsd:token < **CommissionDenominationEnum** (by restriction)

**Sub-types:** None

<b>Name</b>	CommissionDenominationEnum
<b>Content</b>	<ul style="list-style-type: none"> <li>• Base XSD Type: token</li> <li>• value comes from list: {BPS} {Percentage} {CentsPerShare} {FixedAmount}</li> </ul>
<b>Documentation</b>	The unit in which a commission is denominated.

**Diagram**

```

classDiagram
    class CommissionDenominationEnum {
        <<CommissionDenominationEnum>>
        <<xsd:token>>
    }
    class BPS {
        <<BPS>>
        <<xsd:token>>
    }
    class Percentage {
        <<Percentage>>
        <<xsd:token>>
    }
    class CentsPerShare {
        <<CentsPerShare>>
        <<xsd:token>>
    }
    class FixedAmount {
        <<FixedAmount>>
        <<xsd:token>>
    }

    CommissionDenominationEnum <|-- BPS
    CommissionDenominationEnum <|-- Percentage
    CommissionDenominationEnum <|-- CentsPerShare
    CommissionDenominationEnum <|-- FixedAmount

    BPS --> "The commission is expressed in basis points, in reference to the price referenced in the document."
    Percentage --> "The commission is expressed as a percentage of the gross price referenced in the document."
    CentsPerShare --> "The commission is expressed in cents per share."
    FixedAmount --> "The commission is expressed as a absolute amount."
  
```

base: token from: XMLSchema.xsd

**Schema Component Representation**

```

<xsd:simpleType name="CommissionDenominationEnum">
  <xsd:restriction base="#xsd:token">
    <xsd:enumeration value="BPS"/>
    <xsd:enumeration value="Percentage"/>
    <xsd:enumeration value="CentsPerShare"/>
    <xsd:enumeration value="FixedAmount"/>
  </xsd:restriction>
</xsd:simpleType>
  
```

**Simple Type: CommodityBullionSettlementDisruptionEnum**

**Super-types:** xsd:token < **CommodityBullionSettlementDisruptionEnum** (by restriction)

**Sub-types:** None

<b>Name</b>	CommodityBullionSettlementDisruptionEnum
<b>Content</b>	<ul style="list-style-type: none"> <li>• Base XSD Type: token</li> <li>• value comes from list: {Negotiation} {Cancellation and Payment}</li> </ul>
<b>Documentation</b>	The consequences of Bullion Settlement Disruption Events.

**Diagram**

```

classDiagram
    class CommodityBullionSettlementDisruptionEnum {
        <<CommodityBullionSettlementDisruptionEnum>>
        <<xsd:token>>
    }
    class Negotiation {
        <<Negotiation>>
        <<xsd:token>>
    }
    class CancellationAndPayment {
        <<Cancellation and Payment>>
        <<xsd:token>>
    }

    CommodityBullionSettlementDisruptionEnum <|-- Negotiation
    CommodityBullionSettlementDisruptionEnum <|-- CancellationAndPayment

    Negotiation --> "Negotiation will apply in the event of Bullion Settlement Disruption as per Section 10.5(d) of the 2005 Commodity Definitions."
    CancellationAndPayment --> "Cancellation and Payment will apply in the event of Bullion Settlement Disruption as per Section 10.5(d) of the 2005 Commodity Definitions."
  
```

base: token from: XMLSchema.xsd

**Schema Component Representation**

```

<xsd:simpleType name="CommodityBullionSettlementDisruptionEnum">
  <xsd:restriction base="#xsd:token">
    <xsd:enumeration value="Negotiation"/>
    <xsd:enumeration value="Cancellation and Payment"/>
  </xsd:restriction>
</xsd:simpleType>
  
```

**Simple Type: CommodityDayTypeEnum**

Super-types:

None  
None

Name

CommodityDayTypeEnum

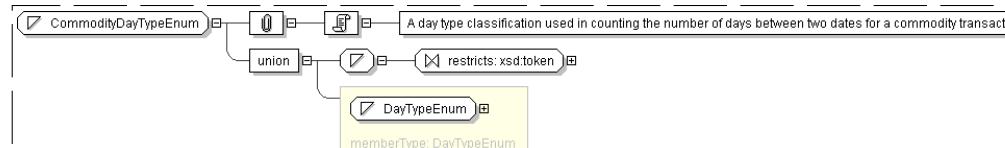
Content

- Union of following types:
  - [DayTypeEnum](#)
  - Locally defined type:
    - Base XSD Type: token
    - value comes from list: ('GasFlow')

Documentation

A day type classification used in counting the number of days between two dates for a commodity transaction.

Diagram



Schema Component Representation

```

<xsd:simpleType name="CommodityDayTypeEnum">
  <xsd:union memberTypes="# DayTypeEnum "#>
    <xsd:simpleType>
      <xsd:restriction base=" xsd:token ">
        <xsd:enumeration value="GasFlow"/>
      </xsd:restriction>
    </xsd:simpleType>
  </xsd:union>
</xsd:simpleType>
  
```

top

**Simple Type: CompoundingMethodEnum**

Super-types:

xsd:token &lt; CompoundingMethodEnum (by restriction)

Sub-types:

None

Name

CompoundingMethodEnum

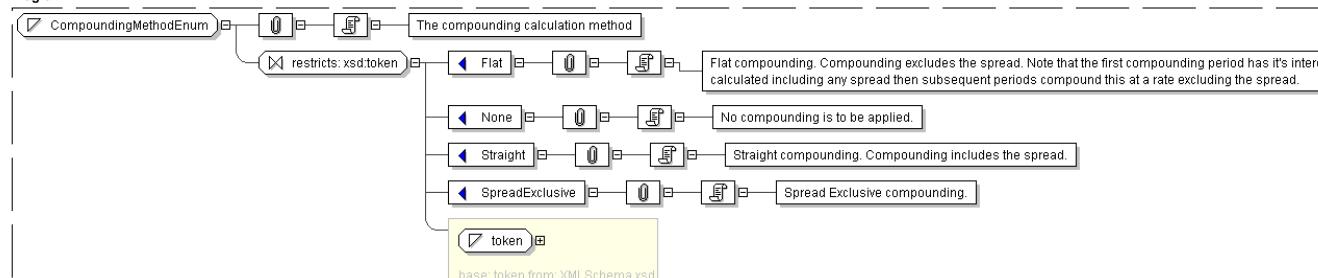
Content

- Base XSD Type: token
- value comes from list: ('Flat'|'None'|'Straight'|'SpreadExclusive')

Documentation

The compounding calculation method

Diagram



Schema Component Representation

```

<xsd:simpleType name="CompoundingMethodEnum">
  <xsd:restriction base=" xsd:token ">
    <xsd:enumeration value="Flat"/>
    <xsd:enumeration value="None"/>
    <xsd:enumeration value="Straight"/>
    <xsd:enumeration value="SpreadExclusive"/>
  </xsd:restriction>
</xsd:simpleType>
  
```

top

**Simple Type: DayOfWeekEnum**

Super-types:

xsd:token &lt; DayOfWeekEnum (by restriction)

Sub-types:

None

Name

DayOfWeekEnum

<b>Content</b>	<ul style="list-style-type: none"> <li>• Base XSD Type: token</li> <li>• <b>value</b> comes from list: {MON TUE WED THU FRI SAT SUN}</li> </ul>
<b>Documentation</b>	A day of the seven-day week.

**Diagram****Schema Component Representation**

```

<xsd:simpleType name="DayOfEnum">
  <xsd:restriction base="xsd:token">
    <xsd:enumeration value="MON"/>
    <xsd:enumeration value="TUE"/>
    <xsd:enumeration value="WED"/>
    <xsd:enumeration value="THU"/>
    <xsd:enumeration value="FRI"/>
    <xsd:enumeration value="SAT"/>
    <xsd:enumeration value="SUN"/>
  </xsd:restriction>
</xsd:simpleType>
  
```

top

**Simple Type: DayTypeEnum**

<b>Super-types:</b>	<code>xsd:token</code> < <code>DayTypeEnum</code> (by restriction)
<b>Sub-types:</b>	None

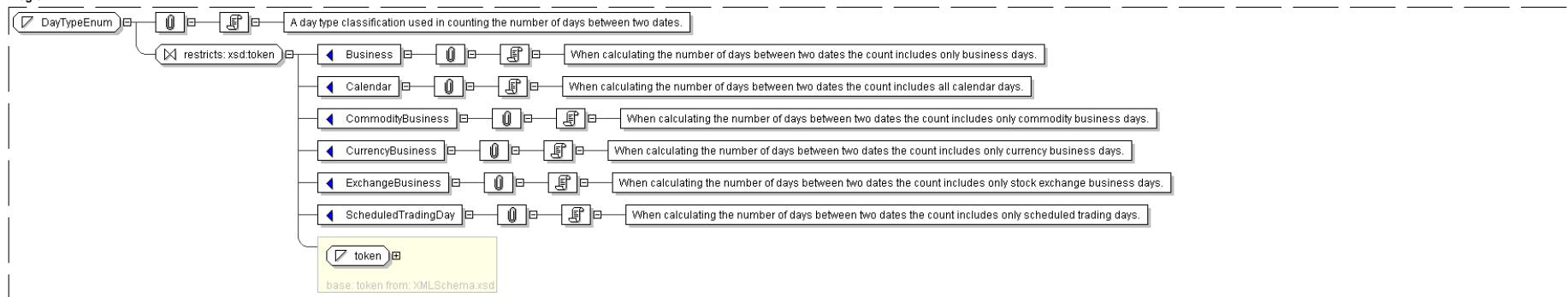
**Name****Content**

DayTypeEnum

- Base XSD Type: token
- **value** comes from list: {Business|Calendar|CommodityBusiness|CurrencyBusiness|ExchangeBusiness|ScheduledTradingDay}

**Documentation**

A day type classification used in counting the number of days between two dates.

**Diagram****Schema Component Representation**

```

<xsd:simpleType name="DayTypeEnum">
  <xsd:restriction base="xsd:token">
    <xsd:enumeration value="Business"/>
    <xsd:enumeration value="Calendar"/>
    <xsd:enumeration value="CommodityBusiness"/>
    <xsd:enumeration value="CurrencyBusiness"/>
    <xsd:enumeration value="ExchangeBusiness"/>
  </xsd:restriction>
</xsd:simpleType>
  
```

```

<xsd:enumeration value="ScheduledTradingDay" />
</xsd:restriction>
</xsd:simpleType>

```

**Simple Type: DeliveryDatesEnum**

**Super-types:** [xsd:token](#) < **DeliveryDatesEnum** (by restriction)

**Sub-types:** None

<b>Name</b>	DeliveryDatesEnum
<b>Content</b>	<ul style="list-style-type: none"> <li>• Base XSD Type: token</li> <li>• <b>value</b> comes from list: ('CalculationPeriod' 'FirstNearby' 'SecondNearby' 'ThirdNearby' 'SixthNearby' 'TwelfthNearby' 'Spot')</li> </ul>
<b>Documentation</b>	In respect of a Transaction and a Commodity Reference Price, the relevant date or month for delivery of the underlying Commodity.

**Diagram**

```

classDiagram
    class DeliveryDatesEnum {
        <<In respect of a Transaction and a Commodity Reference Price, the relevant date or month for delivery of the underlying Commodity.>>
    }
    class CalculationPeriod
    class FirstNearby
    class SecondNearby
    class ThirdNearby
    class SixthNearby
    class TwelfthNearby
    class Spot

    DeliveryDatesEnum <|-- xsd:token
    DeliveryDatesEnum <|-- CalculationPeriod
    DeliveryDatesEnum <|-- FirstNearby
    DeliveryDatesEnum <|-- SecondNearby
    DeliveryDatesEnum <|-- ThirdNearby
    DeliveryDatesEnum <|-- SixthNearby
    DeliveryDatesEnum <|-- TwelfthNearby
    DeliveryDatesEnum <|-- Spot

    CalculationPeriod <|-- FirstNearby
    CalculationPeriod <|-- SecondNearby
    CalculationPeriod <|-- ThirdNearby
    CalculationPeriod <|-- SixthNearby
    CalculationPeriod <|-- TwelfthNearby
    CalculationPeriod <|-- Spot

```

The diagram illustrates the inheritance of the `DeliveryDatesEnum` type from `xsd:token`. It also shows specific restrictions for various delivery periods: `CalculationPeriod`, `FirstNearby`, `SecondNearby`, `ThirdNearby`, `SixthNearby`, `TwelfthNearby`, and `Spot`. Each restriction is accompanied by a detailed description of its meaning.

**Schema Component Representation**

```

<xsd:simpleType name="DeliveryDatesEnum">
  <xsd:restriction base="xsd:token">
    <xsd:enumeration value="CalculationPeriod"/>
    <xsd:enumeration value="FirstNearby"/>
    <xsd:enumeration value="SecondNearby"/>
    <xsd:enumeration value="ThirdNearby"/>
    <xsd:enumeration value="SixthNearby"/>
    <xsd:enumeration value="TwelfthNearby"/>
    <xsd:enumeration value="Spot"/>
  </xsd:restriction>
</xsd:simpleType>

```

**Simple Type: DeliveryTypeEnum**

**Super-types:** [xsd:token](#) < **DeliveryTypeEnum** (by restriction)

**Sub-types:** None

<b>Name</b>	DeliveryTypeEnum
<b>Content</b>	<ul style="list-style-type: none"> <li>• Base XSD Type: token</li> <li>• <b>value</b> comes from list: ('Firm' 'Interruptible')</li> </ul>

**Diagram**

```

classDiagram
    class DeliveryTypeEnum {
        <<base: token from: XMLSchema.xsd>>
    }
    class Firm
    class Interruptible

    DeliveryTypeEnum <|-- xsd:token
    DeliveryTypeEnum <|-- Firm
    DeliveryTypeEnum <|-- Interruptible

```

The diagram illustrates the inheritance of the `DeliveryTypeEnum` type from `xsd:token`. It also shows specific restrictions for `Firm` and `Interruptible`. Each restriction is accompanied by a detailed description of its meaning.

**Schema Component Representation**

file:///C/Irina-Local/Subversion/branches/FpML-5-0/pdf/confirmation/fpmi-enum-5-0.xsd.html (9 of 46) [7/7/2010 11:05:54 AM]

```
<xsd:simpleType name="DeliveryTypeEnum">
  <xsd:restriction base=" xsd:token ">
    <xsd:enumeration value="Firm"/>
    <xsd:enumeration value="Interruptible"/>
  </xsd:restriction>
</xsd:simpleType>
```

[top](#)

## Simple Type: DifferenceSeverityEnum

Super-types: [xsd:token](#) < **DifferenceSeverityEnum** (by restriction)  
 Sub-types: None

Name	DifferenceSeverityEnum
Content	<ul style="list-style-type: none"> <li>• Base XSD Type: token</li> <li>• value comes from list: ('Warning' 'Error')</li> </ul>
Documentation	The ISDA defined value indicating the severity of a difference.

### Diagram



### Schema Component Representation

```
<xsd:simpleType name="DifferenceSeverityEnum">
  <xsd:restriction base=" xsd:token ">
    <xsd:enumeration value="Warning"/>
    <xsd:enumeration value="Error"/>
  </xsd:restriction>
</xsd:simpleType>
```

[top](#)

## Simple Type: DifferenceTypeEnum

Super-types: [xsd:token](#) < **DifferenceTypeEnum** (by restriction)  
 Sub-types: None

Name	DifferenceTypeEnum
Content	<ul style="list-style-type: none"> <li>• Base XSD Type: token</li> <li>• value comes from list: ('Value' 'Reference' 'Structure' 'Scheme')</li> </ul>
Documentation	The ISDA defined value indicating the nature of a difference.

### Diagram

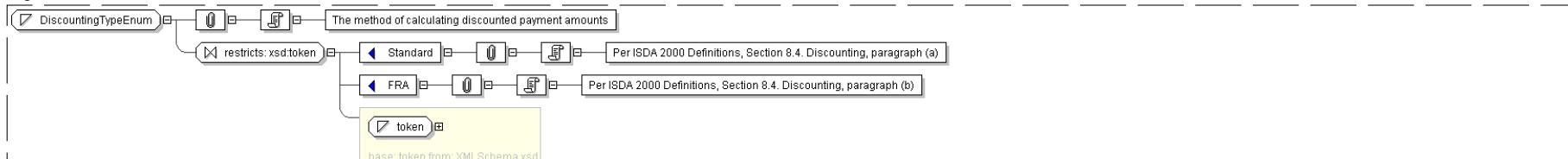


### Schema Component Representation

```
<xsd:simpleType name="DifferenceTypeEnum">
  <xsd:restriction base=" xsd:token ">
    <xsd:enumeration value="Value"/>
    <xsd:enumeration value="Reference"/>
    <xsd:enumeration value="Structure"/>
    <xsd:enumeration value="Scheme"/>
  </xsd:restriction>
</xsd:simpleType>
```

**Simple Type: DiscountingTypeEnum**

<b>Super-types:</b>	<a href="#">xsd:token &lt; DiscountingTypeEnum (by restriction)</a>
<b>Sub-types:</b>	None
<b>Name</b>	DiscountingTypeEnum
<b>Content</b>	<ul style="list-style-type: none"> <li>• Base XSD Type: token</li> <li>• <i>value</i> comes from list: ("Standard","FRA")</li> </ul>
<b>Documentation</b>	The method of calculating discounted payment amounts

**Diagram****Schema Component Representation**

```

<xsd:simpleType name="DiscountingTypeEnum">
  <xsd:restriction base="xsd:token">
    <xsd:enumeration value="Standard"/>
    <xsd:enumeration value="FRA"/>
  </xsd:restriction>
</xsd:simpleType>
  
```

**Simple Type: DisruptionFallbacksEnum**

<b>Super-types:</b>	<a href="#">xsd:token &lt; DisruptionFallbacksEnum (by restriction)</a>
<b>Sub-types:</b>	None
<b>Name</b>	DisruptionFallbacksEnum
<b>Content</b>	<ul style="list-style-type: none"> <li>• Base XSD Type: token</li> <li>• <i>value</i> comes from list: ("AsSpecifiedInMasterAgreement")</li> </ul>
<b>Documentation</b>	The specification of how disruption fallbacks will be represented.

**Diagram****Schema Component Representation**

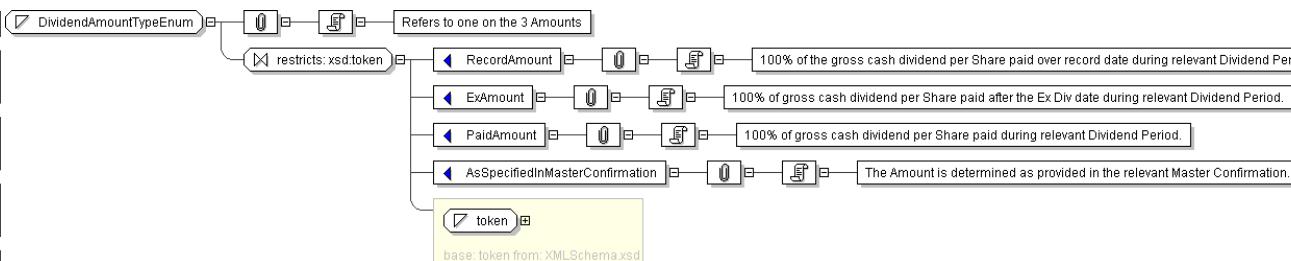
```

<xsd:simpleType name="DisruptionFallbacksEnum">
  <xsd:restriction base="xsd:token">
    <xsd:enumeration value="AsSpecifiedInMasterAgreement"/>
  </xsd:restriction>
</xsd:simpleType>
  
```

**Simple Type: DividendAmountTypeEnum**

<b>Super-types:</b>	<a href="#">xsd:token &lt; DividendAmountTypeEnum (by restriction)</a>
<b>Sub-types:</b>	None
<b>Name</b>	DividendAmountTypeEnum
<b>Content</b>	<ul style="list-style-type: none"> <li>• Base XSD Type: token</li> <li>• <i>value</i> comes from list: ("RecordAmount","ExAmount","PaidAmount","AsSpecifiedInMasterConfirmation")</li> </ul>
<b>Documentation</b>	Refers to one on the 3 Amounts

**Diagram**

**Schema Component Representation**

```

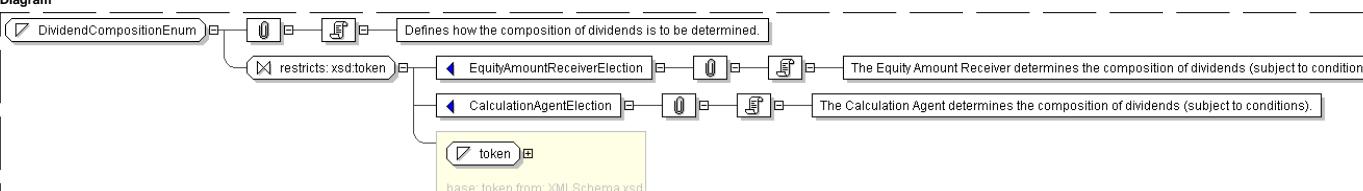
<xsd:simpleType name="DividendAmountTypeEnum">
  <xsd:restriction base=" xsd:token ">
    <xsd:enumeration value="RecordAmount"/>
    <xsd:enumeration value="ExAmount"/>
    <xsd:enumeration value="PaidAmount"/>
    <xsd:enumeration value="AsSpecifiedInMasterConfirmation"/>
  </xsd:restriction>
</xsd:simpleType>
  
```

top

**Simple Type: DividendCompositionEnum**

**Super-types:** xsd:token < **DividendCompositionEnum** (by restriction)  
**Sub-types:** None

<b>Name</b>	DividendCompositionEnum
<b>Content</b>	<ul style="list-style-type: none"> <li>Base XSD Type: token</li> <li>value comes from list: ('EquityAmountReceiverElection' 'CalculationAgentElection')</li> </ul>
<b>Documentation</b>	

**Diagram****Schema Component Representation**

```

<xsd:simpleType name="DividendCompositionEnum">
  <xsd:restriction base=" xsd:token ">
    <xsd:enumeration value="EquityAmountReceiverElection"/>
    <xsd:enumeration value="CalculationAgentElection"/>
  </xsd:restriction>
</xsd:simpleType>
  
```

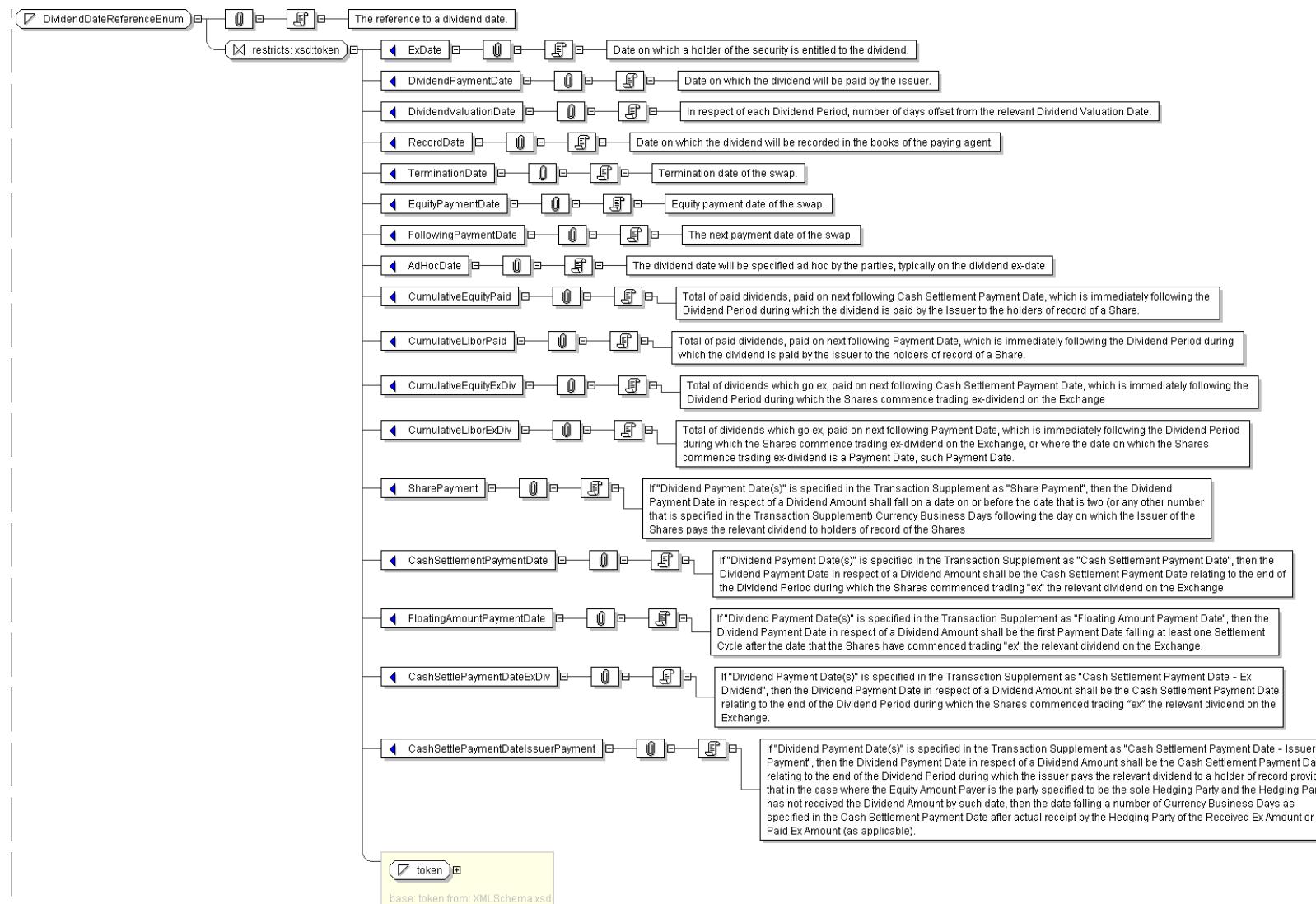
top

**Simple Type: DividendDateReferenceEnum**

**Super-types:** xsd:token < **DividendDateReferenceEnum** (by restriction)  
**Sub-types:** None

<b>Name</b>	DividendDateReferenceEnum
<b>Content</b>	<ul style="list-style-type: none"> <li>Base XSD Type: token</li> <li>value comes from list: ('ExDate' 'DividendPaymentDate' 'DividendValuationDate' 'RecordDate' 'TerminationDate' 'EquityPaymentDate' 'FollowingPaymentDate' 'AdHocDate' 'CumulativeEquityPaid' 'CumulativeLiberPaid' 'CumulativeEquityExDiv' 'CumulativeLiberExDiv' 'SharePayment' 'CashSettlementPaymentDate' 'FloatingAmountPaymentDate' 'CashSettlePaymentDateExDiv' 'CashSettlePaymentDateIssuerPayment')</li> </ul>
<b>Documentation</b>	

**Diagram**



## Schema Component Representation

```

<xsd:simpleType name="DividendDateReferenceEnum">
    <xsd:restriction base="xsd:token">
        <xsd:enumeration value="ExDate"/>
        <xsd:enumeration value="DividendPaymentDate"/>
        <xsd:enumeration value="DividendValuationDate"/>
        <xsd:enumeration value="RecordDate"/>
        <xsd:enumeration value="TerminationDate"/>
        <xsd:enumeration value="EquityPaymentDate"/>
        <xsd:enumeration value="FollowingPaymentDate"/>
        <xsd:enumeration value="AdHocDate"/>
        <xsd:enumeration value="CumulativeEquityPaid"/>
        <xsd:enumeration value="CumulativeLiborPaid"/>
        <xsd:enumeration value="CumulativeEquityExDiv"/>
        <xsd:enumeration value="CumulativeLiborExDiv"/>
        <xsd:enumeration value="SharePayment"/>
        <xsd:enumeration value="CashSettlementPaymentDate"/>
        <xsd:enumeration value="FloatingAmountPaymentDate"/>
        <xsd:enumeration value="CashSettlePaymentDateExDiv"/>
        <xsd:enumeration value="CashSettlePaymentDateIssuerPayment"/>
    </xsd:restriction>
</xsd:simpleType>

```

**Simple Type: DividendEntitlementEnum**

<b>Super-types:</b>	<a href="#">xsd:token &lt; DividendEntitlementEnum (by restriction)</a>
<b>Sub-types:</b>	None
<b>Name</b>	DividendEntitlementEnum
<b>Content</b>	<ul style="list-style-type: none"> <li>Base XSD Type: token</li> <li>value comes from list: (ExDate RecordDate)</li> </ul>
<b>Documentation</b>	The date on which the receiver of the equity return is entitled to the dividend.
<b>Diagram</b>	<pre> classDiagram     class DividendEntitlementEnum {         &lt;&lt;xsd:token&lt; DividendEntitlementEnum (by restriction)&gt;&gt;     }     class ExDate {         &lt;&lt;xsd:enumeration value="ExDate"/&gt;&gt;     }     class RecordDate {         &lt;&lt;xsd:enumeration value="RecordDate"/&gt;&gt;     }     class token {         &lt;&lt;xsd:token from: XMLSchema.xsd&gt;&gt;     }      DividendEntitlementEnum "1" -- "2" ExDate : restricts xsd:token     DividendEntitlementEnum "1" -- "2" RecordDate : restricts xsd:token     ExDate "1" -- "2" token : base: token from: XMLSchema.xsd     RecordDate "1" -- "2" token : base: token from: XMLSchema.xsd   </pre> <p>The diagram illustrates the structure of the DividendEntitlementEnum simple type. It is based on the xsd:token type. It contains two enumerations: ExDate and RecordDate, both of which are derived from the token type. The documentation for the type states that it represents the date on which the receiver of the equity return is entitled to the dividend.</p>
<b>Schema Component Representation</b>	<pre> &lt;xsd:simpleType name="DividendEntitlementEnum"&gt;   &lt;xsd:restriction base=" xsd:token "&gt;     &lt;xsd:enumeration value="ExDate"/&gt;     &lt;xsd:enumeration value="RecordDate"/&gt;   &lt;/xsd:restriction&gt; &lt;/xsd:simpleType&gt;   </pre>

top

**Simple Type: DividendPeriodEnum**

<b>Super-types:</b>	<a href="#">xsd:token &lt; DividendPeriodEnum (by restriction)</a>
<b>Sub-types:</b>	None
<b>Name</b>	DividendPeriodEnum
<b>Content</b>	<ul style="list-style-type: none"> <li>Base XSD Type: token</li> <li>value comes from list: (FirstPeriod SecondPeriod)</li> </ul>
<b>Documentation</b>	Defines the First Period or the Second Period, as specified in the 2002 ISDA Equity Derivatives Definitions.
<b>Diagram</b>	<pre> classDiagram     class DividendPeriodEnum {         &lt;&lt;xsd:token&lt; DividendPeriodEnum (by restriction)&gt;&gt;     }     class FirstPeriod {         &lt;&lt;xsd:enumeration value="FirstPeriod"/&gt;&gt;     }     class SecondPeriod {         &lt;&lt;xsd:enumeration value="SecondPeriod"/&gt;&gt;     }     class token {         &lt;&lt;xsd:token from: XMLSchema.xsd&gt;&gt;     }      DividendPeriodEnum "1" -- "2" FirstPeriod : restricts xsd:token     DividendPeriodEnum "1" -- "2" SecondPeriod : restricts xsd:token     FirstPeriod "1" -- "2" token : base: token from: XMLSchema.xsd     SecondPeriod "1" -- "2" token : base: token from: XMLSchema.xsd   </pre> <p>The diagram illustrates the structure of the DividendPeriodEnum simple type. It is based on the xsd:token type. It contains two enumerations: FirstPeriod and SecondPeriod, both of which are derived from the token type. The documentation for the type states that it defines the First Period or the Second Period, as specified in the 2002 ISDA Equity Derivatives Definitions.</p>
<b>Schema Component Representation</b>	<pre> &lt;xsd:simpleType name="DividendPeriodEnum"&gt;   &lt;xsd:restriction base=" xsd:token "&gt;     &lt;xsd:enumeration value="FirstPeriod"/&gt;     &lt;xsd:enumeration value="SecondPeriod"/&gt;   &lt;/xsd:restriction&gt; &lt;/xsd:simpleType&gt;   </pre>

top

**Simple Type: ElectricityProductTypeEnum**

<b>Super-types:</b>	<a href="#">xsd:token &lt; ElectricityProductTypeEnum (by restriction)</a>
<b>Sub-types:</b>	None
<b>Name</b>	ElectricityProductTypeEnum
<b>Content</b>	<ul style="list-style-type: none"> <li>Base XSD Type: token</li> <li>value comes from list: ('Electricity')</li> </ul>
<b>Documentation</b>	The type of electricity product.
<b>Diagram</b>	<pre> classDiagram     class ElectricityProductTypeEnum {         &lt;&lt;xsd:token&lt; ElectricityProductTypeEnum (by restriction)&gt;&gt;     }     class Electricity {         &lt;&lt;xsd:enumeration value="Electricity"/&gt;&gt;     }     class token {         &lt;&lt;xsd:token from: XMLSchema.xsd&gt;&gt;     }      ElectricityProductTypeEnum "1" -- "2" Electricity : restricts xsd:token     Electricity "1" -- "2" token : base: token from: XMLSchema.xsd   </pre> <p>The diagram illustrates the structure of the ElectricityProductTypeEnum simple type. It is based on the xsd:token type. It contains one enumeration: Electricity, which is derived from the token type. The documentation for the type states that it represents the type of electricity product.</p>

**Schema Component Representation**

```
<xsd:simpleType name="ElectricityProductTypeEnum">
  <xsd:restriction base=" xsd:token ">
    <xsd:enumeration value="Electricity"/>
  </xsd:restriction>
</xsd:simpleType>
```

top

**Simple Type: EquityOptionTypeEnum**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None

<b>Name</b>	<code>EquityOptionTypeEnum</code>
<b>Content</b>	<ul style="list-style-type: none"> <li>Union of following types:           <ul style="list-style-type: none"> <li><code>PutCallEnum</code></li> <li>Locally defined type:               <ul style="list-style-type: none"> <li>Base XSD Type: <code>token</code></li> <li><code>value</code> comes from list: {'Forward'}</li> </ul> </li> </ul> </li> </ul>
<b>Documentation</b>	Specifies an additional Forward type.

**Schema Component Representation**

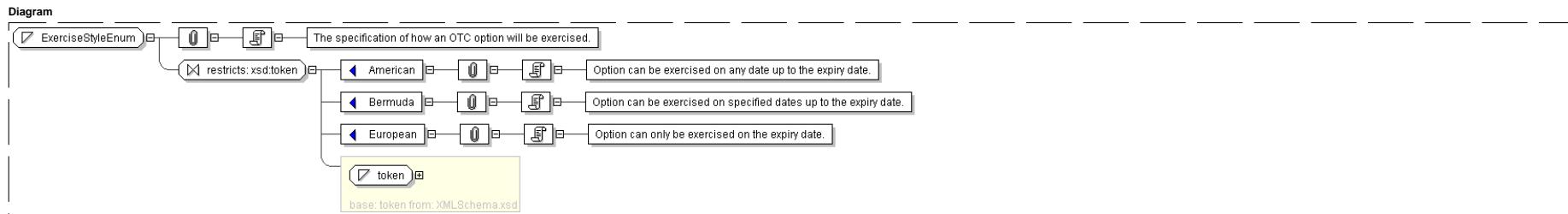
```
<xsd:simpleType name="EquityOptionTypeEnum">
  <xsd:union memberTypes=" PutCallEnum ">
    <xsd:simpleType>
      <xsd:restriction base=" xsd:token ">
        <xsd:enumeration value="Forward" deprecated="true" deprecatedReason="The optionType is to be used if the underlyer does not carry any mention of the resulting trade direction." />
      </xsd:restriction>
    </xsd:simpleType>
  </xsd:union>
</xsd:simpleType>
```

top

**Simple Type: ExerciseStyleEnum**

<b>Super-types:</b>	<code>xsd:token</code> < <code>ExerciseStyleEnum</code> (by restriction)
<b>Sub-types:</b>	None

<b>Name</b>	<code>ExerciseStyleEnum</code>
<b>Content</b>	<ul style="list-style-type: none"> <li>Base XSD Type: <code>token</code></li> <li><code>value</code> comes from list: {'American' 'Bermuda' 'European'}</li> </ul>
<b>Documentation</b>	The specification of how an OTC option will be exercised.



## Schema Component Representation

```
<xsd:simpleType name="ExerciseStyleEnum">
  <xsd:restriction base="xsd:token">
    <xsd:enumeration value="American"/>
    <xsd:enumeration value="Bermuda"/>
    <xsd:enumeration value="European"/>
  </xsd:restriction>
</xsd:simpleType>
```

top

## Simple Type: FPVFinalPriceElectionFallbackEnum

**Super-types:** [xsd:token](#) < **FPVFinalPriceElectionFallbackEnum** (by restriction)  
**Sub-types:** None

<b>Name</b>	FPVFinalPriceElectionFallbackEnum
<b>Content</b>	<ul style="list-style-type: none"> <li>Base XSD Type: token</li> <li>value comes from list: ('FPVClose' 'FPVHedgeExecution')</li> </ul>
<b>Documentation</b>	Specifies the fallback provisions in respect to the applicable Futures Price Valuation.

**Diagram**

```

classDiagram
    class FPVFinalPriceElectionFallbackEnum {
        <<restricts: xsd:token>>
        <<FPVClose>>
        <<FPVHedgeExecution>>
    }
    FPVFinalPriceElectionFallbackEnum --> Note1: In respect of the Early Final Valuation Date, the provisions for FPV Close shall apply.
    FPVFinalPriceElectionFallbackEnum --> Note2: In respect of the Early Final Valuation Date, the provisions for FPV Hedge Execution shall apply.
  
```

base: token from: XMLSchema.xsd

top

## Schema Component Representation

```
<xsd:simpleType name="FPVFinalPriceElectionFallbackEnum">
  <xsd:restriction base="xsd:token">
    <xsd:enumeration value="FPVClose"/>
    <xsd:enumeration value="FPVHedgeExecution"/>
  </xsd:restriction>
</xsd:simpleType>
```

top

## Simple Type: FeeElectionEnum

**Super-types:** [xsd:token](#) < **FeeElectionEnum** (by restriction)  
**Sub-types:** None

<b>Name</b>	FeeElectionEnum
<b>Content</b>	<ul style="list-style-type: none"> <li>Base XSD Type: token</li> <li>value comes from list: ('FlatFee' 'AmortizedFee' 'FundingFee' 'FlatFeeAndFundingFee' 'AmortizedFeeAndFundingFee')</li> </ul>
<b>Documentation</b>	Defines the fee type.

**Diagram**

```

classDiagram
    class FeeElectionEnum {
        <<restricts: xsd:token>>
        <<FlatFee>>
        <<AmortizedFee>>
        <<FundingFee>>
        <<FlatFeeAndFundingFee>>
        <<AmortizedFeeAndFundingFee>>
    }
    FeeElectionEnum --> Note1: The product of (i) the Break Fee Rate multiplied by (ii) the Equity Notional Amount corresponding to the Early Termination Portion.
    FeeElectionEnum --> Note2: The product of (i) the Break Fee Rate multiplied by (ii) the Equity Notional Amount corresponding to the Early Termination Portion multiplied by (iii) the number of days from the Early Termination Date to the later of the Termination Date or the Cash Settlement Payment Date corresponding to the latest Valuation Date.
    FeeElectionEnum --> Note3: The product of (i) the Equity Notional Amount corresponding to the Early Termination Portion multiplied by (ii) the Break Funding Rate multiplied by (iii) the number of days from the Early Termination Date to the next scheduled Reset Date divided by (iv) a number equivalent to the denominator of the Day Count Fraction applicable to the Floating Rate Option.
    FeeElectionEnum --> Note4: Both Flat Fee and Funding Fee are applicable.
    FeeElectionEnum --> Note5: Amortized Fee and Funding Fee are applicable.
  
```

base: token from: XMLSchema.xsd

## Schema Component Representation

```

<xsd:simpleType name="FeeElectioEnum">
  <xsd:restriction base=" xsd:token ">
    <xsd:enumeration value="FlatFee"/>
    <xsd:enumeration value="AmortizedFee"/>
    <xsd:enumeration value="FundingFee"/>
    <xsd:enumeration value="FlatFeeAndFundingFee"/>
    <xsd:enumeration value="AmortizedFeeAndFundingFee"/>
  </xsd:restriction>
</xsd:simpleType>

```

top

**Simple Type: FlatRateEnum**

**Super-types:** [xsd:token](#) < **FlatRateEnum** (by restriction)  
**Sub-types:** None

<b>Name</b>	FlatRateEnum
<b>Content</b>	<ul style="list-style-type: none"> <li>• Base XSD Type: token</li> <li>• <b>value</b> comes from list: ('Fixed' 'Floating')</li> </ul>
<b>Documentation</b>	The method by which the Flat Rate is calculated for a commodity freight transaction.

**Diagram**

base: token from: XMLSchema.xsd

**Schema Component Representation**

```

<xsd:simpleType name="FlatRateEnum">
  <xsd:restriction base=" xsd:token ">
    <xsd:enumeration value="Fixed"/>
    <xsd:enumeration value="Floating"/>
  </xsd:restriction>
</xsd:simpleType>

```

top

**Simple Type: FraDiscountingEnum**

**Super-types:** [xsd:token](#) < **FraDiscountingEnum** (by restriction)  
**Sub-types:** None

<b>Name</b>	FraDiscountingEnum
<b>Content</b>	<ul style="list-style-type: none"> <li>• Base XSD Type: token</li> <li>• <b>value</b> comes from list: ('ISDA' 'AFMA' 'NONE')</li> </ul>
<b>Documentation</b>	The method of FRA discounting, if any, that will apply.

**Diagram**

base: token from: XMLSchema.xsd

**Schema Component Representation**

```

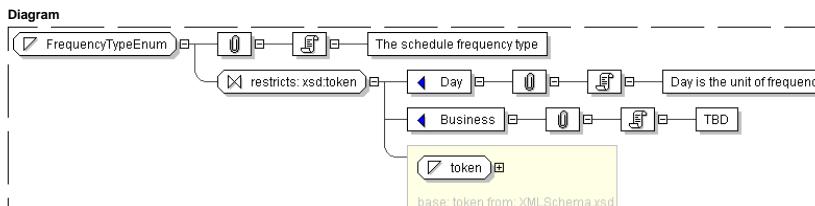
<xsd:simpleType name="FraDiscountingEnum">
  <xsd:restriction base=" xsd:token ">
    <xsd:enumeration value="ISDA"/>
    <xsd:enumeration value="AFMA"/>
    <xsd:enumeration value="NONE"/>
  </xsd:restriction>
</xsd:simpleType>

```

**Simple Type: FrequencyTypeEnum**

<b>Super-types:</b>	<a href="#">xsd:token &lt; FrequencyTypeEnum (by restriction)</a>
<b>Sub-types:</b>	None

<b>Name</b>	FrequencyTypeEnum
<b>Content</b>	<ul style="list-style-type: none"> <li>Base XSD Type: token</li> <li>value comes from list: ('Day' 'Business')</li> </ul>
<b>Documentation</b>	The schedule frequency type.

**Schema Component Representation**

```

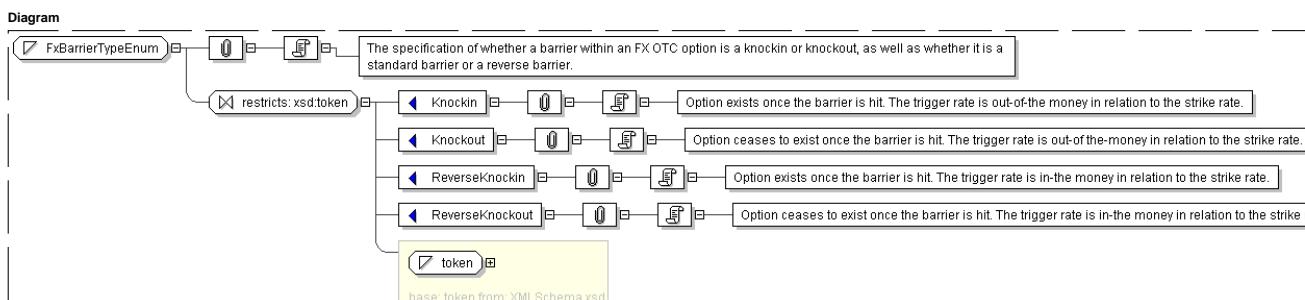
<xsd:simpleType name="FrequencyTypeEnum">
  <xsd:restriction base="xsd:token">
    <xsd:enumeration value="Day"/>
    <xsd:enumeration value="Business"/>
  </xsd:restriction>
</xsd:simpleType>

```

**Simple Type: FxBarrierTypeEnum**

<b>Super-types:</b>	<a href="#">xsd:token &lt; FxBarrierTypeEnum (by restriction)</a>
<b>Sub-types:</b>	None

<b>Name</b>	FxBarrierTypeEnum
<b>Content</b>	<ul style="list-style-type: none"> <li>Base XSD Type: token</li> <li>value comes from list: ('Knockin' 'Knockout' 'ReverseKnockin' 'ReverseKnockout')</li> </ul>
<b>Documentation</b>	The specification of whether a barrier within an FX OTC option is a knockin or knockout, as well as whether it is a standard barrier or a reverse barrier.

**Schema Component Representation**

```

<xsd:simpleType name="FxBarrierTypeEnum">
  <xsd:restriction base="xsd:token">
    <xsd:enumeration value="Knockin"/>
    <xsd:enumeration value="Knockout"/>
    <xsd:enumeration value="ReverseKnockin"/>
    <xsd:enumeration value="ReverseKnockout"/>
  </xsd:restriction>
</xsd:simpleType>

```

**Simple Type: GasProductTypeEnum**

<b>Super-types:</b>	<a href="#">xsd:token &lt; GasProductTypeEnum (by restriction)</a>
<b>Sub-types:</b>	None

Sub-types:

None

Name	GasProductTypeEnum
Content	<ul style="list-style-type: none"> <li>• Base XSD Type: token</li> <li>• value comes from list: ('NaturalGas')</li> </ul>
Documentation	The type of gas product.

Diagram



## Schema Component Representation

```

<xsd:simpleType name="GasProductTypeEnum">
  <xsd:restriction base="xsd:token">
    <xsd:enumeration value="NaturalGas"/>
  </xsd:restriction>
</xsd:simpleType>
  
```

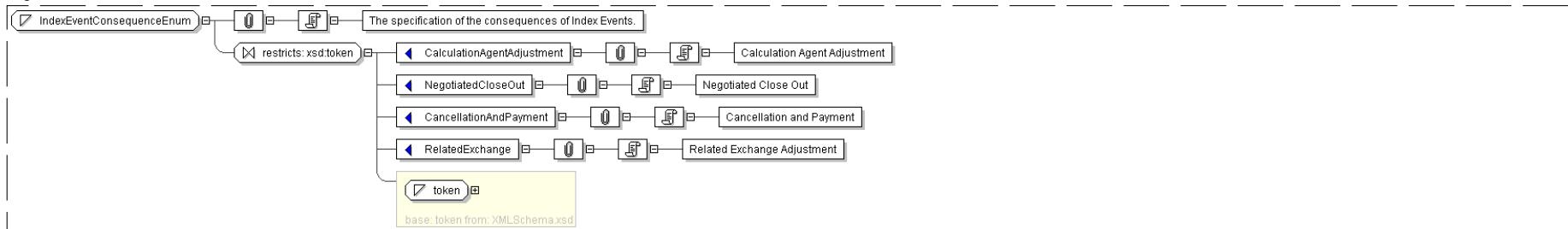
top

## Simple Type: IndexEventConsequenceEnum

Super-types:	<a href="#">xsd:token</a> < <b>IndexEventConsequenceEnum</b> (by restriction)
Sub-types:	None

Name	IndexEventConsequenceEnum
Content	<ul style="list-style-type: none"> <li>• Base XSD Type: token</li> <li>• value comes from list: ('CalculationAgentAdjustment' 'NegotiatedCloseOut' 'CancellationAndPayment' 'RelatedExchange')</li> </ul>
Documentation	The specification of the consequences of Index Events.

Diagram



## Schema Component Representation

```

<xsd:simpleType name="IndexEventConsequenceEnum">
  <xsd:restriction base="xsd:token">
    <xsd:enumeration value="CalculationAgentAdjustment"/>
    <xsd:enumeration value="NegotiatedCloseOut"/>
    <xsd:enumeration value="CancellationAndPayment"/>
    <xsd:enumeration value="RelatedExchange"/>
  </xsd:restriction>
</xsd:simpleType>
  
```

top

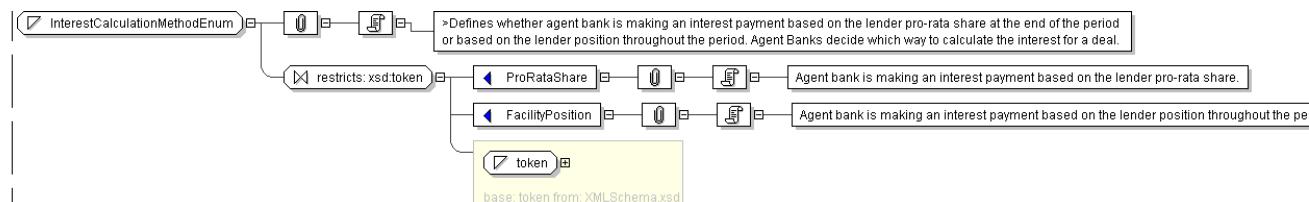
## Simple Type: InterestCalculationMethodEnum

Super-types:	<a href="#">xsd:token</a> < <b>InterestCalculationMethodEnum</b> (by restriction)
Sub-types:	None

Name	InterestCalculationMethodEnum
Content	<ul style="list-style-type: none"> <li>• Base XSD Type: token</li> <li>• value comes from list: ('ProRataShare' 'FacilityPosition')</li> </ul>
Documentation	>Defines whether agent bank is making an interest payment based on the lender pro-rata share at the end of the period or based on the lender position throughout the period. Agent Banks decide which way to calculate the interest for a deal.

Diagram





## Schema Component Representation

```

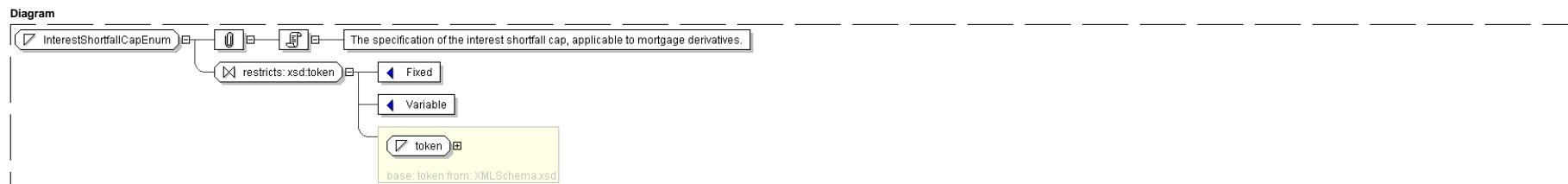
<xsd:simpleType name="InterestCalculationMethodEnum">
  <xsd:restriction base="#token">
    <xsd:enumeration value="ProRataShare"/>
    <xsd:enumeration value="FacilityPosition"/>
  </xsd:restriction>
</xsd:simpleType>
  
```

top

Simple Type: `InterestShortfallCapEnum`

Super-types:	<code>xsd:token</code> < <code>InterestShortfallCapEnum</code> (by restriction)
Sub-types:	None

Name	<code>InterestShortfallCapEnum</code>
Content	<ul style="list-style-type: none"> <li>Base XSD Type: <code>token</code></li> <li><code>value</code> comes from list: ('Fixed' 'Variable')</li> </ul>
Documentation	The specification of the interest shortfall cap, applicable to mortgage derivatives.



## Schema Component Representation

```

<xsd:simpleType name="InterestShortfallCapEnum">
  <xsd:restriction base="#token">
    <xsd:enumeration value="Fixed"/>
    <xsd:enumeration value="Variable"/>
  </xsd:restriction>
</xsd:simpleType>
  
```

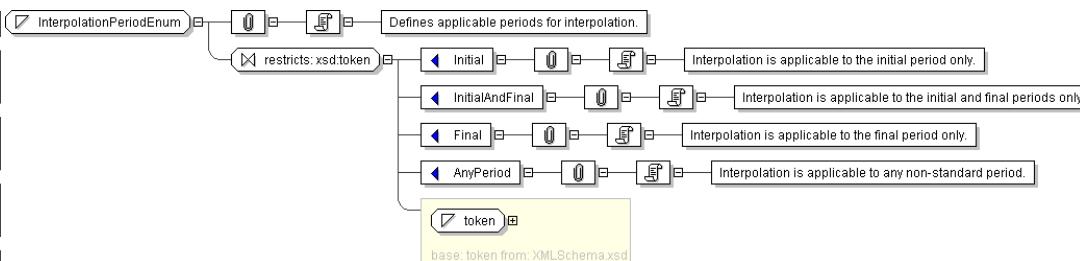
top

Simple Type: `InterpolationPeriodEnum`

Super-types:	<code>xsd:token</code> < <code>InterpolationPeriodEnum</code> (by restriction)
Sub-types:	None

Name	<code>InterpolationPeriodEnum</code>
Content	<ul style="list-style-type: none"> <li>Base XSD Type: <code>token</code></li> <li><code>value</code> comes from list: ('Initial' 'InitialAndFinal' 'Final' 'AnyPeriod')</li> </ul>
Documentation	Defines applicable periods for interpolation.





## Schema Component Representation

```

<xsd:simpleType name="InterpolationPeriodEnum">
  <xsd:restriction base="xsd:token">
    <xsd:enumeration value="Initial"/>
    <xsd:enumeration value="InitialAndFinal"/>
    <xsd:enumeration value="Final"/>
    <xsd:enumeration value="AnyPeriod"/>
  </xsd:restriction>
</xsd:simpleType>
  
```

top

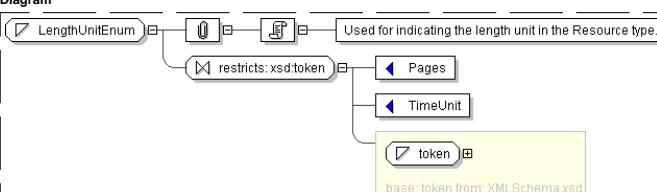
## Simple Type: LengthUnitEnum

Super-types:	xsd:token < LengthUnitEnum (by restriction)
Sub-types:	None
Name	LengthUnitEnum
Content	<ul style="list-style-type: none"> <li>Base XSD Type: token</li> <li>value comes from list: {Pages TimeUnit}</li> </ul>

Documentation

Used for indicating the length unit in the Resource type.

## Diagram



## Schema Component Representation

```

<xsd:simpleType name="LengthUnitEnum">
  <xsd:restriction base="xsd:token">
    <xsd:enumeration value="Pages"/>
    <xsd:enumeration value="TimeUnit"/>
  </xsd:restriction>
</xsd:simpleType>
  
```

top

## Simple Type: MarketDisruptionEventsEnum

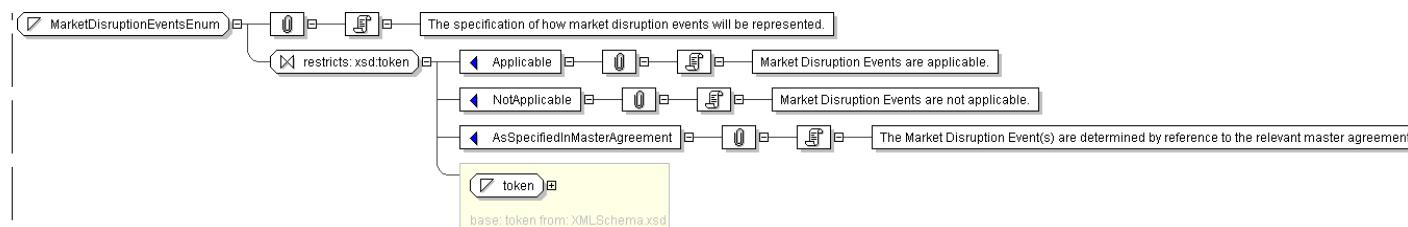
Super-types:	xsd:token < MarketDisruptionEventsEnum (by restriction)
Sub-types:	None
Name	MarketDisruptionEventsEnum
Content	<ul style="list-style-type: none"> <li>Base XSD Type: token</li> <li>value comes from list: {Applicable NotApplicable AsSpecifiedInMasterAgreement}</li> </ul>

Documentation

The specification of how market disruption events will be represented.

## Diagram



**Schema Component Representation**

```

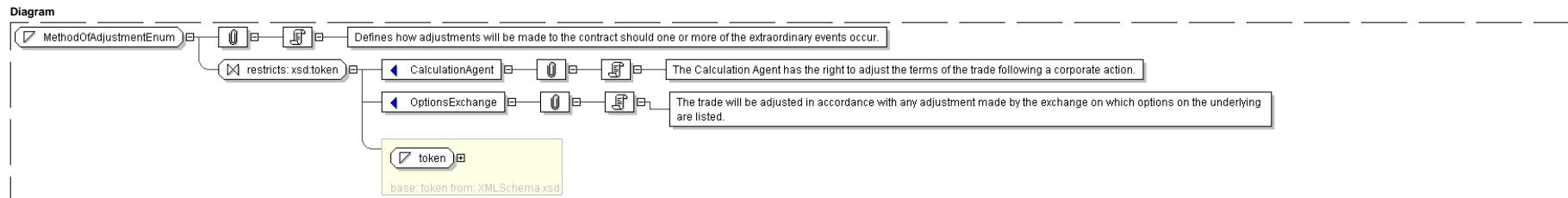
<xsd:simpleType name="MarketDisruptionEventsEnum">
  <xsd:restriction base=" xsd:token ">
    <xsd:enumeration value="Applicable"/>
    <xsd:enumeration value="NotApplicable"/>
    <xsd:enumeration value="AsSpecifiedInMasterAgreement"/>
  </xsd:restriction>
</xsd:simpleType>
  
```

top

**Simple Type: MethodOfAdjustmentEnum**

Super-types:	<a href="#">xsd:token</a> < <b>MethodOfAdjustmentEnum</b> (by restriction)
Sub-types:	None

Name	MethodOfAdjustmentEnum
Content	<ul style="list-style-type: none"> <li>Base XSD Type: token</li> <li>value comes from list: ('CalculationAgent' 'OptionsExchange')</li> </ul>
Documentation	Defines how adjustments will be made to the contract should one or more of the extraordinary events occur.

**Schema Component Representation**

```

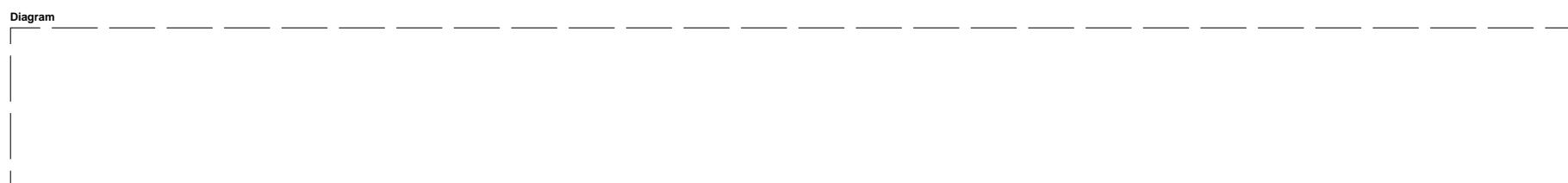
<xsd:simpleType name="MethodOfAdjustmentEnum">
  <xsd:restriction base=" xsd:token ">
    <xsd:enumeration value="CalculationAgent"/>
    <xsd:enumeration value="OptionsExchange"/>
  </xsd:restriction>
</xsd:simpleType>
  
```

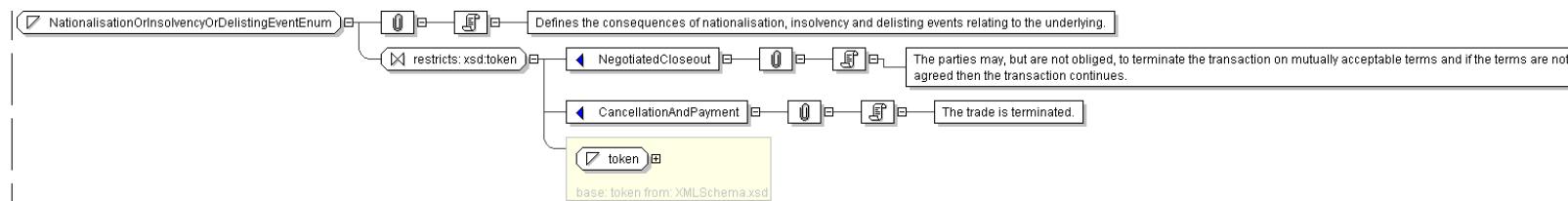
top

**Simple Type: NationalisationOrInsolvencyOrDelistingEventEnum**

Super-types:	<a href="#">xsd:token</a> < <b>NationalisationOrInsolvencyOrDelistingEventEnum</b> (by restriction)
Sub-types:	None

Name	NationalisationOrInsolvencyOrDelistingEventEnum
Content	<ul style="list-style-type: none"> <li>Base XSD Type: token</li> <li>value comes from list: ('NegotiatedCloseout' 'CancellationAndPayment')</li> </ul>
Documentation	Defines the consequences of nationalisation, insolvency and delisting events relating to the underlying.



**Schema Component Representation**

```

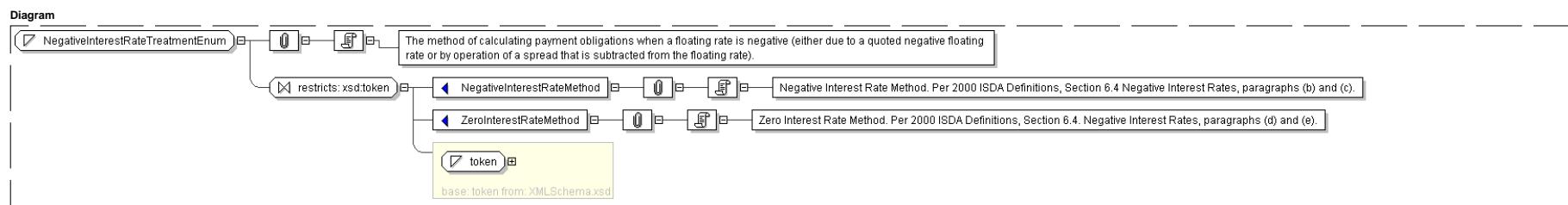
<xsd:simpleType name="NationalisationOrInsolvencyOrDelistingEventEnum">
  <xsd:restriction base=" xsd:token ">
    <xsd:enumeration value="NegotiatedCloseout"/>
    <xsd:enumeration value="CancellationAndPayment"/>
  </xsd:restriction>
</xsd:simpleType>
  
```

top

**Simple Type: NegativeInterestRateTreatmentEnum**

<b>Super-types:</b>	<code>xsd:token &lt; NegativeInterestRateTreatmentEnum (by restriction)</code>
<b>Sub-types:</b>	None

<b>Name</b>	NegativeInterestRateTreatmentEnum
<b>Content</b>	<ul style="list-style-type: none"> <li>Base XSD Type: token</li> <li>value comes from list: ('NegativeInterestRateMethod' 'ZeroInterestRateMethod')</li> </ul>
<b>Documentation</b>	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).

**Schema Component Representation**

```

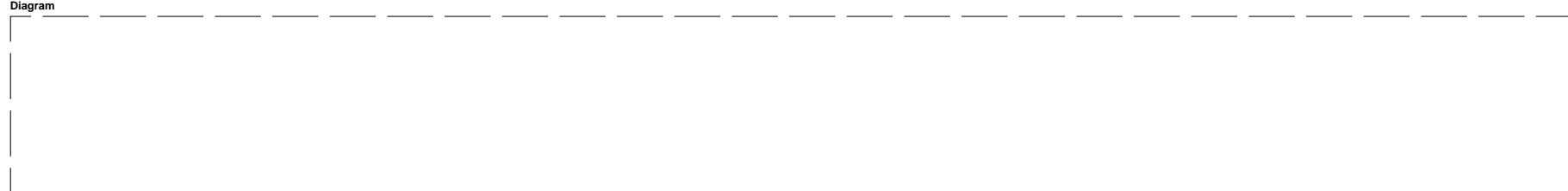
<xsd:simpleType name="NegativeInterestRateTreatmentEnum">
  <xsd:restriction base=" xsd:token ">
    <xsd:enumeration value="NegativeInterestRateMethod"/>
    <xsd:enumeration value="ZeroInterestRateMethod"/>
  </xsd:restriction>
</xsd:simpleType>
  
```

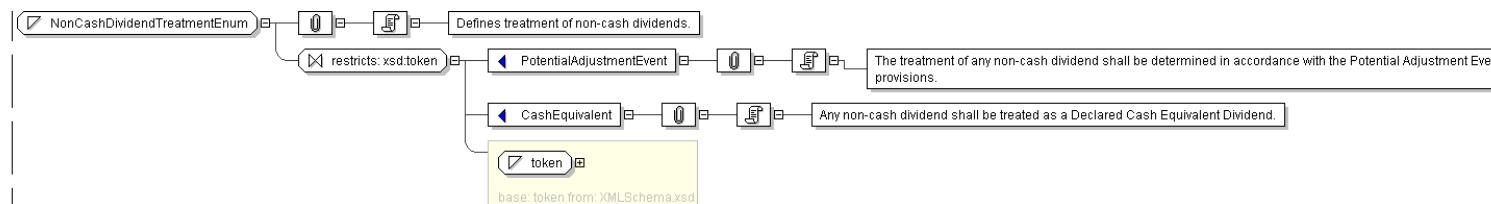
top

**Simple Type: NonCashDividendTreatmentEnum**

<b>Super-types:</b>	<code>xsd:token &lt; NonCashDividendTreatmentEnum (by restriction)</code>
<b>Sub-types:</b>	None

<b>Name</b>	NonCashDividendTreatmentEnum
<b>Content</b>	<ul style="list-style-type: none"> <li>Base XSD Type: token</li> <li>value comes from list: ('PotentialAdjustmentEvent' 'CashEquivalent')</li> </ul>
<b>Documentation</b>	Defines treatment of non-cash dividends.



**Schema Component Representation**

```

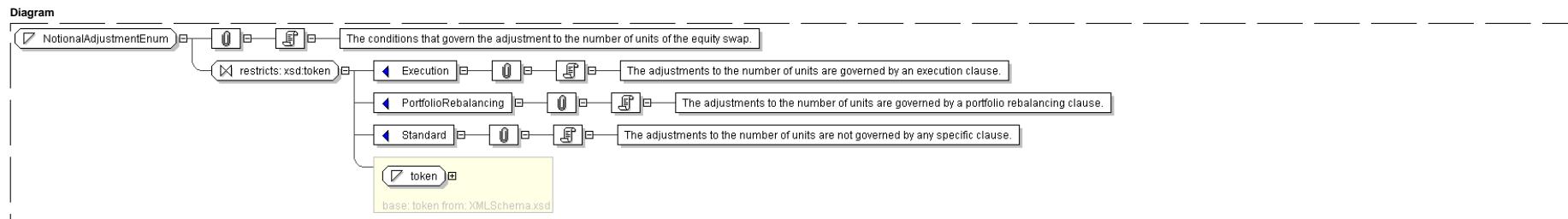
<xsd:simpleType name="NonCashDividendTreatmentEnum">
  <xsd:restriction base=" xsd:token ">
    <xsd:enumeration value="PotentialAdjustmentEvent"/>
    <xsd:enumeration value="CashEquivalent"/>
  </xsd:restriction>
</xsd:simpleType>
  
```

top

**Simple Type: NotionalAdjustmentEnum**

<b>Super-types:</b>	<a href="#">xsd:token</a> < <b>NotionalAdjustmentEnum</b> (by restriction)
<b>Sub-types:</b>	None

<b>Name</b>	NotionalAdjustmentEnum
<b>Content</b>	<ul style="list-style-type: none"> <li>Base XSD Type: token</li> <li>value comes from list: ('Execution' 'PortfolioRebalancing' 'Standard')</li> </ul>
<b>Documentation</b>	The conditions that govern the adjustment to the number of units of the equity swap.

**Schema Component Representation**

```

<xsd:simpleType name="NotionalAdjustmentEnum">
  <xsd:restriction base=" xsd:token ">
    <xsd:enumeration value="Execution"/>
    <xsd:enumeration value="PortfolioRebalancing"/>
    <xsd:enumeration value="Standard"/>
  </xsd:restriction>
</xsd:simpleType>
  
```

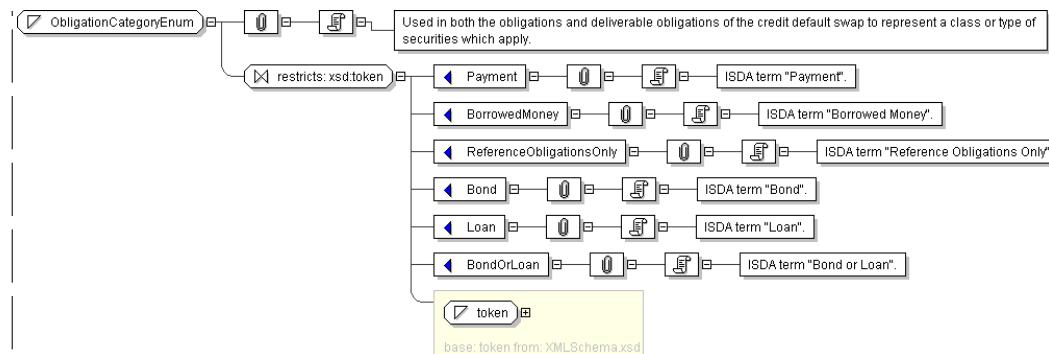
top

**Simple Type: ObligationCategoryEnum**

<b>Super-types:</b>	<a href="#">xsd:token</a> < <b>ObligationCategoryEnum</b> (by restriction)
<b>Sub-types:</b>	None

<b>Name</b>	ObligationCategoryEnum
<b>Content</b>	<ul style="list-style-type: none"> <li>Base XSD Type: token</li> <li>value comes from list: ('Payment' 'BorrowedMoney' 'ReferenceObligationsOnly' 'Bond' 'Loan' 'BondOrLoan')</li> </ul>
<b>Documentation</b>	Used in both the obligations and deliverable obligations of the credit default swap to represent a class or type of securities which apply.





## Schema Component Representation

```

<xsd:simpleType name="ObligationCategoryEnum">
  <xsd:restriction base="#token">
    <xsd:enumeration value="Payment"/>
    <xsd:enumeration value="BorrowedMoney"/>
    <xsd:enumeration value="ReferenceObligationsOnly"/>
    <xsd:enumeration value="Bond"/>
    <xsd:enumeration value="Loan"/>
    <xsd:enumeration value="BondOrLoan"/>
  </xsd:restriction>
</xsd:simpleType>
  
```

top

## Simple Type: OptionTypeEnum

Super-types:	None
Sub-types:	None

Name	OptionTypeEnum
Content	<ul style="list-style-type: none"> <li>Union of following types:           <ul style="list-style-type: none"> <li>PutCallEnum</li> <li>Locally defined type:               <ul style="list-style-type: none"> <li>Base XSD Type: token</li> <li>value comes from list: ('Payer' 'Receiver' 'Straddle')</li> </ul> </li> </ul> </li> </ul>
Documentation	Specifies the type of the option.

## Diagram



## Schema Component Representation

```

<xsd:simpleType name="OptionTypeEnum">
  <xsd:union memberTypes=" PutCallEnum ">
    <xsd:simpleType>
      <xsd:restriction base="#token">
        <xsd:enumeration value="Payer"/>
        <xsd:enumeration value="Receiver"/>
        <xsd:enumeration value="Straddle"/>
      </xsd:restriction>
    </xsd:simpleType>
  </xsd:union>
</xsd:simpleType>
  
```

top

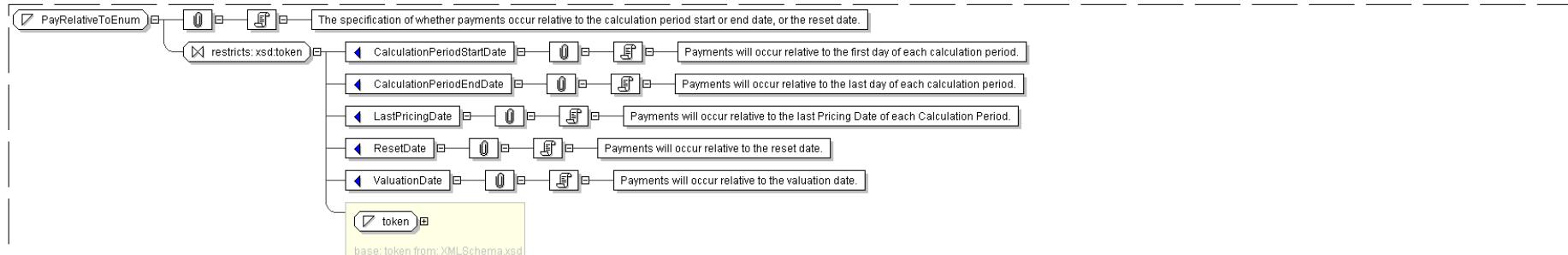
## Simple Type: PayRelativeToEnum

Super-types:	#token < PayRelativeToEnum (by restriction)
Sub-types:	None

Name	PayRelativeToEnum
Content	<ul style="list-style-type: none"> <li>Base XSD Type: token</li> <li>value comes from list: ('CalculationPeriodStartDate' 'CalculationPeriodEndDate' 'LastPricingDate' 'ResetDate' 'ValuationDate')</li> </ul>
Documentation	

**Documentation**

The specification of whether payments occur relative to the calculation period start or end date, or the reset date.

**Diagram****Schema Component Representation**

```

<xsd:simpleType name="PayRelativeTypeEnum">
  <xsd:restriction base="xsd:token">
    <xsd:enumeration value="CalculationPeriodStartDate"/>
    <xsd:enumeration value="CalculationPeriodEndDate"/>
    <xsd:enumeration value="LastPricingDate"/>
    <xsd:enumeration value="ResetDate"/>
    <xsd:enumeration value="ValuationDate"/>
  </xsd:restriction>
</xsd:simpleType>

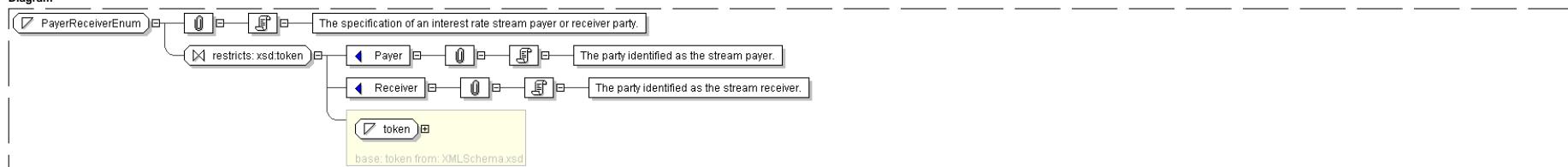
```

top

**Simple Type: PayerReceiverEnum**

Super-types: [xsd:token](#) < **PayerReceiverEnum** (by restriction)  
Sub-types: None

<b>Name</b>	PayerReceiverEnum
<b>Content</b>	<ul style="list-style-type: none"> <li>Base XSD Type: token</li> <li>value comes from list: ('Payer' 'Receiver')</li> </ul>
<b>Documentation</b>	The specification of an interest rate stream payer or receiver party.

**Diagram****Schema Component Representation**

```

<xsd:simpleType name="PayerReceiverEnum">
  <xsd:restriction base="xsd:token">
    <xsd:enumeration value="Payer"/>
    <xsd:enumeration value="Receiver"/>
  </xsd:restriction>
</xsd:simpleType>

```

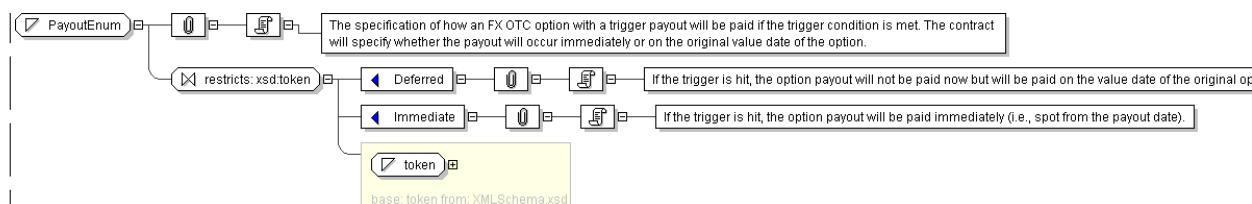
top

**Simple Type: PayoutEnum**

Super-types: [xsd:token](#) < **PayoutEnum** (by restriction)  
Sub-types: None

<b>Name</b>	PayoutEnum
<b>Content</b>	<ul style="list-style-type: none"> <li>Base XSD Type: token</li> <li>value comes from list: ('Deferred' 'Immediate')</li> </ul>
<b>Documentation</b>	The specification of how an FX OTC option with a trigger payout will be paid if the trigger condition is met. The contract will specify whether the payout will occur immediately or on the original value date of the option.

**Diagram**

**Schema Component Representation**

```

<xsd:simpleType name="PayoutEnum">
  <xsd:restriction base=" xsd:token ">
    <xsd:enumeration value="Deferred"/>
    <xsd:enumeration value="Immediate"/>
  </xsd:restriction>
</xsd:simpleType>
  
```

top

**Simple Type: PeriodEnum**

Super-types:	xsd:token < PeriodEnum (by restriction)
Sub-types:	None

Name	PeriodEnum
Content	<ul style="list-style-type: none"> <li>Base XSD Type: token</li> <li>value comes from list: ('D' 'W' 'M' 'Y')</li> </ul>
Documentation	The specification of a time period

**Schema Component Representation**

```

<xsd:simpleType name="PeriodEnum">
  <xsd:restriction base=" xsd:token ">
    <xsd:enumeration value="D"/>
    <xsd:enumeration value="W"/>
    <xsd:enumeration value="M"/>
    <xsd:enumeration value="Y"/>
  </xsd:restriction>
</xsd:simpleType>
  
```

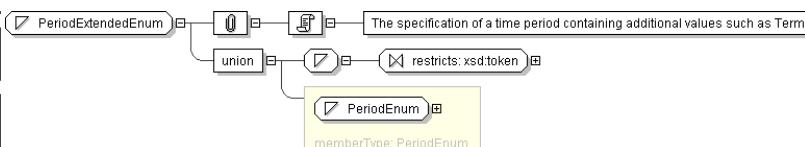
top

**Simple Type: PeriodExtendedEnum**

Super-types:	None
Sub-types:	None

Name	PeriodExtendedEnum
Content	<ul style="list-style-type: none"> <li>Union of following types:           <ul style="list-style-type: none"> <li>PeriodEnum</li> <li>Locally defined type:               <ul style="list-style-type: none"> <li>Base XSD Type: token</li> <li>value comes from list: ('T')</li> </ul> </li> </ul> </li> </ul>
Documentation	The specification of a time period containing additional values such as Term.



**Schema Component Representation**

```

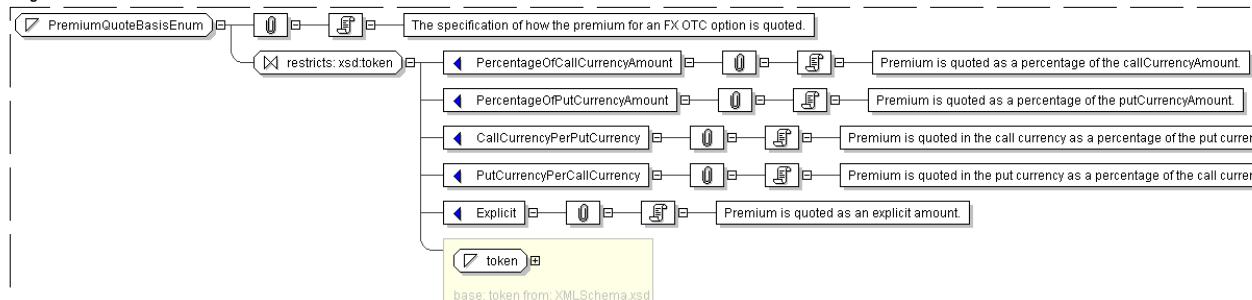
<xsd:simpleType name="PeriodExtendedEnum">
  <xsd:union memberTypes=" PeriodEnum ">
    <xsd:simpleType>
      <xsd:restriction base=" xsd:token ">
        <xsd:enumeration value="T"/>
      </xsd:restriction>
    </xsd:simpleType>
  </xsd:union>
</xsd:simpleType>
  
```

top

**Simple Type: PremiumQuoteBasisEnum**

Super-types:	<a href="#">xsd:token</a> < PremiumQuoteBasisEnum (by restriction)
Sub-types:	None

Name	PremiumQuoteBasisEnum
Content	<ul style="list-style-type: none"> <li>Base XSD Type: token</li> <li>value comes from list: ('PercentageOfCallCurrencyAmount' 'PercentageOfPutCurrencyAmount' 'CallCurrencyPerPutCurrency' 'PutCurrencyPerCallCurrency' 'Explicit')</li> </ul>
Documentation	The specification of how the premium for an FX OTC option is quoted.

**Diagram****Schema Component Representation**

```

<xsd:simpleType name="PremiumQuoteBasisEnum">
  <xsd:restriction base=" xsd:token ">
    <xsd:enumeration value="PercentageOfCallCurrencyAmount"/>
    <xsd:enumeration value="PercentageOfPutCurrencyAmount"/>
    <xsd:enumeration value="CallCurrencyPerPutCurrency"/>
    <xsd:enumeration value="PutCurrencyPerCallCurrency"/>
    <xsd:enumeration value="Explicit"/>
  </xsd:restriction>
</xsd:simpleType>
  
```

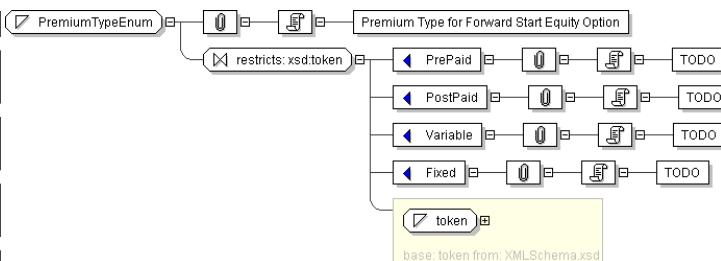
top

**Simple Type: PremiumTypeEnum**

Super-types:	<a href="#">xsd:token</a> < PremiumTypeEnum (by restriction)
Sub-types:	None

Name	PremiumTypeEnum
Content	<ul style="list-style-type: none"> <li>Base XSD Type: token</li> <li>value comes from list: ('PrePaid' 'PostPaid' 'Variable' 'Fixed')</li> </ul>
Documentation	Premium Type for Forward Start Equity Option

**Diagram**

**Schema Component Representation**

```

<xsd:simpleType name="PremiumTypeEnum">
  <xsd:restriction base="#token">
    <xsd:enumeration value="PrePaid"/>
    <xsd:enumeration value="PostPaid"/>
    <xsd:enumeration value="Variable"/>
    <xsd:enumeration value="Fixed"/>
  </xsd:restriction>
</xsd:simpleType>
  
```

top

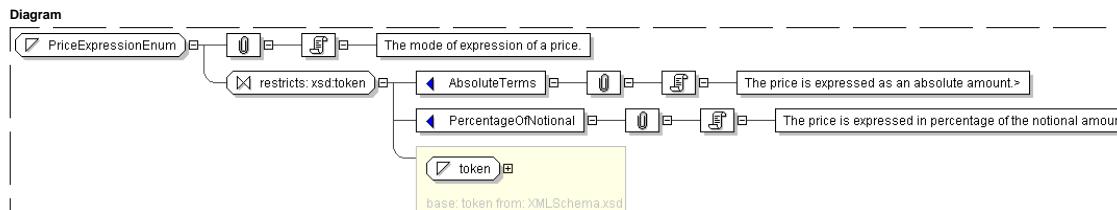
**Simple Type: PriceExpressionEnum**

**Super-types:** *xsd:token* < **PriceExpressionEnum** (by restriction)  
**Sub-types:** None

**Name** PriceExpressionEnum  
**Content**

- Base XSD Type: token
- *value* comes from list: ('AbsoluteTerms'|'PercentageOfNotional')

**Documentation** The mode of expression of a price.

**Schema Component Representation**

```

<xsd:simpleType name="PriceExpressionEnum">
  <xsd:restriction base="#token">
    <xsd:enumeration value="AbsoluteTerms"/>
    <xsd:enumeration value="PercentageOfNotional"/>
  </xsd:restriction>
</xsd:simpleType>
  
```

top

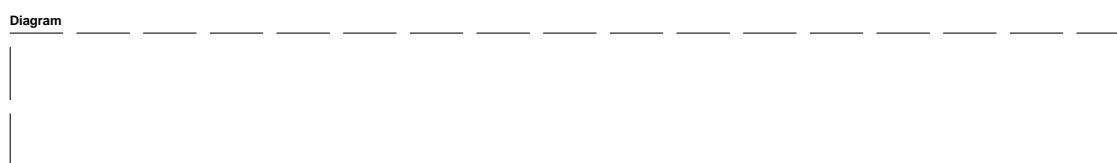
**Simple Type: PutCallEnum**

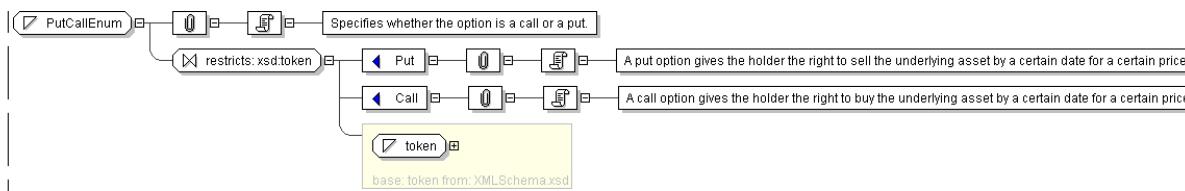
**Super-types:** *xsd:token* < **PutCallEnum** (by restriction)  
**Sub-types:** None

**Name** PutCallEnum  
**Content**

- Base XSD Type: token
- *value* comes from list: ('Put'|'Call')

**Documentation** Specifies whether the option is a call or a put.



**Schema Component Representation**

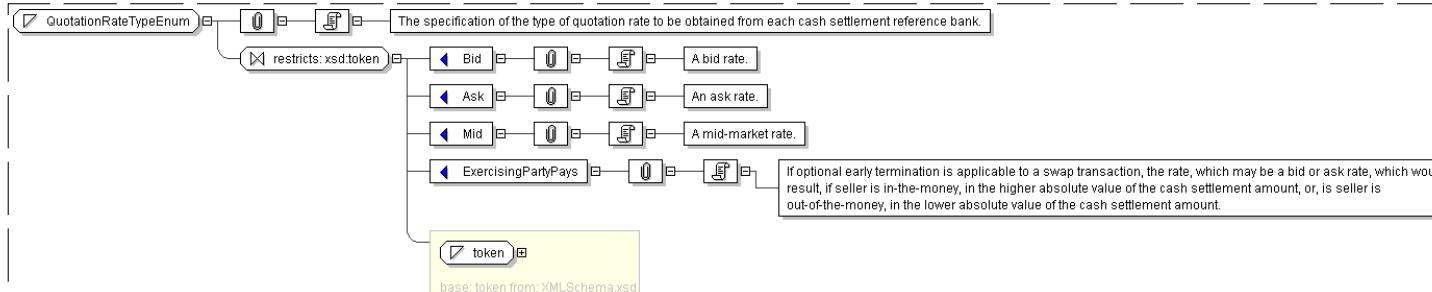
```
<xsd:simpleType name="PutCallEnum">
  <xsd:restriction base=" xsd:token ">
    <xsd:enumeration value="Put"/>
    <xsd:enumeration value="Call"/>
  </xsd:restriction>
</xsd:simpleType>
```

top

**Simple Type: QuotationRateTypeEnum**

**Super-types:** [xsd:token](#) < `QuotationRateTypeEnum` (by restriction)  
**Sub-types:** None

<b>Name</b>	QuotationRateTypeEnum
<b>Content</b>	<ul style="list-style-type: none"> <li>Base XSD Type: token</li> <li><code>value</code> comes from list: ('Bid' 'Ask' 'Mid' 'ExercisingPartyPays')</li> </ul>
<b>Documentation</b>	The specification of the type of quotation rate to be obtained from each cash settlement reference bank.

**Diagram****Schema Component Representation**

```
<xsd:simpleType name="QuotationRateTypeEnum">
  <xsd:restriction base=" xsd:token ">
    <xsd:enumeration value="Bid"/>
    <xsd:enumeration value="Ask"/>
    <xsd:enumeration value="Mid"/>
    <xsd:enumeration value="ExercisingPartyPays"/>
  </xsd:restriction>
</xsd:simpleType>
```

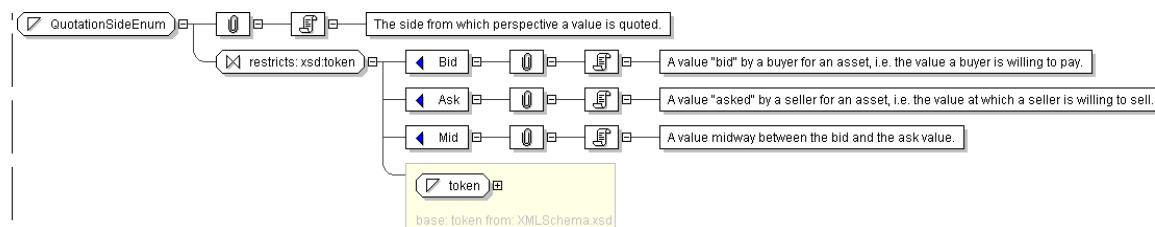
top

**Simple Type: QuotationSideEnum**

**Super-types:** [xsd:token](#) < `QuotationSideEnum` (by restriction)  
**Sub-types:** None

<b>Name</b>	QuotationSideEnum
<b>Content</b>	<ul style="list-style-type: none"> <li>Base XSD Type: token</li> <li><code>value</code> comes from list: ('Bid' 'Ask' 'Mid')</li> </ul>
<b>Documentation</b>	The side from which perspective a value is quoted.

**Diagram**

**Schema Component Representation**

```

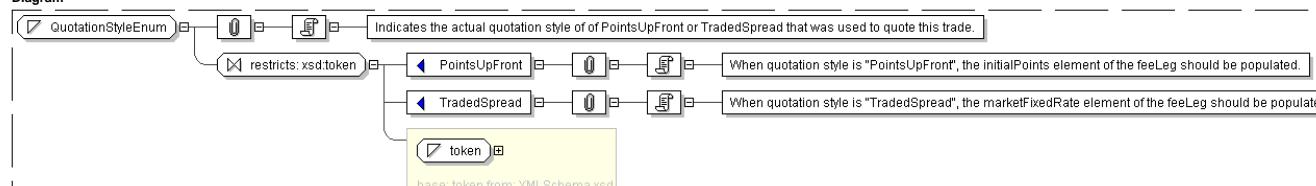
<xsd:simpleType name="QuotationSideEnum">
  <xsd:restriction base="xsd:token">
    <xsd:enumeration value="Bid"/>
    <xsd:enumeration value="Ask"/>
    <xsd:enumeration value="Mid"/>
  </xsd:restriction>
</xsd:simpleType>
  
```

top

**Simple Type: QuotationStyleEnum**

**Super-types:** [xsd:token](#) < QuotationStyleEnum (by restriction)  
**Sub-types:** None

<b>Name</b>	QuotationStyleEnum
<b>Content</b>	<ul style="list-style-type: none"> <li>Base XSD Type: token</li> <li>value comes from list: ('PointsUpFront' 'TradedSpread')</li> </ul>
<b>Documentation</b>	Indicates the actual quotation style of PointsUpFront or TradedSpread that was used to quote this trade.

**Diagram****Schema Component Representation**

```

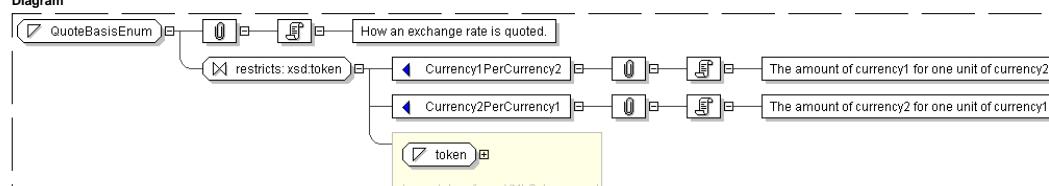
<xsd:simpleType name="QuotationStyleEnum">
  <xsd:restriction base="xsd:token">
    <xsd:enumeration value="PointsUpFront"/>
    <xsd:enumeration value="TradedSpread"/>
  </xsd:restriction>
</xsd:simpleType>
  
```

top

**Simple Type: QuoteBasisEnum**

**Super-types:** [xsd:token](#) < QuoteBasisEnum (by restriction)  
**Sub-types:** None

<b>Name</b>	QuoteBasisEnum
<b>Content</b>	<ul style="list-style-type: none"> <li>Base XSD Type: token</li> <li>value comes from list: ('Currency1PerCurrency2' 'Currency2PerCurrency1')</li> </ul>
<b>Documentation</b>	How an exchange rate is quoted.

**Diagram**

## Schema Component Representation

```
<xsd:simpleType name="QuoteBasisEnum">
  <xsd:restriction base="xsd:token">
    <xsd:enumeration value="Currency1PerCurrency2"/>
    <xsd:enumeration value="Currency2PerCurrency1"/>
  </xsd:restriction>
</xsd:simpleType>
```

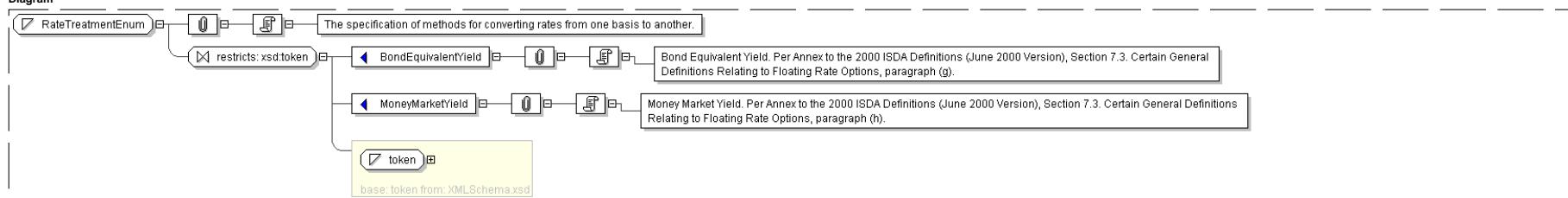
top

## Simple Type: RateTreatmentEnum

Super-types: [xsd:token](#) < **RateTreatmentEnum** (by restriction)  
 Sub-types: None

<b>Name</b>	RateTreatmentEnum
<b>Content</b>	<ul style="list-style-type: none"> <li>• Base XSD Type: token</li> <li>• value comes from list: ('BondEquivalentYield' 'MoneyMarketYield')</li> </ul>
<b>Documentation</b>	The specification of methods for converting rates from one basis to another.

## Diagram



## Schema Component Representation

```
<xsd:simpleType name="RateTreatmentEnum">
  <xsd:restriction base="xsd:token">
    <xsd:enumeration value="BondEquivalentYield"/>
    <xsd:enumeration value="MoneyMarketYield"/>
  </xsd:restriction>
</xsd:simpleType>
```

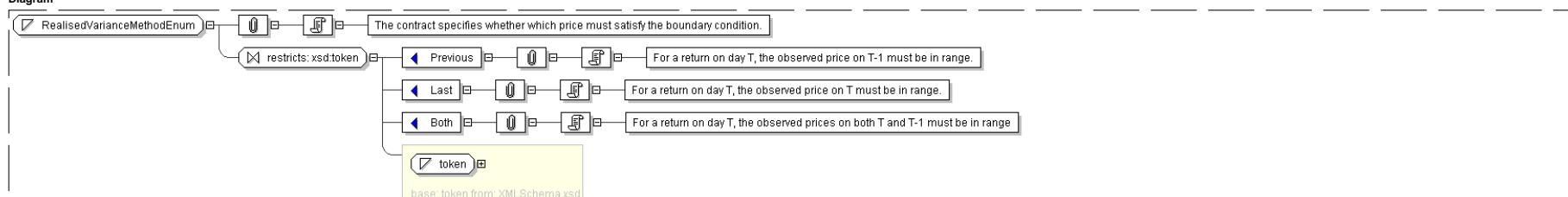
top

## Simple Type: RealisedVarianceMethodEnum

Super-types: [xsd:token](#) < **RealisedVarianceMethodEnum** (by restriction)  
 Sub-types: None

<b>Name</b>	RealisedVarianceMethodEnum
<b>Content</b>	<ul style="list-style-type: none"> <li>• Base XSD Type: token</li> <li>• value comes from list: ('Previous' 'Last' 'Both')</li> </ul>
<b>Documentation</b>	The contract specifies whether which price must satisfy the boundary condition.

## Diagram



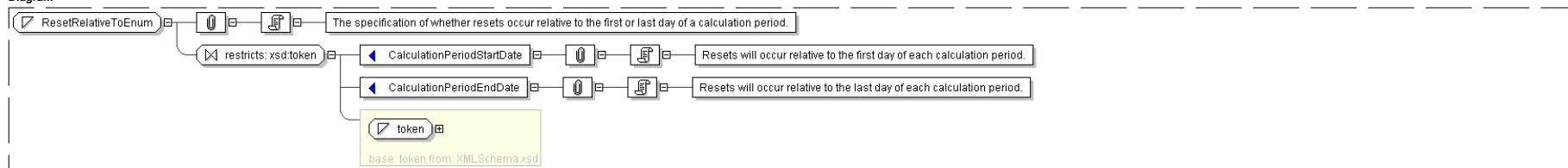
## Schema Component Representation

```
<xsd:simpleType name="RealisedVarianceMethodEnum">
  <xsd:restriction base="xsd:token">
    <xsd:enumeration value="Previous"/>
    <xsd:enumeration value="Last"/>
    <xsd:enumeration value="Both"/>
  </xsd:restriction>
</xsd:simpleType>
```

**Simple Type: ResetRelativeToEnum**

<b>Super-types:</b>	<a href="#">xsd:token &lt; ResetRelativeToEnum</a> (by restriction)
<b>Sub-types:</b>	None

<b>Name</b>	ResetRelativeToEnum
<b>Content</b>	<ul style="list-style-type: none"> <li>• Base XSD Type: token</li> <li>• value comes from list: ('CalculationPeriodStartDate' 'CalculationPeriodEndDate')</li> </ul>
<b>Documentation</b>	The specification of whether resets occur relative to the first or last day of a calculation period.

**Diagram****Schema Component Representation**

```

<xsd:simpleType name="ResetRelativeToEnum">
  <xsd:restriction base=" xsd:token ">
    <xsd:enumeration value="CalculationPeriodStartDate"/>
    <xsd:enumeration value="CalculationPeriodEndDate"/>
  </xsd:restriction>
</xsd:simpleType>
  
```

**Simple Type: ReturnTypeEnum**

<b>Super-types:</b>	<a href="#">xsd:token &lt; ReturnTypeEnum</a> (by restriction)
<b>Sub-types:</b>	None

<b>Name</b>	ReturnTypeEnum
<b>Content</b>	<ul style="list-style-type: none"> <li>• Base XSD Type: token</li> <li>• value comes from list: ('Dividend' 'Price' 'Total')</li> </ul>
<b>Documentation</b>	The type of return associated with the equity swap.

**Diagram****Schema Component Representation**

```

<xsd:simpleType name="ReturnTypeEnum">
  <xsd:restriction base=" xsd:token ">
    <xsd:enumeration value="Dividend"/>
    <xsd:enumeration value="Price"/>
    <xsd:enumeration value="Total"/>
  </xsd:restriction>
</xsd:simpleType>
  
```

**Simple Type: RollConventionEnum**

<b>Super-types:</b>	<a href="#">xsd:token &lt; RollConventionEnum</a> (by restriction)
<b>Sub-types:</b>	None

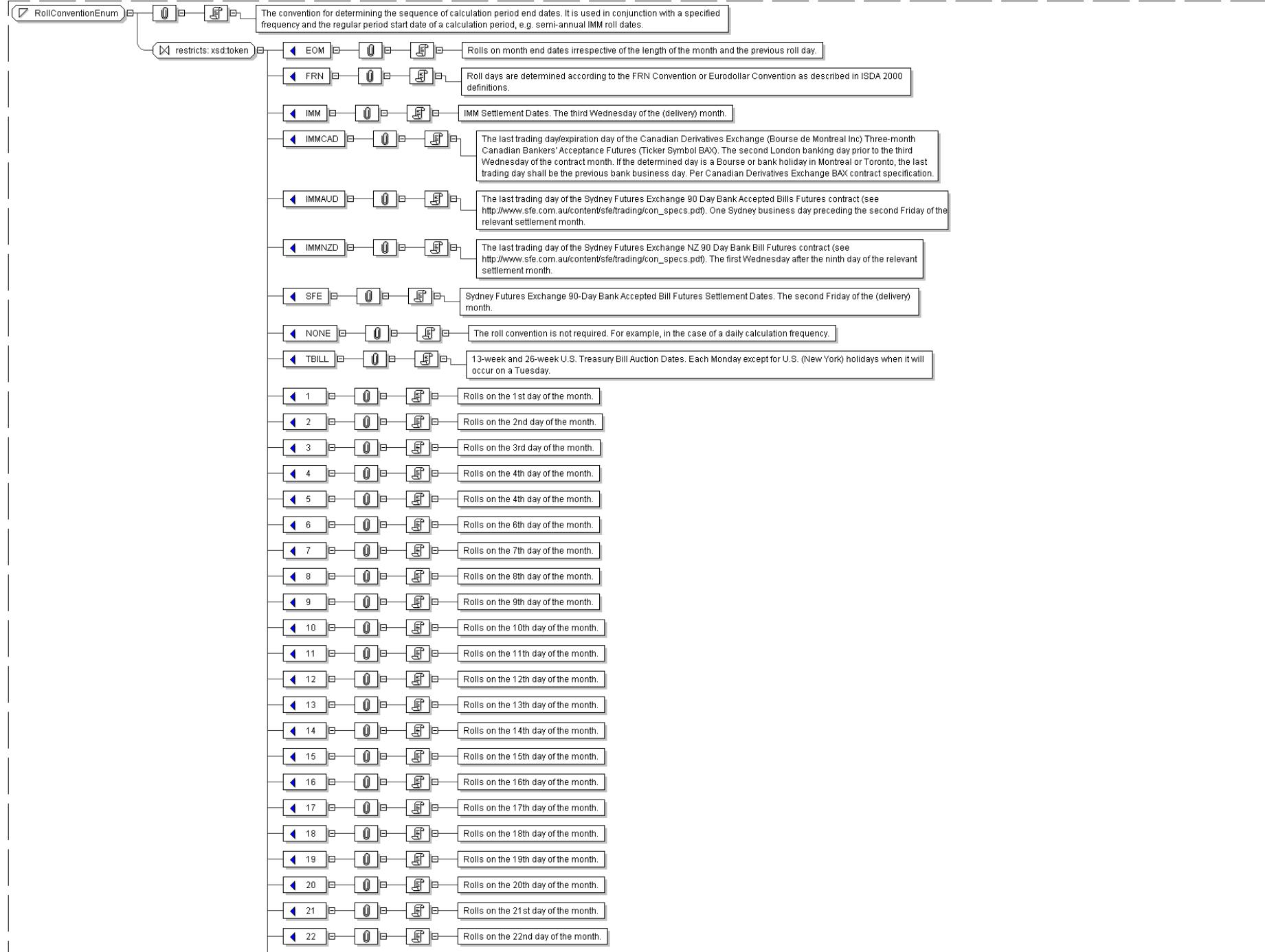
<b>Name</b>	RollConventionEnum
-------------	--------------------

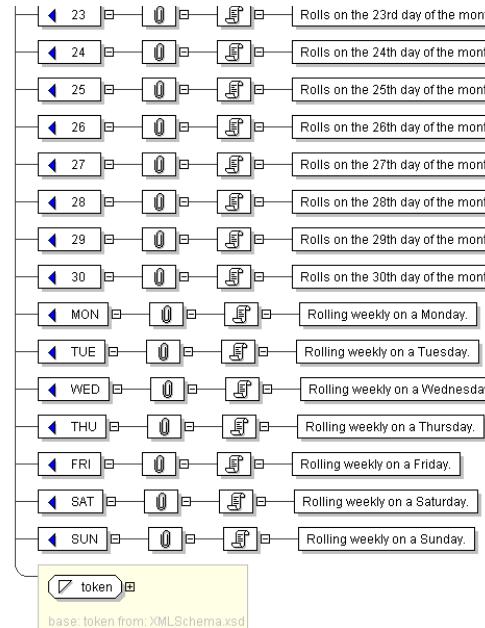
**Content**

- Base XSD Type: token

**Documentation**

- **value** comes from list: {EOM|FRN|IMM|IMMCAD|IMMAUD|IMMNZD|SFE|NONE|TBILL}|1|2|3|4|5|6|7|8|9|10|11|12|13|14|15|16|17|18|19|20|21|22|23|24|25|26|27|28|29|30|MON|TUE|WED|THU|FRI|SAT|SUN}
- 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.

**Diagram**



## Schema Component Representation

```

<xsd:simpleType name="RollConventionEnum">
  <xsd:restriction base="xsd:token">
    <xsd:enumeration value="EOM"/>
    <xsd:enumeration value="FRN"/>
    <xsd:enumeration value="IMM"/>
    <xsd:enumeration value="IMMCAD"/>
    <xsd:enumeration value="IMMAUD"/>
    <xsd:enumeration value="IMMNZN"/>
    <xsd:enumeration value="SPE"/>
    <xsd:enumeration value="NONE"/>
    <xsd:enumeration value="TBILL"/>
    <xsd:enumeration value="1"/>
    <xsd:enumeration value="2"/>
    <xsd:enumeration value="3"/>
    <xsd:enumeration value="4"/>
    <xsd:enumeration value="5"/>
    <xsd:enumeration value="6"/>
    <xsd:enumeration value="7"/>
    <xsd:enumeration value="8"/>
    <xsd:enumeration value="9"/>
    <xsd:enumeration value="10"/>
    <xsd:enumeration value="11"/>
    <xsd:enumeration value="12"/>
    <xsd:enumeration value="13"/>
    <xsd:enumeration value="14"/>
    <xsd:enumeration value="15"/>
    <xsd:enumeration value="16"/>
    <xsd:enumeration value="17"/>
    <xsd:enumeration value="18"/>
    <xsd:enumeration value="19"/>
    <xsd:enumeration value="20"/>
    <xsd:enumeration value="21"/>
    <xsd:enumeration value="22"/>
    <xsd:enumeration value="23"/>
    <xsd:enumeration value="24"/>
    <xsd:enumeration value="25"/>
    <xsd:enumeration value="26"/>
    <xsd:enumeration value="27"/>
    <xsd:enumeration value="28"/>
    <xsd:enumeration value="29"/>
    <xsd:enumeration value="30"/>
    <xsd:enumeration value="MON"/>
    <xsd:enumeration value="TUE"/>
    <xsd:enumeration value="WED"/>
    <xsd:enumeration value="THU"/>
    <xsd:enumeration value="FRI"/>
    <xsd:enumeration value="SAT"/>
    <xsd:enumeration value="SUN"/>
  </xsd:restriction>
</xsd:simpleType>

```

**Simple Type: RoundingDirectionEnum**

**Super-types:** [xsd:token](#) < **RoundingDirectionEnum** (by restriction)

**Sub-types:** None

<b>Name</b>	RoundingDirectionEnum
<b>Content</b>	<ul style="list-style-type: none"> <li>• Base XSD Type: token</li> <li>• value comes from list: ('Up' 'Down' 'Nearest')</li> </ul>
<b>Documentation</b>	The method of rounding a fractional number.

**Diagram**

base: token from: XMLSchema.xsd

**Schema Component Representation**

```
<xsd:simpleType name="RoundingDirectionEnum">
  <xsd:restriction base="xsd:token">
    <xsd:enumeration value="Up"/>
    <xsd:enumeration value="Down"/>
    <xsd:enumeration value="Nearest"/>
  </xsd:restriction>
</xsd:simpleType>
```

top

**Simple Type: SettlementPeriodDurationEnum**

**Super-types:** [xsd:token](#) < **SettlementPeriodDurationEnum** (by restriction)

**Sub-types:** None

<b>Name</b>	SettlementPeriodDurationEnum
<b>Content</b>	<ul style="list-style-type: none"> <li>• Base XSD Type: token</li> <li>• value comes from list: ('2Hours' '1Hour' '30Minutes' '15Minutes')</li> </ul>
<b>Documentation</b>	Defines the Settlement Period Duration for an Electricity Transaction.

**Diagram**

base: token from: XMLSchema.xsd

**Schema Component Representation**

```
<xsd:simpleType name="SettlementPeriodDurationEnum">
  <xsd:restriction base="xsd:token">
    <xsd:enumeration value="2Hours"/>
    <xsd:enumeration value="1Hour"/>
    <xsd:enumeration value="30Minutes"/>
    <xsd:enumeration value="15Minutes"/>
  </xsd:restriction>
</xsd:simpleType>
```

top

**Simple Type: SettlementTypeEnum**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None
<b>Name</b>	SettlementTypeEnum
<b>Content</b>	<ul style="list-style-type: none"> <li>Union of following types:           <ul style="list-style-type: none"> <li><a href="#">CashPhysicalEnum</a></li> <li>Locally defined type:               <ul style="list-style-type: none"> <li>Base XSD Type: token</li> <li>value comes from list: ('Election')</li> </ul> </li> </ul> </li> </ul>

Documentation

Shows how the transaction is to be settled when it is exercised.

**Diagram****Schema Component Representation**

```

<xsd:simpleType name="SettlementTypeEnum">
  <xsd:union memberTypes=" CashPhysicalEnum >
    <xsd:simpleType>
      <xsd:restriction base=" xsd:token ">
        <xsd:enumeration value="Election"/>
      </xsd:restriction>
    </xsd:simpleType>
  </xsd:union>
</xsd:simpleType>
  
```

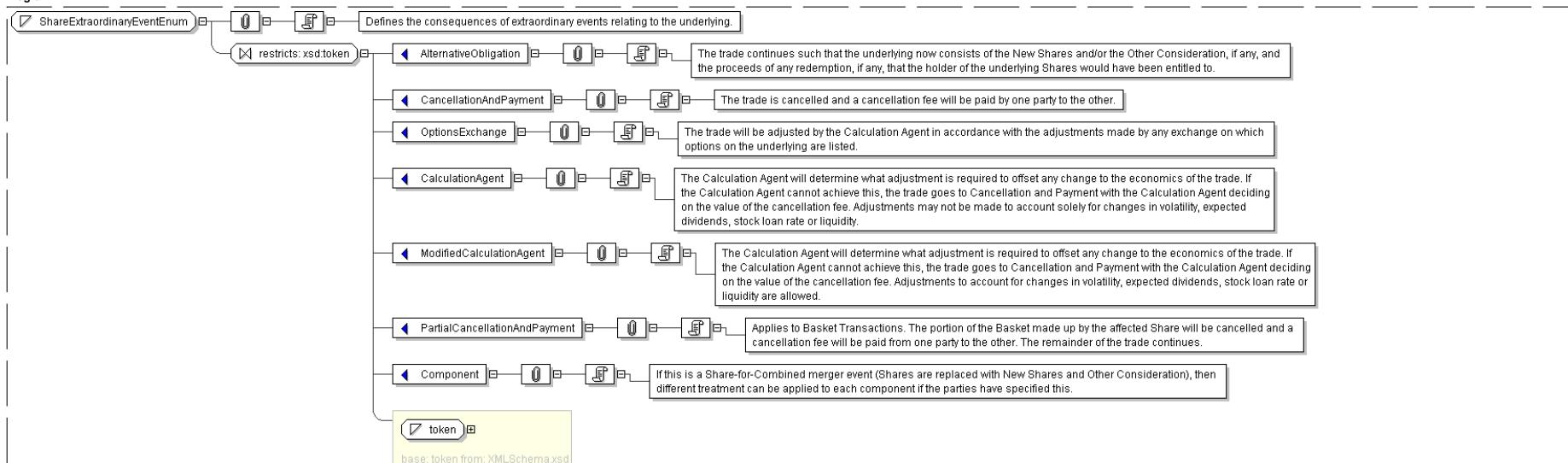
top

**Simple Type: ShareExtraordinaryEventEnum**

<b>Super-types:</b>	<a href="#">xsd:token</a> < ShareExtraordinaryEventEnum (by restriction)
<b>Sub-types:</b>	None
<b>Name</b>	ShareExtraordinaryEventEnum
<b>Content</b>	<ul style="list-style-type: none"> <li>Base XSD Type: token</li> <li>value comes from list: ('AlternativeObligation' 'CancellationAndPayment' 'OptionsExchange' 'CalculationAgent' 'ModifiedCalculationAgent' 'PartialCancellationAndPayment' 'Component')</li> </ul>

Documentation

Defines the consequences of extraordinary events relating to the underlying.

**Diagram****Schema Component Representation**

```

<xsd:simpleType name="ShareExtraordinaryEventEnum">
  
```

```

<xsd:restriction base="#token">
  <xsd:enumeration value="AlternativeObligation"/>
  <xsd:enumeration value="CancellationAndPayment"/>
  <xsd:enumeration value="OptionsExchange"/>
  <xsd:enumeration value="CalculationAgent"/>
  <xsd:enumeration value="ModifiedCalculationAgent"/>
  <xsd:enumeration value="PartialCancellationAndPayment"/>
  <xsd:enumeration value="Component"/>
</xsd:restriction>
</xsd:simpleType>

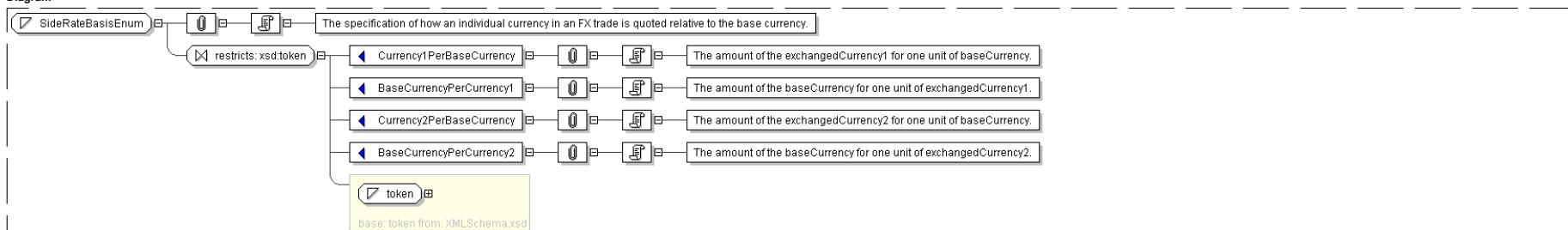
```

top

**Simple Type: SideRateBasisEnum**

Super-types: [xsd:token](#) < **SideRateBasisEnum** (by restriction)  
 Sub-types: None

<b>Name</b>	SideRateBasisEnum
<b>Content</b>	<ul style="list-style-type: none"> <li>Base XSD Type: token</li> <li><b>value</b> comes from list: ('Currency1PerBaseCurrency' 'BaseCurrencyPerCurrency1' 'Currency2PerBaseCurrency' 'BaseCurrencyPerCurrency2')</li> </ul>
<b>Documentation</b>	The specification of how an individual currency in an FX trade is quoted relative to the base currency.

**Diagram****Schema Component Representation**

```

<xsd:simpleType name="SideRateBasisEnum">
  <xsd:restriction base="#token">
    <xsd:enumeration value="Currency1PerBaseCurrency"/>
    <xsd:enumeration value="BaseCurrencyPerCurrency1"/>
    <xsd:enumeration value="Currency2PerBaseCurrency"/>
    <xsd:enumeration value="BaseCurrencyPerCurrency2"/>
  </xsd:restriction>
</xsd:simpleType>

```

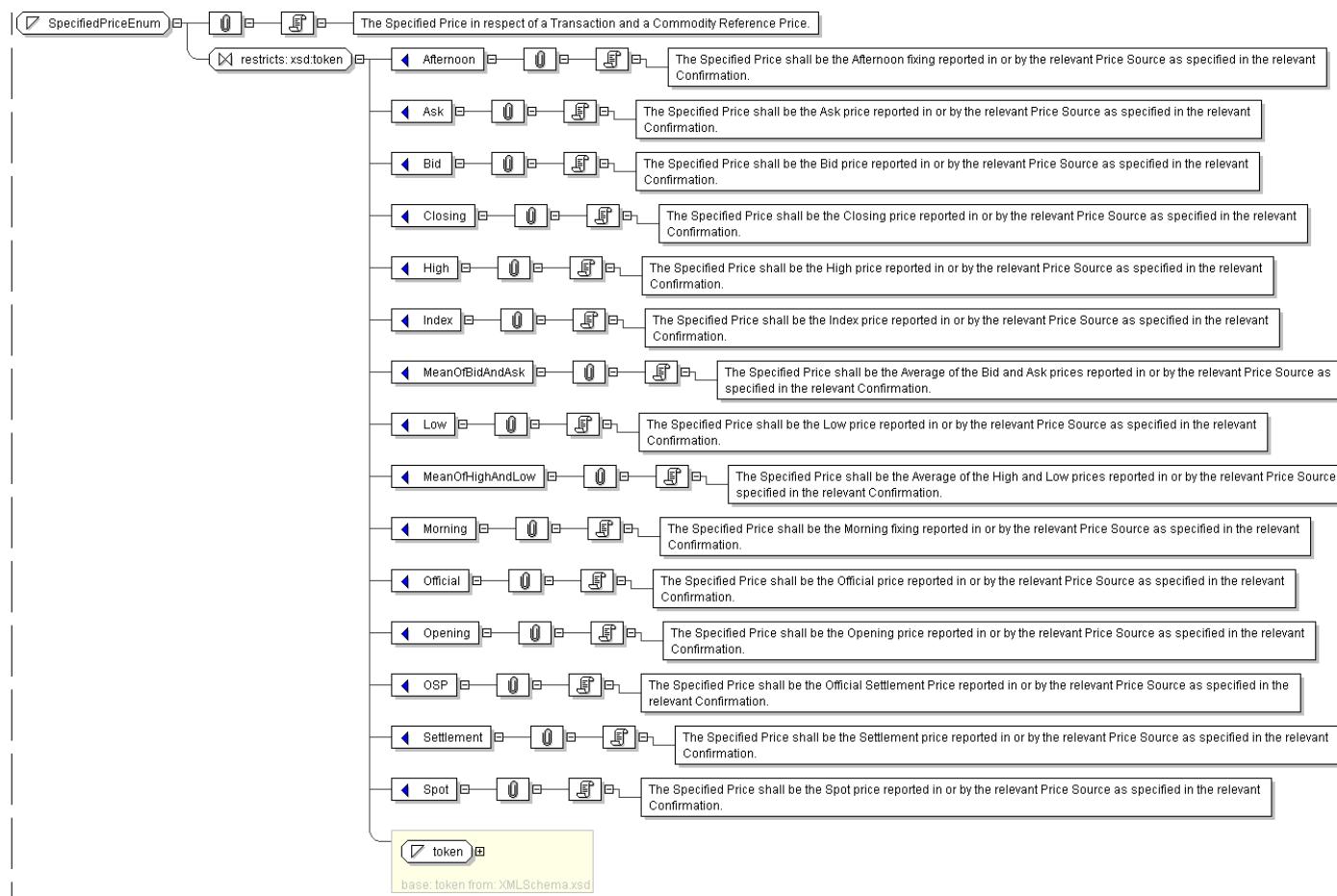
top

**Simple Type: SpecifiedPriceEnum**

Super-types: [xsd:token](#) < **SpecifiedPriceEnum** (by restriction)  
 Sub-types: None

<b>Name</b>	SpecifiedPriceEnum
<b>Content</b>	<ul style="list-style-type: none"> <li>Base XSD Type: token</li> <li><b>value</b> comes from list: ('Afternoon' 'Ask' 'Bid' 'Closing' 'High' 'Index' 'Low' 'MeanOfHighAndLow' 'Morning' 'Official' 'Opening' 'OSP' 'Settlement' 'Spot')</li> </ul>
<b>Documentation</b>	The Specified Price in respect of a Transaction and a Commodity Reference Price.

**Diagram**

**Schema Component Representation**

```

<xsd:simpleType name="SpecifiedPriceEnum">
  <xsd:restriction base="#token">
    <xsd:enumeration value="Afternoon"/>
    <xsd:enumeration value="Ask"/>
    <xsd:enumeration value="Bid"/>
    <xsd:enumeration value="Closing"/>
    <xsd:enumeration value="High"/>
    <xsd:enumeration value="Index"/>
    <xsd:enumeration value="MeanOfBidAndAsk"/>
    <xsd:enumeration value="Low"/>
    <xsd:enumeration value="MeanOfHighAndLow"/>
    <xsd:enumeration value="Morning"/>
    <xsd:enumeration value="Official"/>
    <xsd:enumeration value="Opening"/>
    <xsd:enumeration value="OSP"/>
    <xsd:enumeration value="Settlement"/>
    <xsd:enumeration value="Spot"/>
  </xsd:restriction>
</xsd:simpleType>

```

top

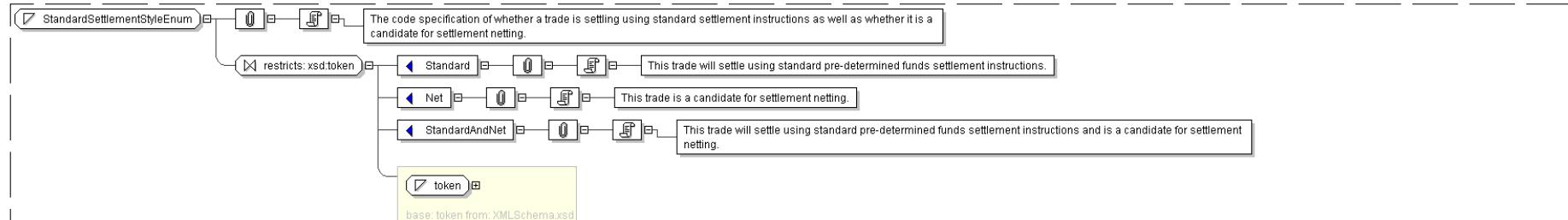
**Simple Type: StandardSettlementStyleEnum**

<b>Super-types:</b>	<code>xsd:token</code> < <code>StandardSettlementStyleEnum</code> (by restriction)
<b>Sub-types:</b>	None
<b>Name</b>	<code>StandardSettlementStyleEnum</code>
<b>Content</b>	<ul style="list-style-type: none"> <li>Base XSD Type: <code>token</code></li> <li><code>value</code> comes from list: ("Standard" "Net" "StandardAndNet")</li> </ul>

## Documentation

The code specification of whether a trade is settling using standard settlement instructions as well as whether it is a candidate for settlement netting.

## Diagram



## Schema Component Representation

```

<xsd:simpleType name="StandardSettlementStyleEnum">
  <xsd:restriction base="xsd:token">
    <xsd:enumeration value="Standard"/>
    <xsd:enumeration value="Net"/>
    <xsd:enumeration value="StandardAndNet"/>
  </xsd:restriction>
</xsd:simpleType>
  
```

top

## Simple Type: StepRelativeToEnum

Super-types:  
Sub-types:

xsd:token < StepRelativeToEnum (by restriction)  
None

Name  
Content

StepRelativeToEnum

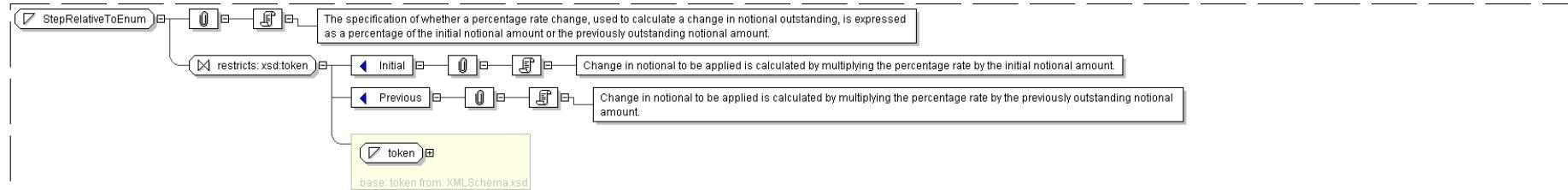
- Base XSD Type: token

- *value* comes from list: ('Initial'|'Previous')

Documentation

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.

## Diagram



## Schema Component Representation

```

<xsd:simpleType name="StepRelativeToEnum">
  <xsd:restriction base="xsd:token">
    <xsd:enumeration value="Initial"/>
    <xsd:enumeration value="Previous"/>
  </xsd:restriction>
</xsd:simpleType>
  
```

top

## Simple Type: StrikeQuoteBasisEnum

Super-types:  
Sub-types:

xsd:token < StrikeQuoteBasisEnum (by restriction)  
None

Name  
Content

StrikeQuoteBasisEnum

- Base XSD Type: token

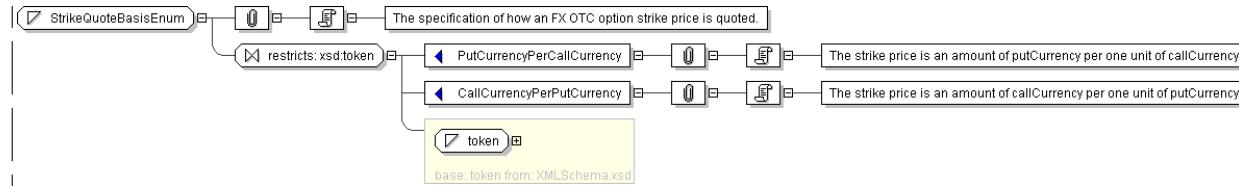
- *value* comes from list: ('PutCurrencyPerCallCurrency'|'CallCurrencyPerPutCurrency')

Documentation

The specification of how an FX OTC option strike price is quoted.

## Diagram



**Schema Component Representation**

```

<xsd:simpleType name="StrikeQuoteBasisEnum">
  <xsd:restriction base=" xsd:token ">
    <xsd:enumeration value="PutCurrencyPerCallCurrency"/>
    <xsd:enumeration value="CallCurrencyPerPutCurrency"/>
  </xsd:restriction>
</xsd:simpleType>
  
```

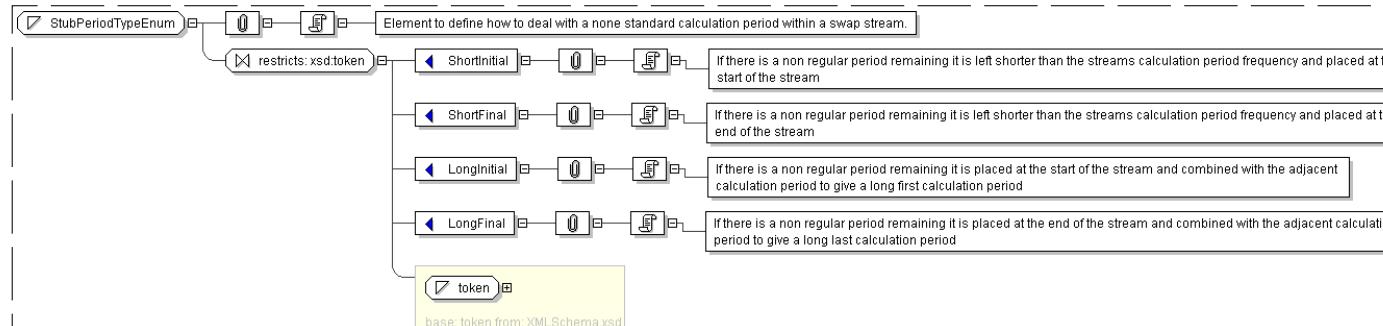
top

**Simple Type: StubPeriodTypeEnum**

**Super-types:** xsd:token < StubPeriodTypeEnum (by restriction)

**Sub-types:** None

<b>Name</b>	StubPeriodTypeEnum
<b>Content</b>	<ul style="list-style-type: none"> <li>Base XSD Type: token</li> <li>value comes from list: ('ShortInitial' 'ShortFinal' 'LongInitial' 'LongFinal')</li> </ul>
<b>Documentation</b>	Element to define how to deal with a none standard calculation period within a swap stream.

**Diagram****Schema Component Representation**

```

<xsd:simpleType name="StubPeriodTypeEnum">
  <xsd:restriction base=" xsd:token ">
    <xsd:enumeration value="ShortInitial"/>
    <xsd:enumeration value="ShortFinal"/>
    <xsd:enumeration value="LongInitial"/>
    <xsd:enumeration value="LongFinal"/>
  </xsd:restriction>
</xsd:simpleType>
  
```

top

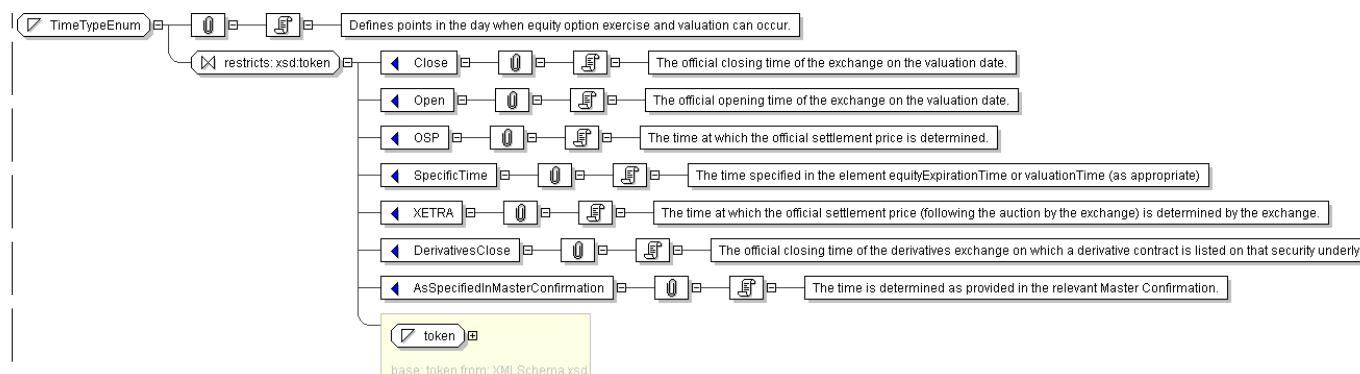
**Simple Type: TimeTypeEnum**

**Super-types:** xsd:token < TimeTypeEnum (by restriction)

**Sub-types:** None

<b>Name</b>	TimeTypeEnum
<b>Content</b>	<ul style="list-style-type: none"> <li>Base XSD Type: token</li> <li>value comes from list: ('Close' 'Open' 'OSP' 'SpecificTime' 'XETRA' 'DerivativesClose' 'AsSpecifiedInMasterConfirmation')</li> </ul>
<b>Documentation</b>	Defines points in the day when equity option exercise and valuation can occur.

**Diagram**



## Schema Component Representation

```

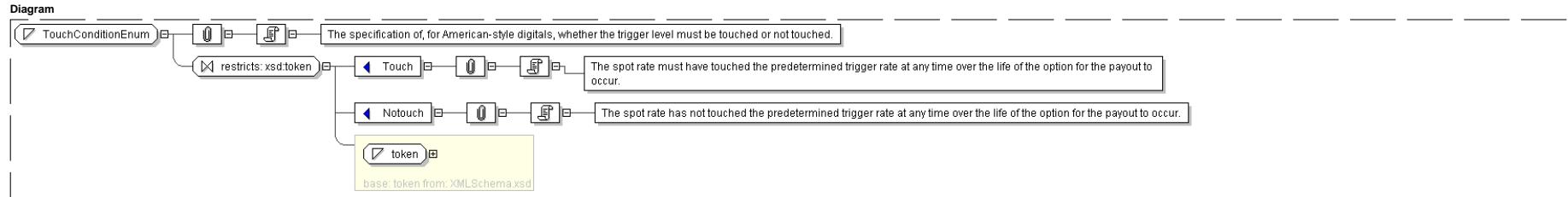
<xsd:simpleType name="TimeTypeEnum">
  <xsd:restriction base="#token">
    <xsd:enumeration value="Close"/>
    <xsd:enumeration value="Open"/>
    <xsd:enumeration value="OSP"/>
    <xsd:enumeration value="SpecificTime"/>
    <xsd:enumeration value="XETRA"/>
    <xsd:enumeration value="DerivativesClose"/>
    <xsd:enumeration value="AsSpecifiedInMasterConfirmation"/>
  </xsd:restriction>
</xsd:simpleType>
  
```

top

## Simple Type: TouchConditionEnum

Super-types:	<a href="#">xsd:token</a> < <b>TouchConditionEnum</b> (by restriction)
Sub-types:	None
Name	TouchConditionEnum
Content	<ul style="list-style-type: none"> <li>Base XSD Type: token</li> <li>value comes from list: ('Touch' 'Notouch')</li> </ul>

Documentation The specification of, for American-style digitals, whether the trigger level must be touched or not touched.



## Schema Component Representation

```

<xsd:simpleType name="TouchConditionEnum">
  <xsd:restriction base="#token">
    <xsd:enumeration value="Touch"/>
    <xsd:enumeration value="Notouch"/>
  </xsd:restriction>
</xsd:simpleType>
  
```

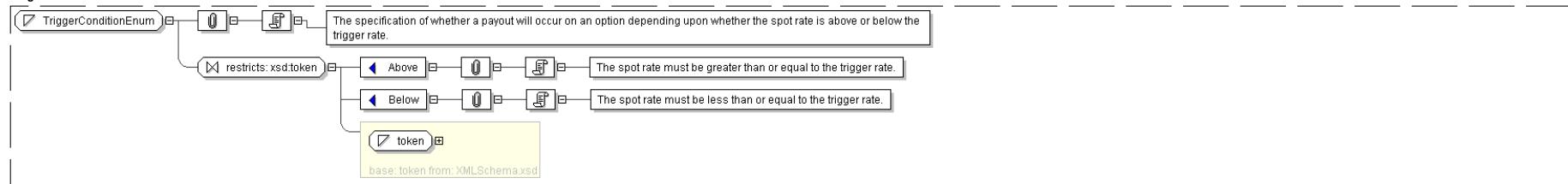
top

## Simple Type: TriggerConditionEnum

Super-types:	<a href="#">xsd:token</a> < <b>TriggerConditionEnum</b> (by restriction)
Sub-types:	None
Name	TriggerConditionEnum
Content	<ul style="list-style-type: none"> <li>Base XSD Type: token</li> <li>value comes from list: ('Above' 'Below')</li> </ul>

Documentation The specification of whether a payout will occur on an option depending upon whether the spot rate is above or below the trigger rate.

## Diagram



## Schema Component Representation

```

<xsd:simpleType name="TriggerConditionEnum">
  <xsd:restriction base="#token">
    <xsd:enumeration value="Above"/>
    <xsd:enumeration value="Below"/>
  </xsd:restriction>
</xsd:simpleType>
  
```

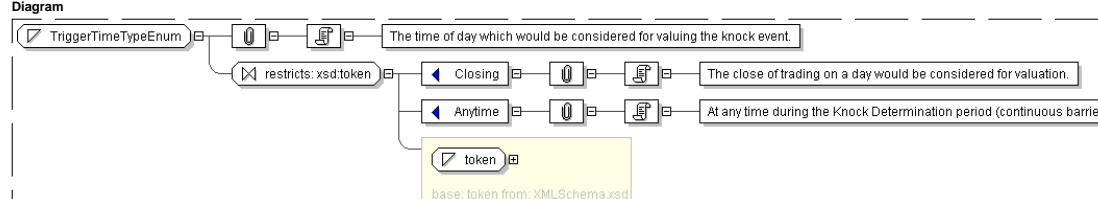
top

Simple Type: `TriggerTimeTypeEnum`

Super-types:	<code>xsd:token</code> < <code>TriggerTimeTypeEnum</code> (by restriction)
Sub-types:	None

Name	TriggerTimeTypeEnum
Content	<ul style="list-style-type: none"> <li>Base XSD Type: <code>token</code></li> <li><code>value</code> comes from list: ('Closing' 'Anytime')</li> </ul>
Documentation	

The time of day which would be considered for valuing the knock event.



## Schema Component Representation

```

<xsd:simpleType name="TriggerTimeTypeEnum">
  <xsd:restriction base="#token">
    <xsd:enumeration value="Closing"/>
    <xsd:enumeration value="Anytime"/>
  </xsd:restriction>
</xsd:simpleType>
  
```

top

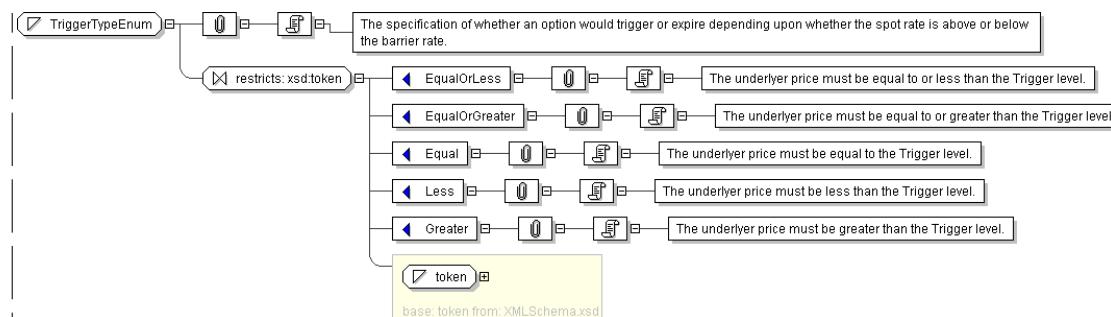
Simple Type: `TriggerTypeEnum`

Super-types:	<code>xsd:token</code> < <code>TriggerTypeEnum</code> (by restriction)
Sub-types:	None

Name	TriggerTypeEnum
Content	<ul style="list-style-type: none"> <li>Base XSD Type: <code>token</code></li> <li><code>value</code> comes from list: ('EqualOrLess' 'EqualOrGreater' 'Equal' 'Less' 'Greater')</li> </ul>
Documentation	

## Diagram



**Schema Component Representation**

```

<xsd:simpleType name="TriggerTypeEnum">
    <xsd:restriction base=" xsd:token ">
        <xsd:enumeration value="EqualOrLess"/>
        <xsd:enumeration value="EqualOrGreater"/>
        <xsd:enumeration value="Equal"/>
        <xsd:enumeration value="Less"/>
        <xsd:enumeration value="Greater"/>
    </xsd:restriction>
</xsd:simpleType>

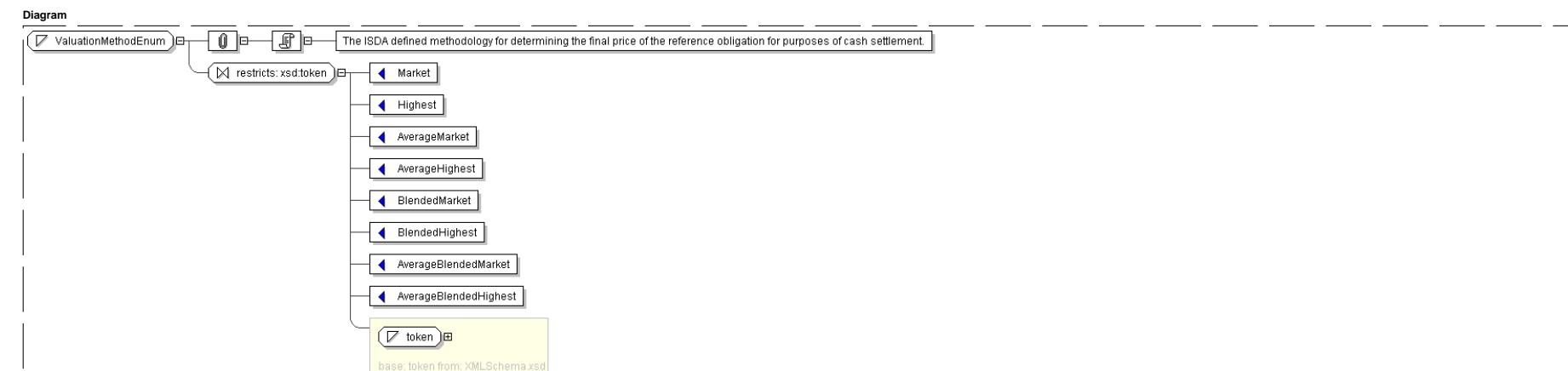
```

top

**Simple Type: ValuationMethodEnum**

Super-types:	<code>xsd:token &lt; ValuationMethodEnum</code> (by restriction)
Sub-types:	None

Name	ValuationMethodEnum
Content	<ul style="list-style-type: none"> <li>Base XSD Type: token</li> <li><code>value</code> comes from list: ('Market' 'Highest' 'AverageMarket' 'AverageHighest' 'BlendedMarket' 'BlendedHighest')</li> </ul>
Documentation	

**Schema Component Representation**

```

<xsd:simpleType name="ValuationMethodEnum">
    <xsd:restriction base=" xsd:token ">
        <xsd:enumeration value="Market"/>
        <xsd:enumeration value="Highest"/>
        <xsd:enumeration value="AverageMarket"/>
        <xsd:enumeration value="AverageHighest"/>
        <xsd:enumeration value="BlendedMarket"/>
        <xsd:enumeration value="BlendedHighest"/>
        <xsd:enumeration value="AverageBlendedMarket"/>
        <xsd:enumeration value="AverageBlendedHighest"/>
    </xsd:restriction>
</xsd:simpleType>

```

top

**Simple Type: WeeklyRollConventionEnum**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None
<b>Name</b>	WeeklyRollConventionEnum
<b>Content</b>	<ul style="list-style-type: none"> <li>Union of following types:           <ul style="list-style-type: none"> <li><a href="#">DayOfWeekEnum</a></li> <li>Locally defined type:               <ul style="list-style-type: none"> <li>Base XSD Type: token</li> <li>value comes from list: ('TBILL')</li> </ul> </li> </ul> </li> </ul>

**Documentation** The specification of a weekly roll day.

**Diagram****Schema Component Representation**

```

<xsd:simpleType name="WeeklyRollConventionEnum">
  <xsd:union memberTypes=" DayOfWeekEnum ">
    <xsd:simpleType>
      <xsd:restriction base=" xsd:token ">
        <xsd:enumeration value="TBILL"/>
      </xsd:restriction>
    </xsd:simpleType>
  </xsd:union>
</xsd:simpleType>
  
```

top

**Legend**

**Complex Type:** AusAddress  
Schema Component Type

**AusAddress**  
Schema Component Name

<b>Super-types:</b>	Address < AusAddress (by extension)
<b>Sub-types:</b>	<ul style="list-style-type: none"> <li><a href="#">OLDAddress</a> (by restriction)</li> </ul>

If this schema component is a type definition, its type hierarchy is shown in a gray-bordered box.

<b>Name</b>	AusAddress
<b>Abstract</b>	no

The table above displays the properties of this schema component.

**XML Instance Representation**

```

<... country="Australia" >
<unitNo> string </unitNo> [0..1]
<houseNo> string </houseNo> [1]
<street> string </street> [1]
Start Choice [1]
<city> string </city> [1]
<town> string </town> [1]
End Choice
<state> AusStates </state> [1]
<postcode> string <pattern = "[1-9][0-9]{3}>> </postcode> [1]
</...>
  
```

The XML Instance Representation table above shows the schema component's content as an XML instance.

- The minimum and maximum occurrence of elements and attributes are provided in square brackets, e.g. [0..1].
- Model group information are shown in gray, e.g. Start Choice ... End Choice.
- For type derivations, the elements and attributes that have been added to or changed from the base type's content are shown in bold.
- If an element/attribute has a fixed value, the fixed value is shown in green, e.g. country="Australia".
- Otherwise, the type of the element/attribute is displayed.
  - If the element/attribute's type is in the schema, a link is provided to it.
  - For local simple type definitions, the constraints are displayed in angle brackets, e.g. <<pattern = "[1-9][0-9]{3}>>.

**Schema Component Representation**

```

<complexType name="AusAddress">
  <complexContent>
    <extension base=" Address ">
      <sequence>
        <element name="state" type=" AusStates " />
      </sequence>
    </extension>
  </complexContent>
</complexType>
  
```

```

<element name="postcode">
<simpleType>
<restriction base=" string ">
<pattern value="[1-9][0-9]{3}" />
</restriction>
</simpleType>
</element>
</sequence>
<attribute name="country" type=" string " fixed="Australia" />
</extension>
</complexContent>
</complexType>

```

The Schema Component Representation table above displays the underlying XML representation of the schema component. (Annotations are not shown.)

top

## Glossary

**Abstract** (Applies to complex type definitions and element declarations). An abstract element or complex type cannot be used to validate an element instance. If there is a reference to an abstract element, only element declarations that can substitute the abstract element can be used to validate the instance. For references to abstract type definitions, only derived types can be used.

**All Model Group** Child elements can be provided *in any order* in instances. See: <http://www.w3.org/TR/xmlschema-1/#element-all>.

**Choice Model Group** Only one from the list of child elements and model groups can be provided in instances. See: <http://www.w3.org/TR/xmlschema-1/#element-choice>.

**Collapse Whitespace Policy** Replace tab, line feed, and carriage return characters with space character (Unicode character 32). Then, collapse contiguous sequences of space characters into single space character, and remove leading and trailing space characters.

**Disallowed Substitutions** (Applies to element declarations). If *substitution* is specified, then *substitution group* members cannot be used in place of the given element declaration to validate element instances. If *derivation methods*, e.g. extension, restriction, are specified, then the given element declaration will not validate element instances that have types derived from the element declaration's type using the specified derivation methods. Normally, element instances can override their declaration's type by specifying an *xsi:type* attribute.

**Key Constraint** Like [Uniqueness Constraint](#), but additionally requires that the specified value(s) must be provided. See: [http://www.w3.org/TR/xmlschema-1/#cIdentity\\_constraint\\_Definitions](http://www.w3.org/TR/xmlschema-1/#cIdentity_constraint_Definitions).

**Key Reference Constraint** Ensures that the specified value(s) must match value(s) from a [Key Constraint](#) or [Uniqueness Constraint](#). See: [http://www.w3.org/TR/xmlschema-1/#cIdentity\\_constraint\\_Definitions](http://www.w3.org/TR/xmlschema-1/#cIdentity_constraint_Definitions).

**Model Group** Groups together element content, specifying the order in which the element content can occur and the number of times the group of element content may be repeated. See: [http://www.w3.org/TR/xmlschema-1/#Model\\_Groups](http://www.w3.org/TR/xmlschema-1/#Model_Groups).

**Nillable** (Applies to element declarations). If an element declaration is nillable, instances can use the *xsi:nil* attribute. The *xsi:nil* attribute is the boolean attribute, *nil* from the <http://www.w3.org/2001/XMLSchema-instance> namespace. If an element instance has an *xsi:nil* attribute set to true, it can be left empty, even though its element declaration may have required content.

**Notation** A notation is used to identify the format of a piece of data. Values of elements and attributes that are of type, NOTATION, must come from the names of declared notations. See: [http://www.w3.org/TR/xmlschema-1/#cNotation\\_Declarations](http://www.w3.org/TR/xmlschema-1/#cNotation_Declarations).

**Preserve Whitespace Policy** Preserve whitespaces exactly as they appear in instances.

**Prohibited Derivations** (Applies to type definitions). Derivation methods that cannot be used to create sub-types from a given type definition.

**Prohibited Substitutions** (Applies to complex type definitions). Prevents sub-types that have been derived using the specified derivation methods from validating element instances in place of the given type definition.

**Replace Whitespace Policy** Replace tab, line feed, and carriage return characters with space character (Unicode character 32).

**Sequence Model Group** Child elements and model groups must be provided *in the specified order* in instances. See: <http://www.w3.org/TR/xmlschema-1/#element-sequence>.

**Substitution Group** Elements that are *members* of a substitution group can be used wherever the *head* element of the substitution group is referenced.

**Substitution Group Exclusions** (Applies to element declarations). Prohibits element declarations from nominating themselves as being able to substitute a given element declaration, if they have types that are derived from the original element's type using the specified derivation methods.

**Target Namespace** The target namespace identifies the namespace that components in this schema belongs to. If no target namespace is provided, then the schema components do not belong to any namespace.

**Uniqueness Constraint** Ensures uniqueness of an element/attribute value, or a combination of values, within a specified scope. See: [http://www.w3.org/TR/xmlschema-1/#cIdentity\\_constraint\\_Definitions](http://www.w3.org/TR/xmlschema-1/#cIdentity_constraint_Definitions).

top

# XML Schema Documentation

## Table of Contents

- [Schema Document Properties](#)
- [Global Declarations](#)
  - [Element: brokerEquityOption](#)
  - [Element: equityForward](#)
  - [Element: equityOption](#)
  - [Element: equityOptionTransactionSupplement](#)
- [Global Definitions](#)
  - [Complex Type: BrokerEquityOption](#)
  - [Complex Type: EquityAmericanExercise](#)
  - [Complex Type: EquityBermudaExercise](#)
  - [Complex Type: EquityDerivativeBase](#)
  - [Complex Type: EquityDerivativeLongFormBase](#)
  - [Complex Type: EquityDerivativeShortFormBase](#)
  - [Complex Type: EquityEuropeanExercise](#)
  - [Complex Type: EquityExerciseValuationSettlement](#)
  - [Complex Type: EquityForward](#)
  - [Complex Type: EquityMultipleExercise](#)
  - [Complex Type: EquityOption](#)
  - [Complex Type: EquityOptionTermination](#)
  - [Complex Type: EquityOptionTransactionSupplement](#)
  - [Complex Type: PrePayment](#)
  - [Model Group: EquityExpiration.model](#)
- [Legend](#)
- [Glossary](#)

top

## Schema Document Properties

<b>Target Namespace</b>	<a href="http://www.fpml.org/FpML-5/confirmation">http://www.fpml.org/FpML-5/confirmation</a>
<b>Version</b>	\$Revision: 2864 \$
<b>Element and Attribute Namespaces</b>	<ul style="list-style-type: none"> <li>• Global element and attribute declarations belong to this schema's target namespace.</li> <li>• By default, local element declarations belong to this schema's target namespace.</li> <li>• By default, local attribute declarations have no namespace.</li> </ul>
<b>Schema Composition</b>	<ul style="list-style-type: none"> <li>• This schema includes components from the following schema document(s):           <ul style="list-style-type: none"> <li>◦ <a href="#">fpml-eq-shared-5-0.xsd</a></li> </ul> </li> </ul>

## Declared Namespaces

Prefix	Namespace
Default namespace	<a href="http://www.fpml.org/FpML-5/confirmation">http://www.fpml.org/FpML-5/confirmation</a>
xml	<a href="http://www.w3.org/XML/1998/namespace">http://www.w3.org/XML/1998/namespace</a>
fpml-annotation	<a href="http://www.fpml.org/annotation">http://www.fpml.org/annotation</a>
xsd	<a href="http://www.w3.org/2001/XMLSchema">http://www.w3.org/2001/XMLSchema</a>
ecore	<a href="http://www.eclipse.org/emf/2002/Ecore">http://www.eclipse.org/emf/2002/Ecore</a>

## Schema Component Representation

```

<xsd:schema targetNamespace="http://www.fpml.org/FpML-5/confirmation"
  documentRoot="FpML" nsPrefix="conf" package="org.fpml.confirmation" version="$Revision: 2864 "
  $" elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xsd:include schemaLocation="fpml-eq-shared-5-0.xsd"/>
  ...
</xsd:schema>

```

top

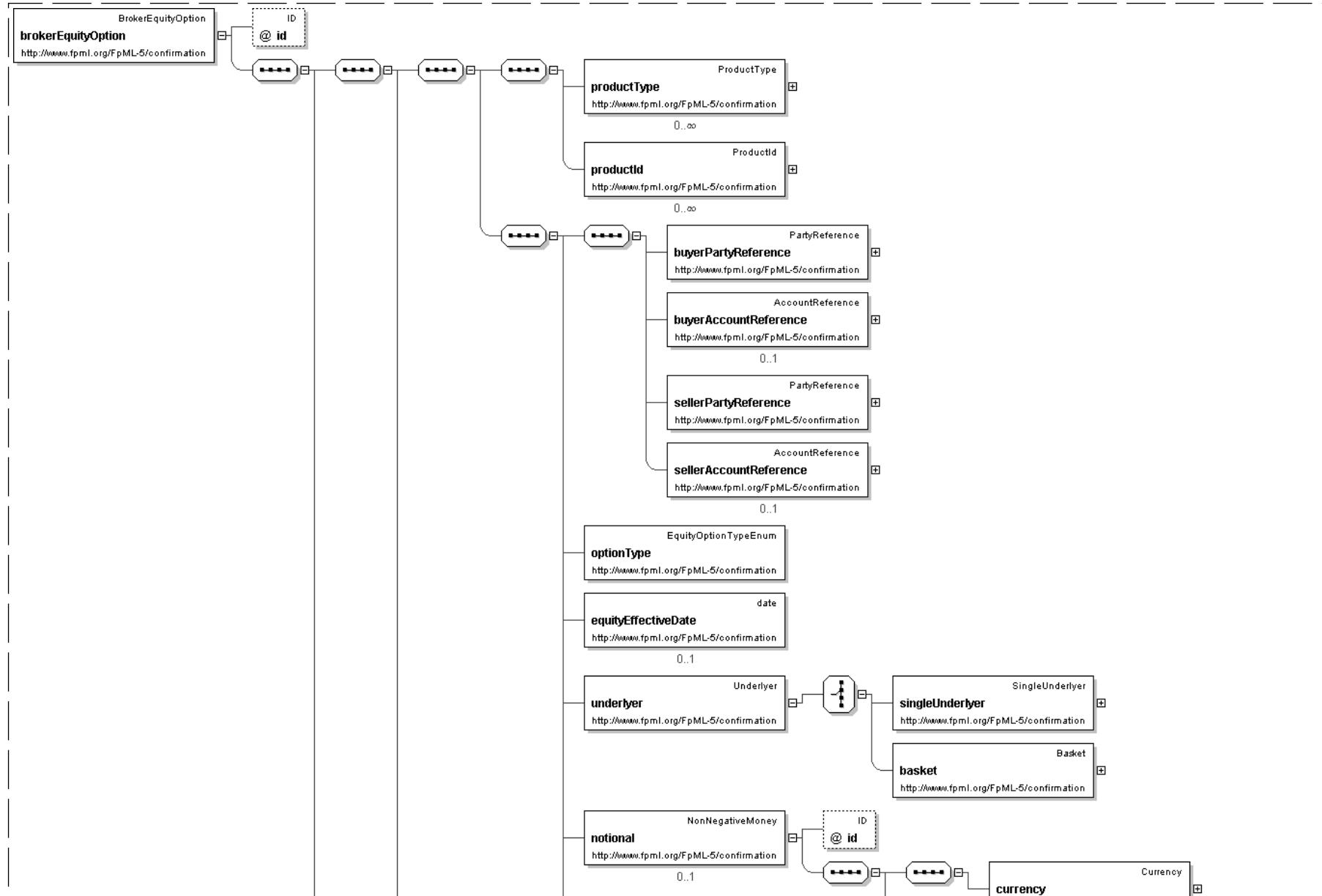
## Global Declarations

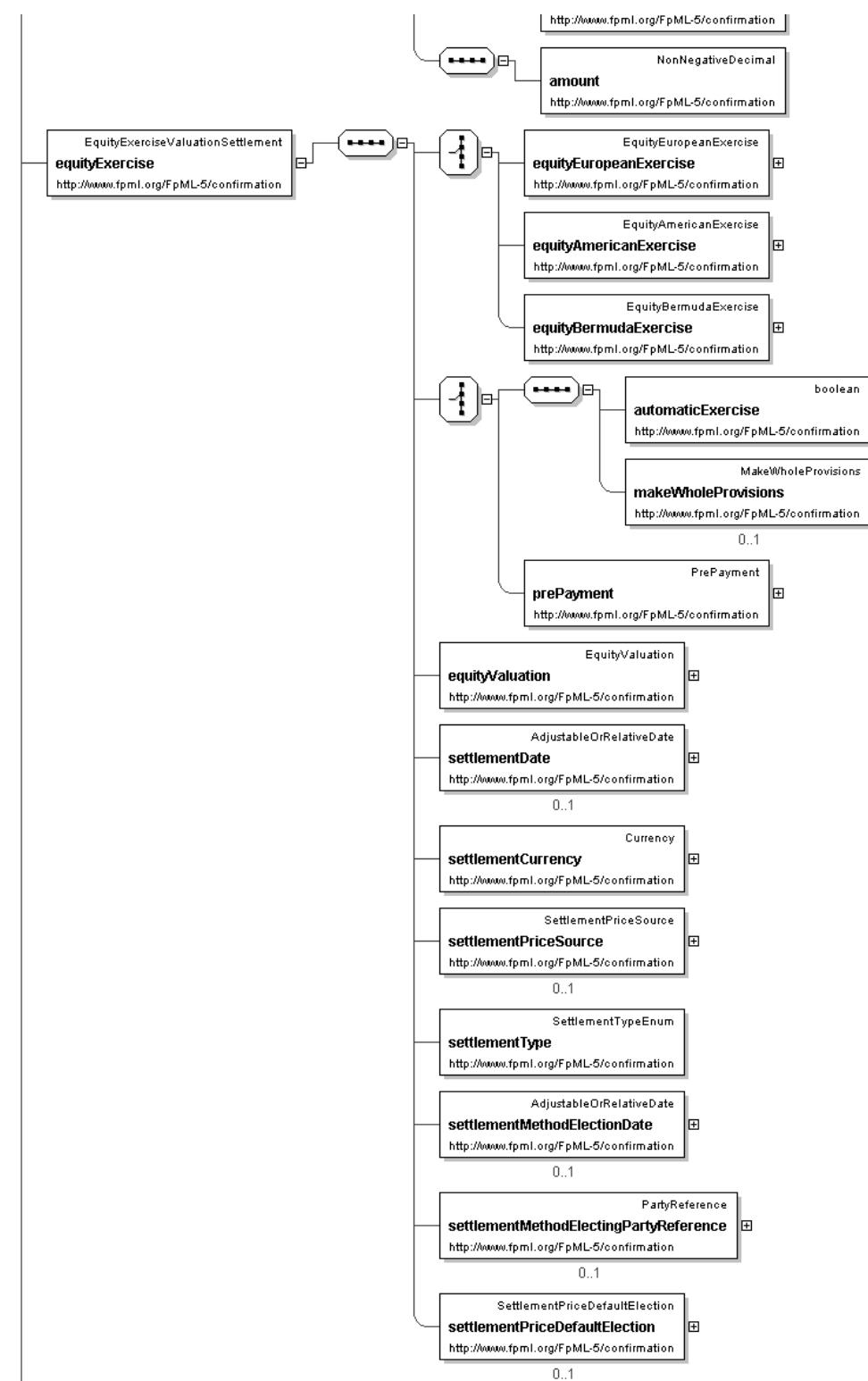
**Element: brokerEquityOption**

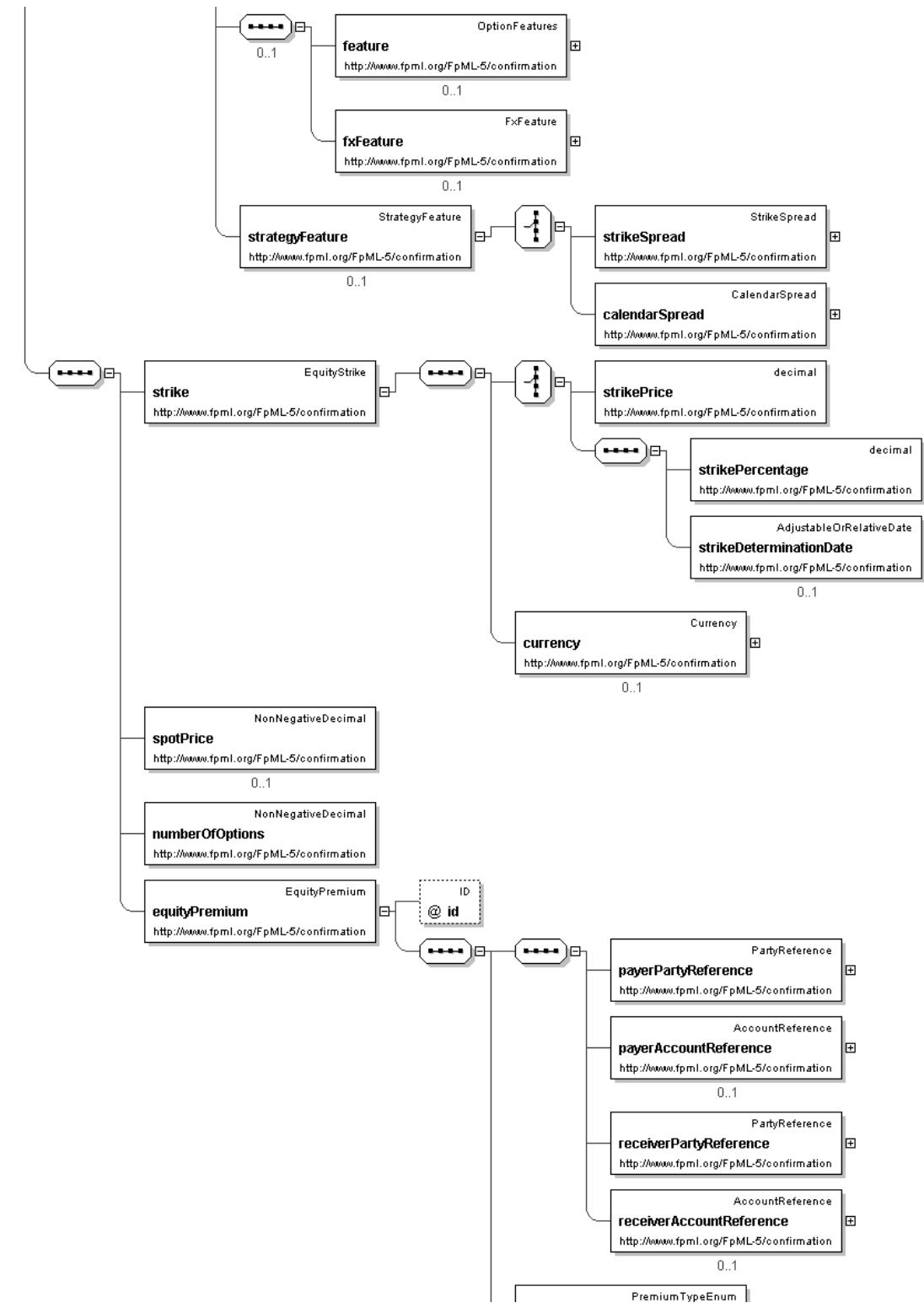
- This element can be used wherever the following element is referenced:

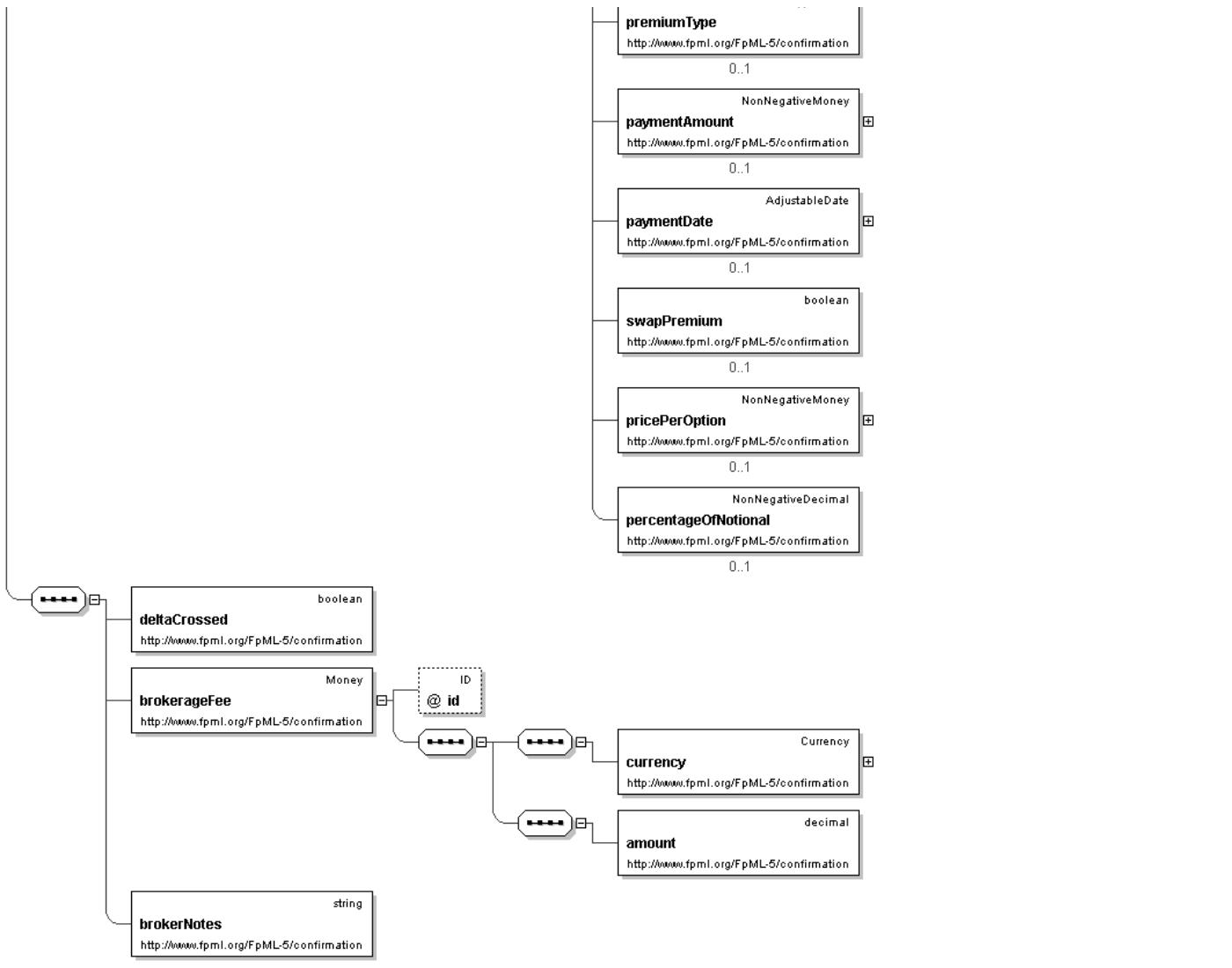
→ [product](#)

<b>Name</b>	brokerEquityOption
<b>Type</b>	<a href="#">BrokerEquityOption</a>
<b>Nillable</b>	no
<b>Abstract</b>	no
<b>Documentation</b>	A component describing a Broker View of an Equity Option.

**Logical Diagram**







'A reference to the account that buys this instrument.'

<sellerPartyReference> [PartyReference](#) </sellerPartyReference> [1]

'A reference to the party that sells ("writes") this instrument, i.e. that grants the rights defined by this instrument and in return receives a payment for it. See 2000 ISDA definitions Article 11.1 (a). In the case of FRAs this is the floating rate payer.'

<sellerAccountReference> [AccountReference](#) </sellerAccountReference> [0..1]

'A reference to the account that sells this instrument.'

<optionType> [EquityOptionTypeEnum](#) </optionType> [1]

'The type of option transaction.'

<equityEffectiveDate> [xsd:date](#) </equityEffectiveDate> [0..1]

'Effective date for a forward starting option.'

<underlyer> [Underlyer](#) </underlyer> [1]

'Specifies the underlying component, which can be either one or many and consists in either equity, index or convertible bond component, or a combination of these.'

<notional> [NonNegativeMoney](#) </notional> [0..1]

'The notional amount.'

<equityExercise> [EquityExerciseValuationSettlement](#) </equityExercise> [1]

'The parameters for defining how the equity option can be exercised, how it is valued and how it is settled.'

Start Group: [Feature.model](#) [0..1]

<feature> [OptionFeatures](#) </feature> [0..1]

'Asian, Barrier, Knock and Pass Through features.'

<fxFeature> [FxFeature](#) </fxFeature> [0..1]

'Quanto, Composite, or Cross Currency FX features.'

End Group: [Feature.model](#)

<strategyFeature> [StrategyFeature](#) </strategyFeature> [0..1]

'A equity option simple strategy feature.'

<strike> [EquityStrike](#) </strike> [1]

'Defines whether it is a price or level at which the option has been, or will be, struck.'

<spotPrice> [NonNegativeDecimal](#) </spotPrice> [0..1]

'The price per share, index or basket observed on the trade or effective date.'

<numberOfOptions> [NonNegativeDecimal](#) </numberOfOptions> [1]

'The number of options comprised in the option transaction.'

<equityPremium> [EquityPremium](#) </equityPremium> [1]

'The equity option premium payable by the buyer to the seller.'

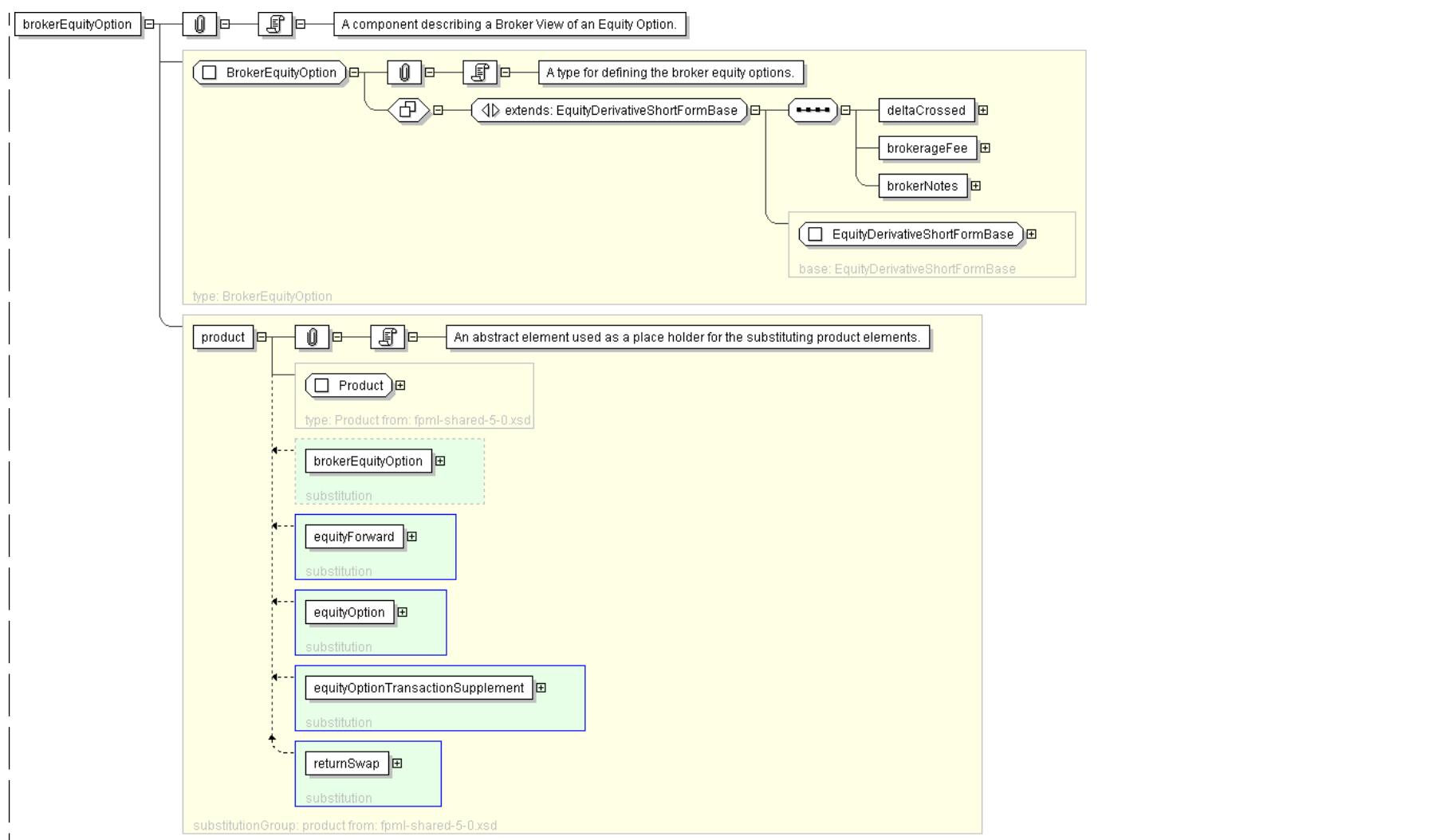
<deltaCrossed> [xsd:boolean](#) </deltaCrossed> [1]

<brokerageFee> [Money](#) </brokerageFee> [1]

<brokerNotes> [xsd:string](#) </brokerNotes> [1]

</brokerEquityOption>

**Diagram**

**Schema Component Representation**

```
<xsd:element name="brokerEquityOption" type="BrokerEquityOption" substitutionGroup="product"/>
```

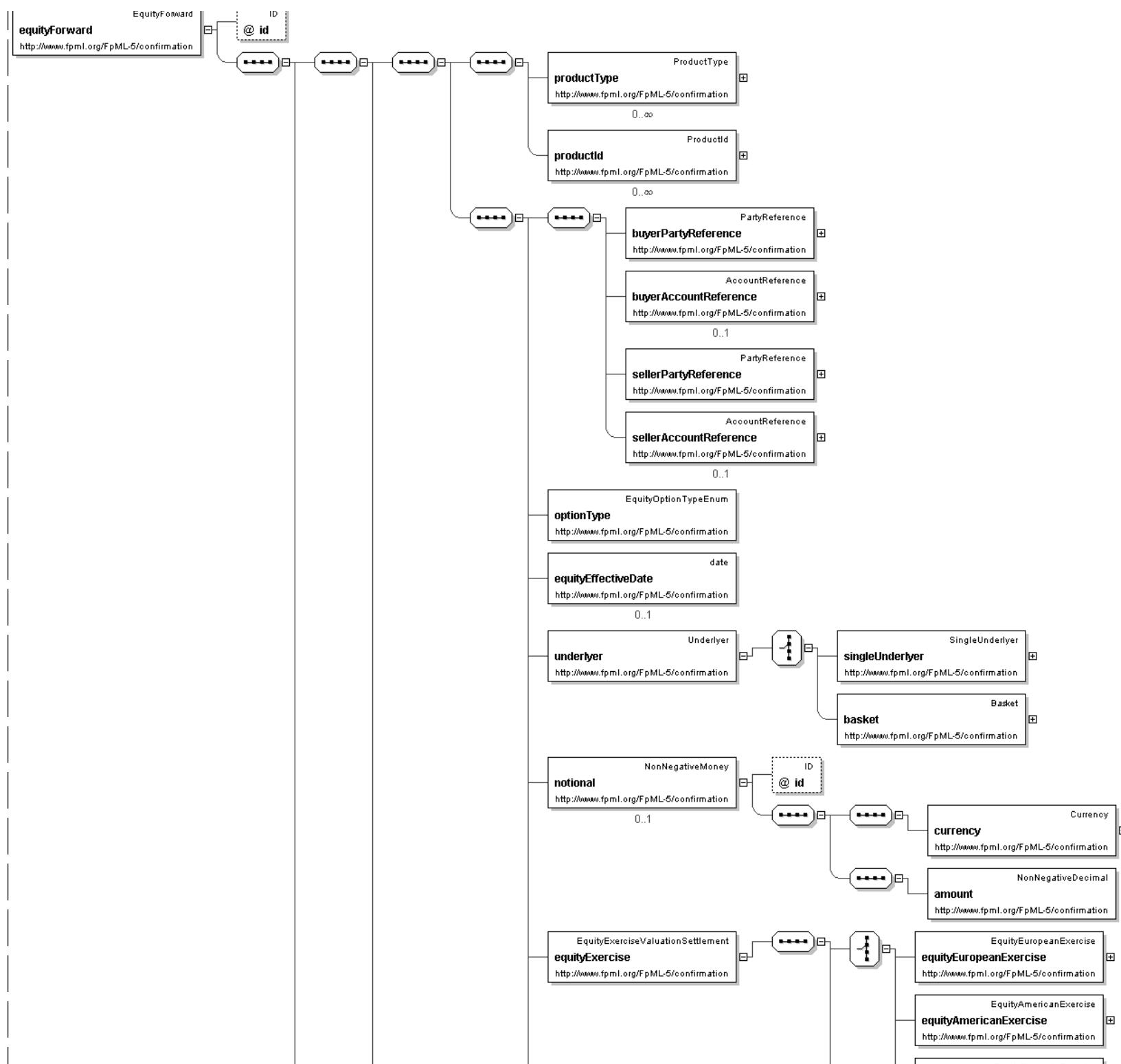
top

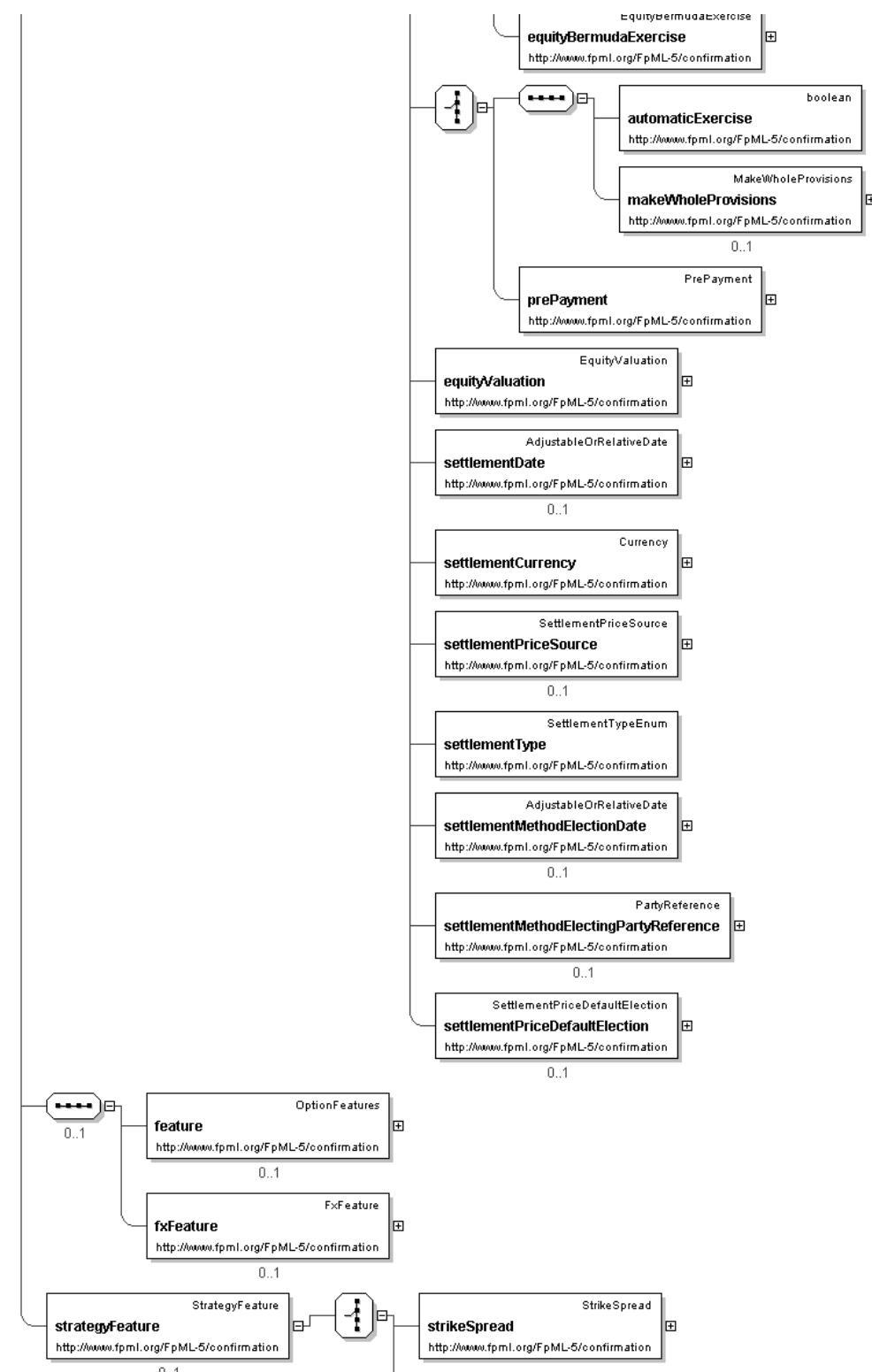
**Element: equityForward**

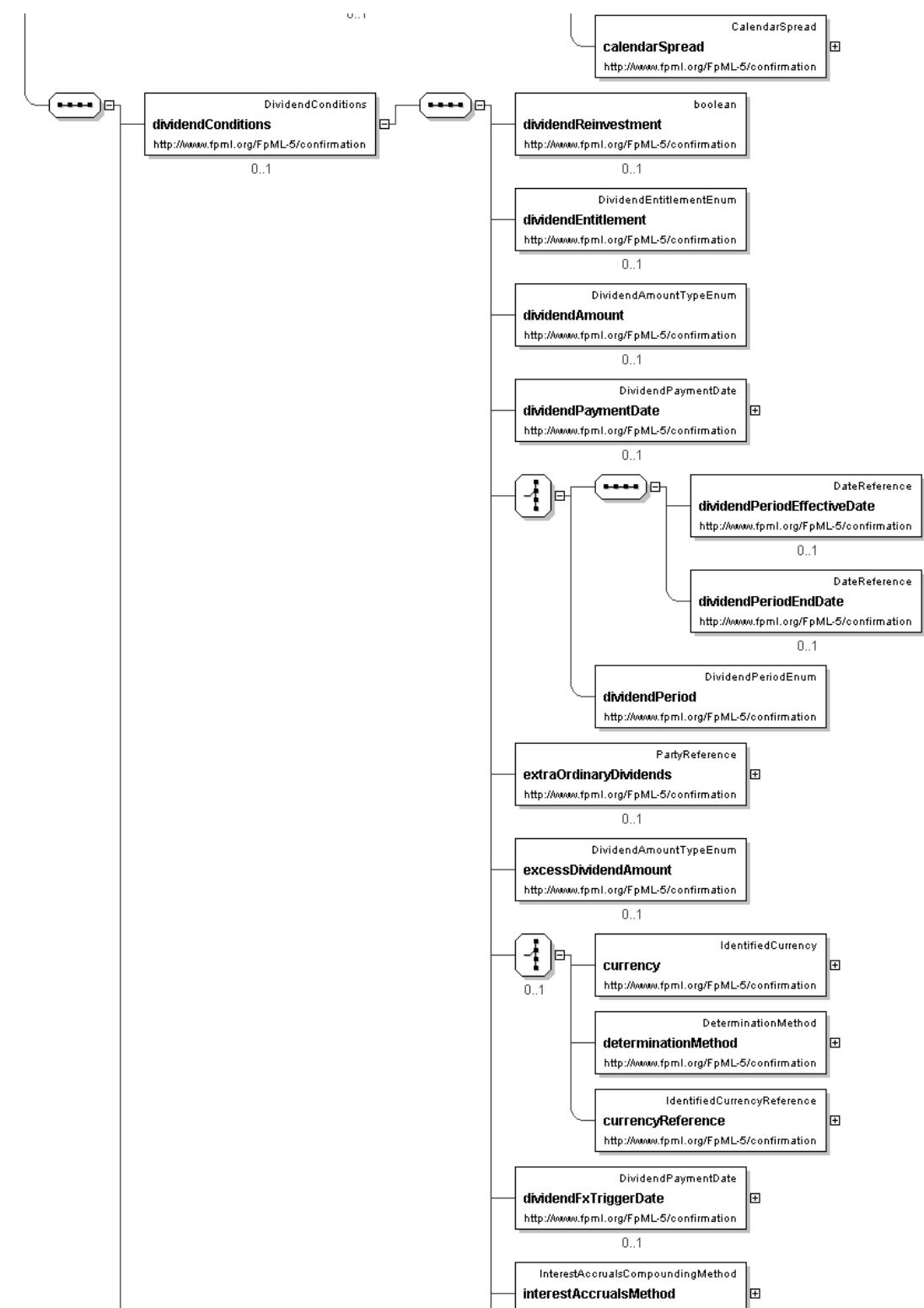
- This element can be used wherever the following element is referenced:
  - product

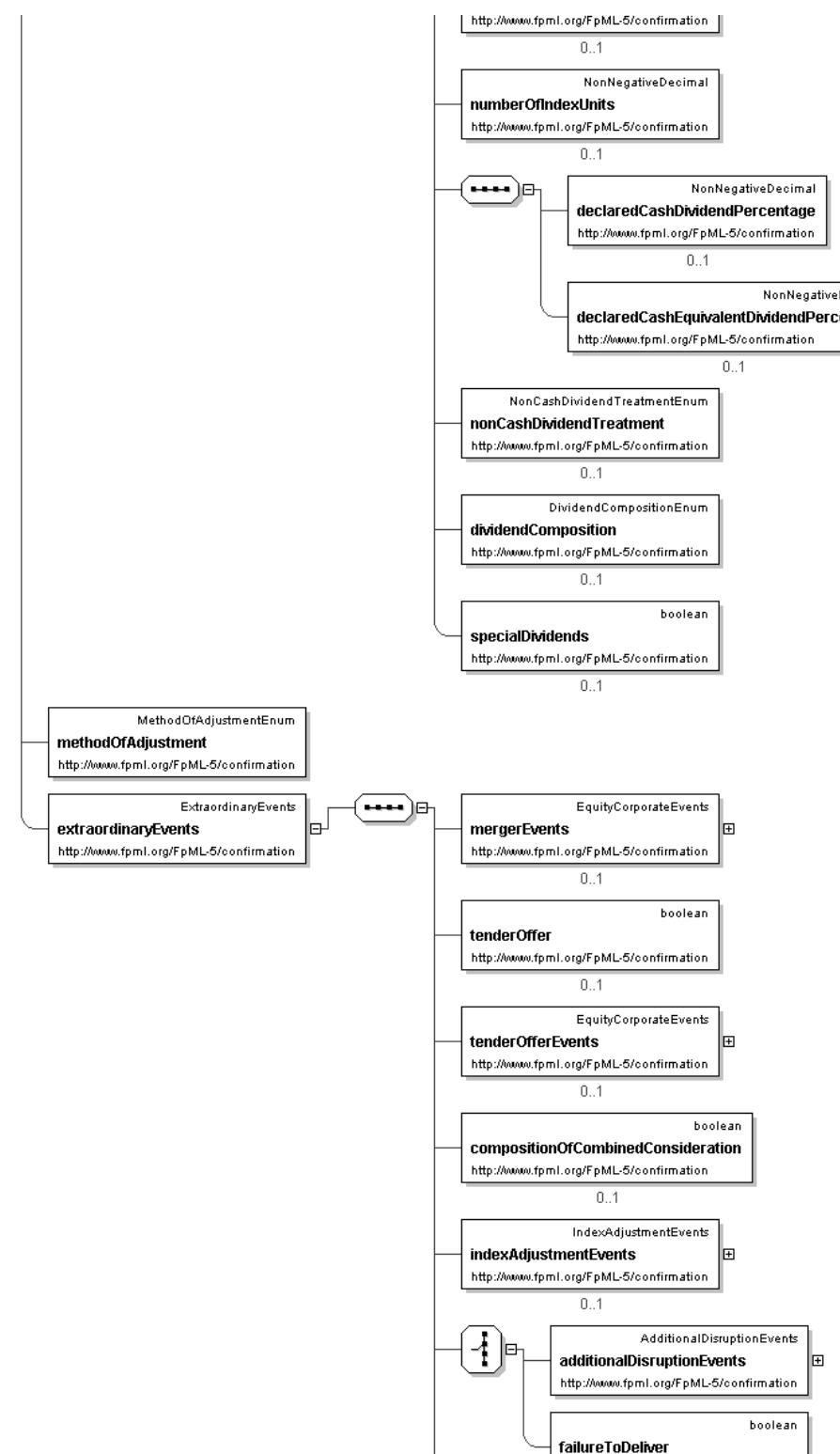
Name	equityForward
Type	<a href="#">EquityForward</a>
Nillable	no
Abstract	no
Documentation	A component describing an Equity Forward product.

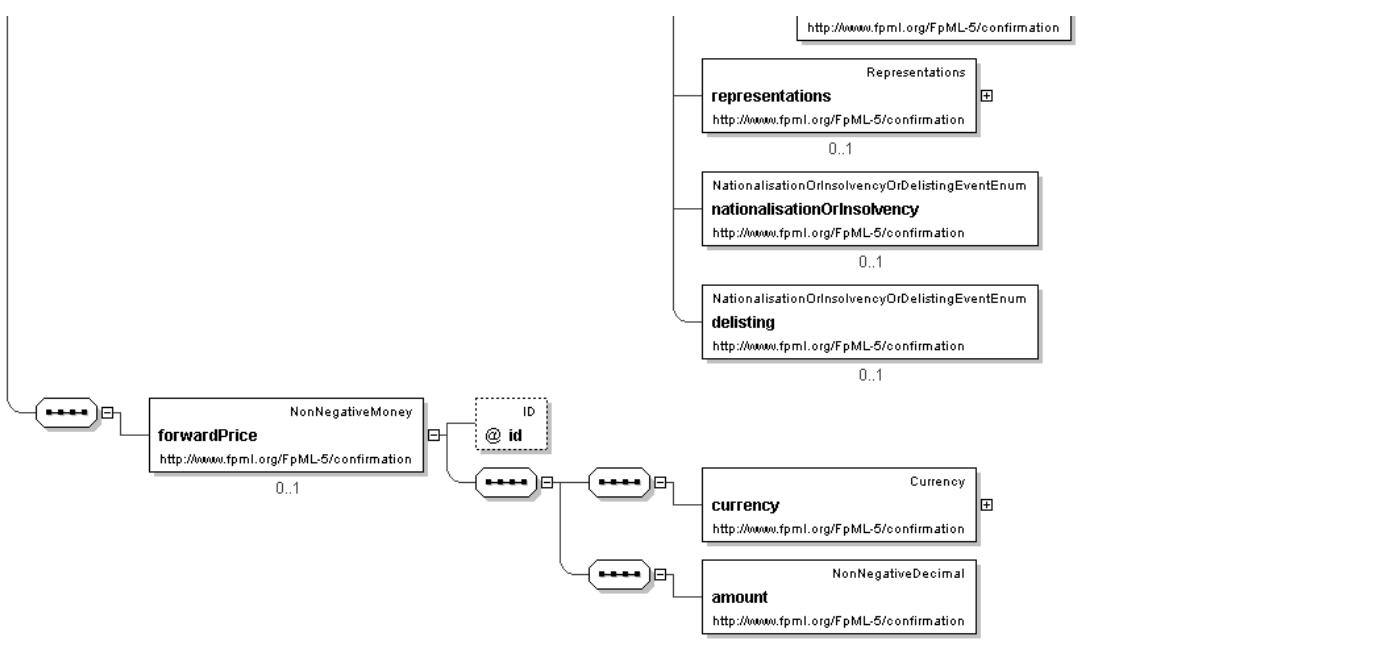
**Logical Diagram**









**XML Instance Representation**

```

<equityForward
  id=" xsd:ID [0..1]">
  <productType> ProductType </productType> [0..*]
  'A classification of the type of product. FpML defines a simple product categorization using
  a coding scheme.'
  <productId> ProductId </productId> [0..*]
  'A product 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.'
  <buyerPartyReference> PartyReference </buyerPartyReference> [1]
  'A reference to the party that buys this instrument, i.e. pays for this instrument and
  receives the rights defined by it. See 2000 ISDA definitions Article 11.1 (b). In the case
  of FRAs this is the fixed rate payer.'
  <buyerAccountReference> AccountReference </buyerAccountReference> [0..1]
  'A reference to the account that buys this instrument.'
  <sellerPartyReference> PartyReference </sellerPartyReference> [1]
  'A reference to the party that sells ("writes") this instrument, i.e. that grants the
  rights defined by this instrument and in return receives a payment for it. See 2000
  ISDA definitions Article 11.1 (a). In the case of FRAs this is the floating rate payer.'
  <sellerAccountReference> AccountReference </sellerAccountReference> [0..1]
  'A reference to the account that sells this instrument.'
  <optionType> EquityOptionTypeEnum </optionType> [1]
  'The type of option transaction.'
  <equityEffectiveDate> xsd:date </equityEffectiveDate> [0..1]
  'Effective date for a forward starting option.'
  <underlyer> Underlyer </underlyer> [1]
  'Specifies the underlying component, which can be either one or many and consists in
  either equity, index or convertible bond component, or a combination of these.'

```

```

<notional> NonNegativeMoney </notional> [0..1]
'The notional amount.'

<equityExercise> EquityExerciseValuationSettlement </equityExercise> [1]
'The parameters for defining how the equity option can be exercised, how it is valued and
how it is settled.'

Start Group: Feature.model [0..1]
<feature> OptionFeatures </feature> [0..1]
'Asian, Barrier, Knock and Pass Through features.'

<fxFeature> FxFeature </fxFeature> [0..1]
'Quanto, Composite, or Cross Currency FX features.'

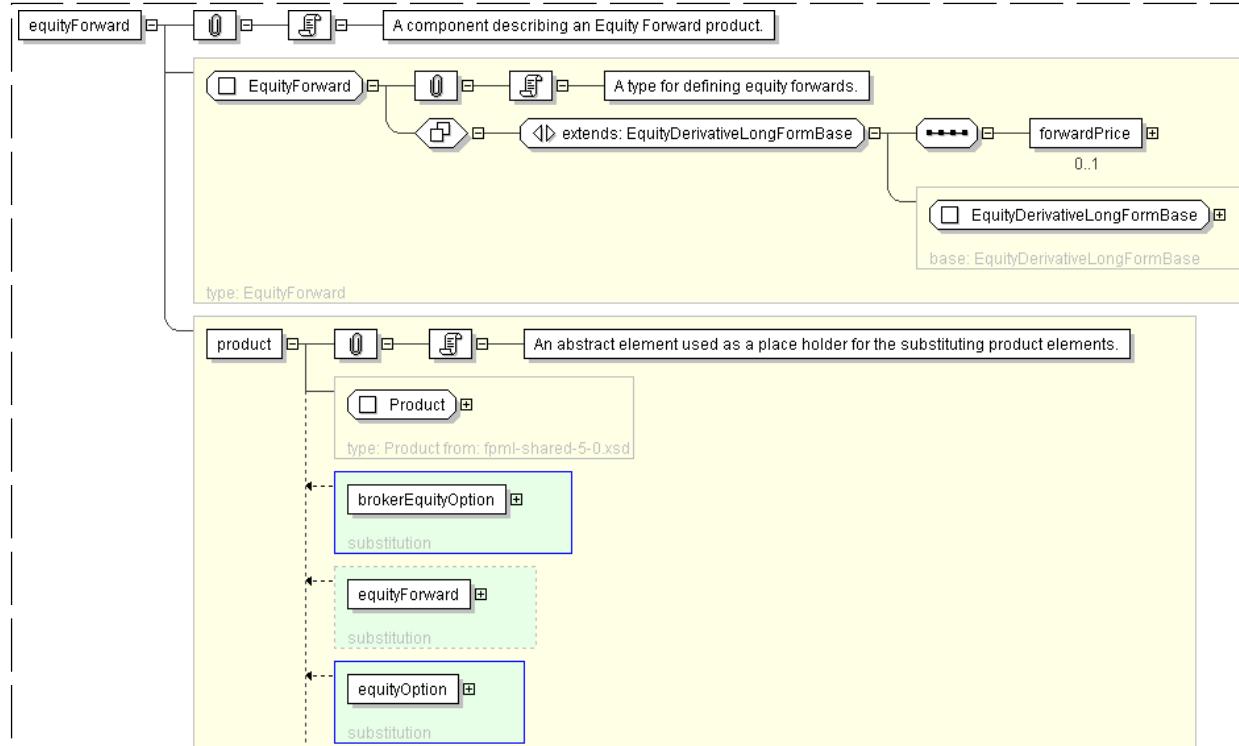
End Group: Feature.model
<strategyFeature> StrategyFeature </strategyFeature> [0..1]
'A equity option simple strategy feature.'

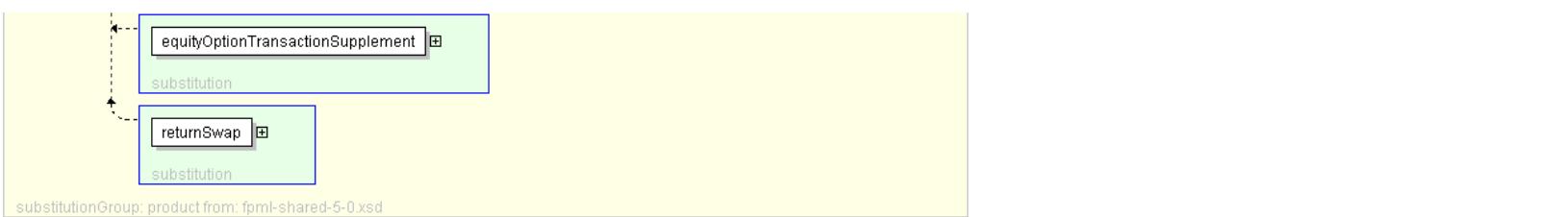
<dividendConditions> DividendConditions </dividendConditions> [0..1]
<methodOfAdjustment> MethodOfAdjustmentEnum </methodOfAdjustment> [1]
'Defines how adjustments will be made to the contract should one or more of the
extraordinary events occur.'

<extraordinaryEvents> ExtraordinaryEvents </extraordinaryEvents> [1]
'Where the underlying is shares, specifies events affecting the issuer of those shares that
may require the terms of the transaction to be adjusted.'

<forwardPrice> NonNegativeMoney </forwardPrice> [0..1]
'The forward price per share, index or basket.'
```

&lt;/equityForward&gt;

**Diagram**

**Schema Component Representation**

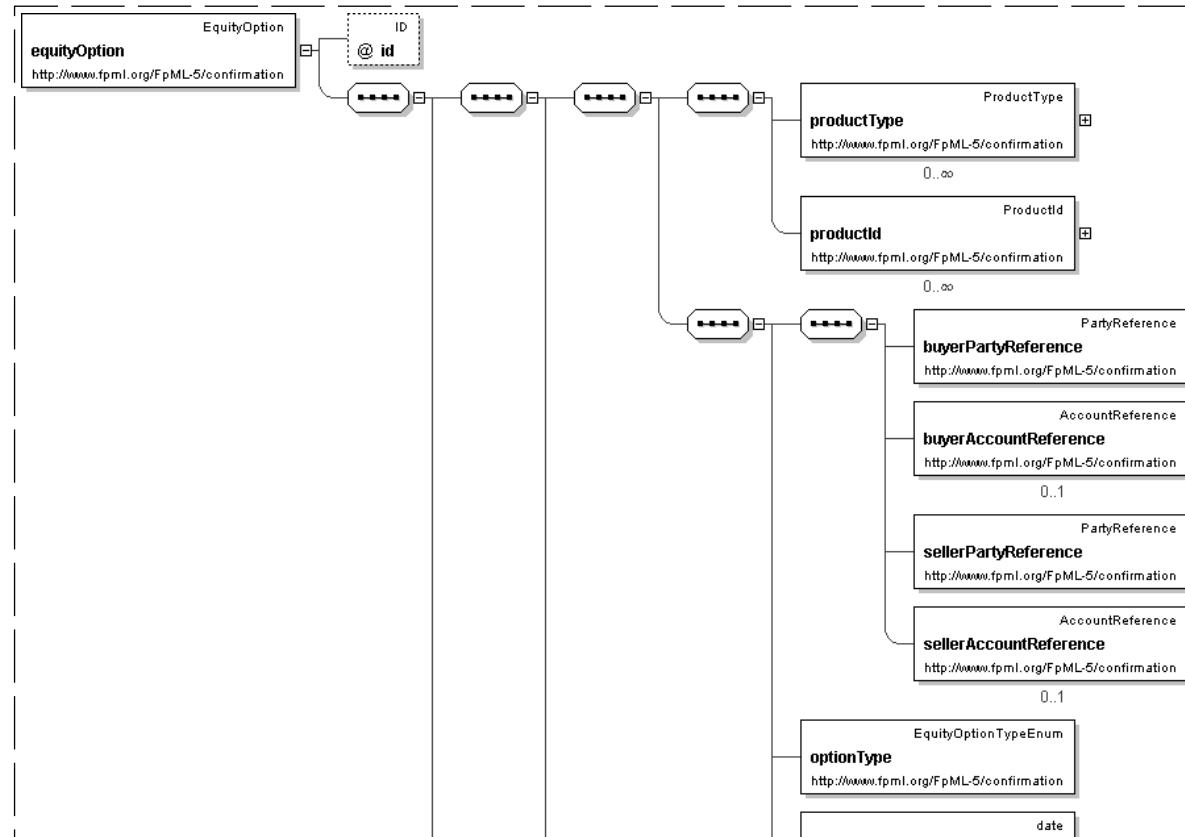
```
<xsd:element name="equityForward" type="EquityForward" substitutionGroup="product" />
```

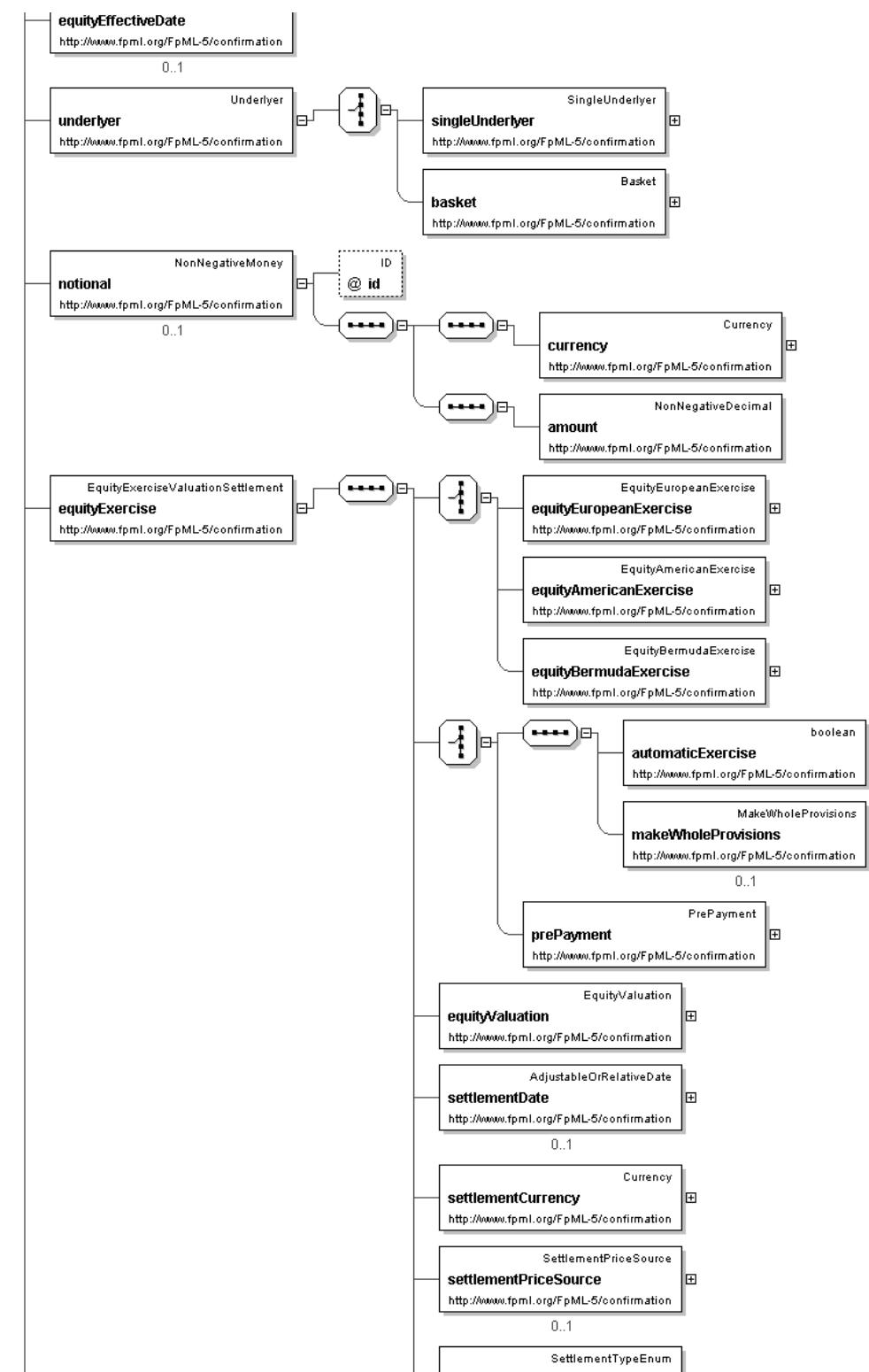
top

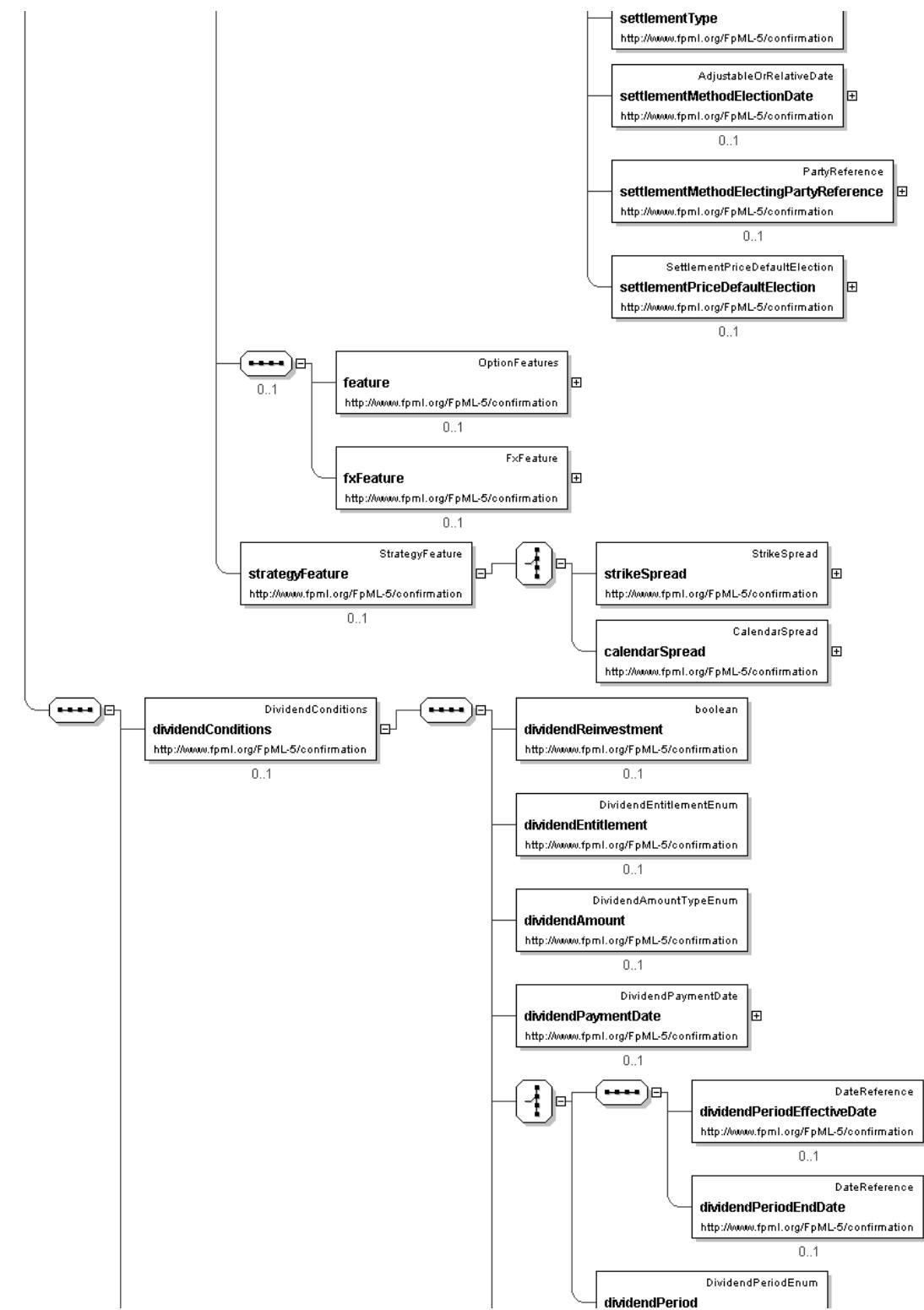
**Element: equityOption**

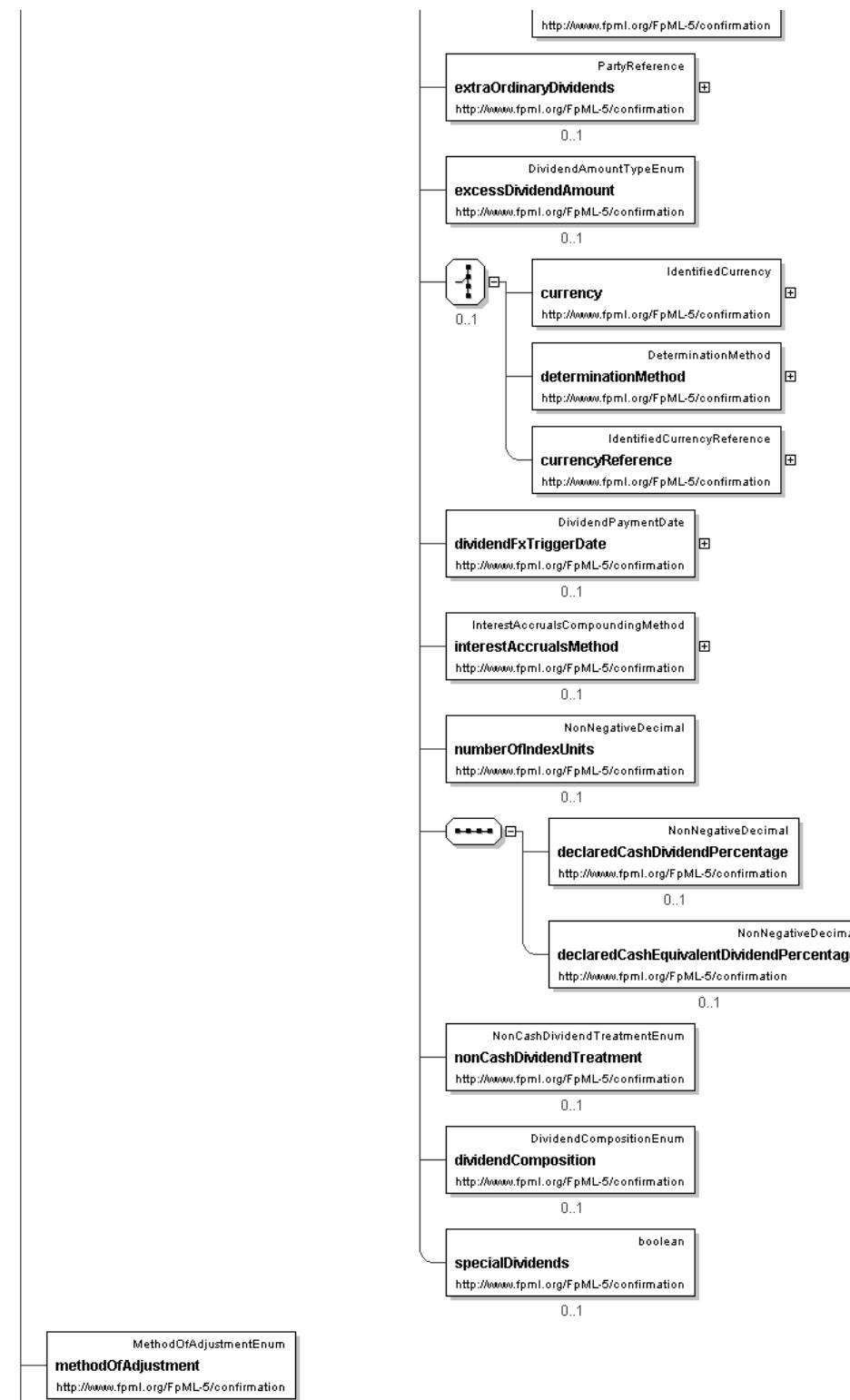
- This element can be used wherever the following element is referenced:
  - product

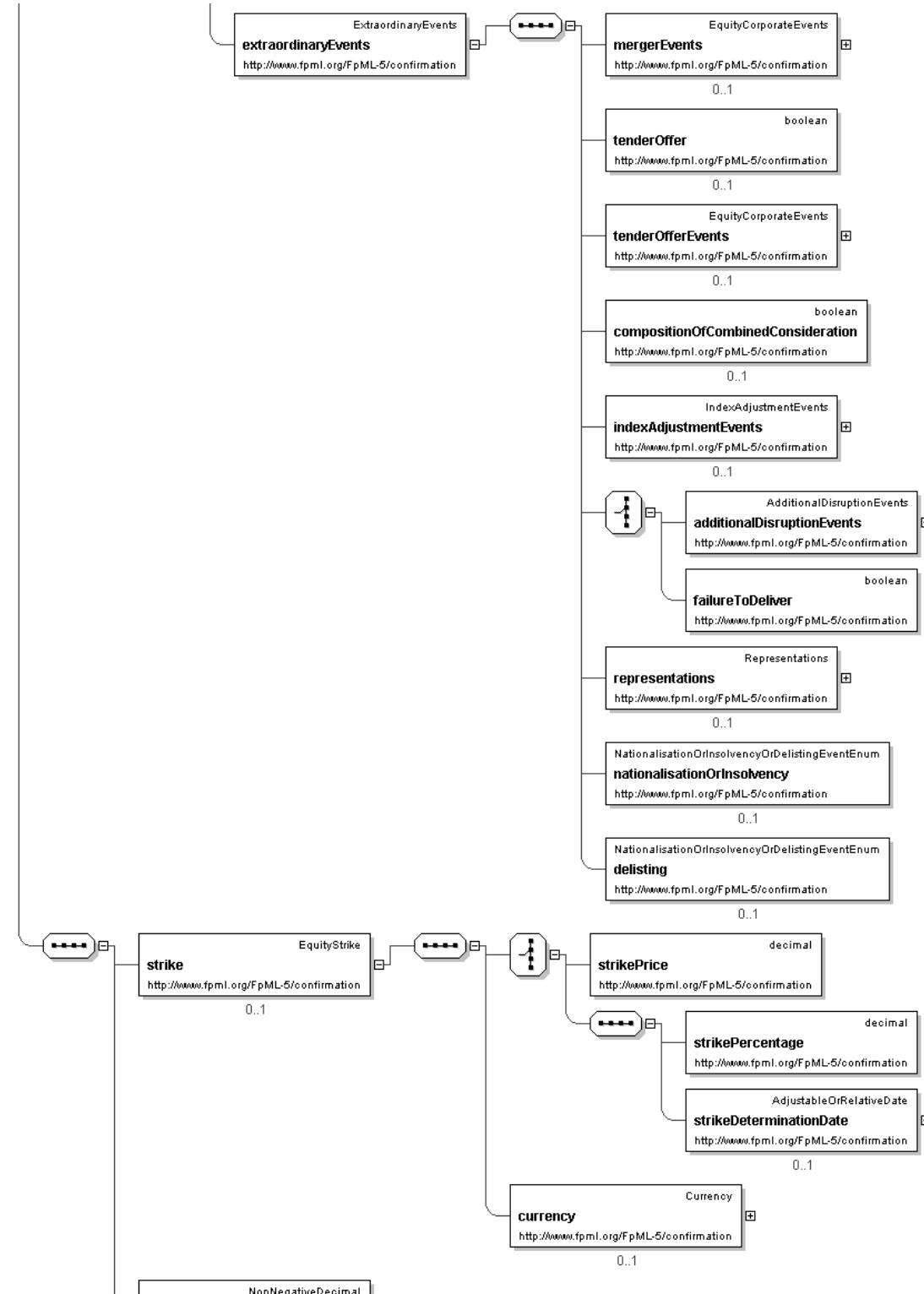
<b>Name</b>	equityOption
<b>Type</b>	EquityOption
<b>Nillable</b>	no
<b>Abstract</b>	no
<b>Documentation</b>	A component describing an Equity Option product.

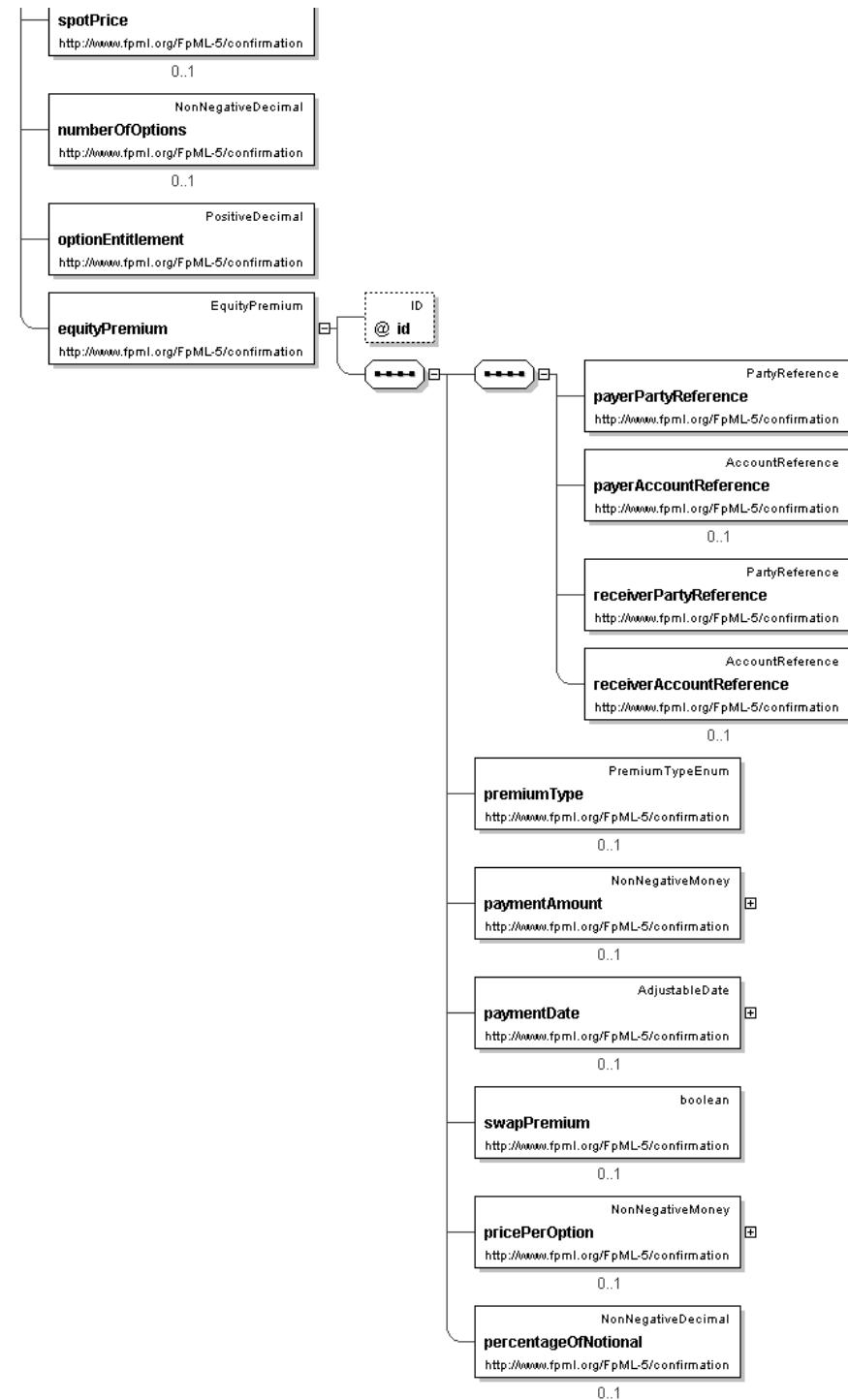
**Logical Diagram**









**XML Instance Representation**

&lt;equityOption

```

id=" xsd:ID [0..1]">
<productType> ProductType </productType> [0..*]
'A classification of the type of product. FpML defines a simple product categorization using
a coding scheme.'

<productId> ProductId </productId> [0..*]
'A product 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.'

<buyerPartyReference> PartyReference </buyerPartyReference> [1]
'A reference to the party that buys this instrument, ie. pays for this instrument and
receives the rights defined by it. See 2000 ISDA definitions Article 11.1 (b). In the case
of FRAs this is the fixed rate payer.'

<buyerAccountReference> AccountReference </buyerAccountReference> [0..1]
'A reference to the account that buys this instrument.'

<sellerPartyReference> PartyReference </sellerPartyReference> [1]
'A reference to the party that sells ("writes") this instrument, i.e. that grants the
rights defined by this instrument and in return receives a payment for it. See 2000
ISDA definitions Article 11.1 (a). In the case of FRAs this is the floating rate payer.'

<sellerAccountReference> AccountReference </sellerAccountReference> [0..1]
'A reference to the account that sells this instrument.'

<optionType> EquityOptionTypeEnum </optionType> [1]
'The type of option transaction.'

<equityEffectiveDate> xsd:date </equityEffectiveDate> [0..1]
'Effective date for a forward starting option.'

<underlyer> Underlyer </underlyer> [1]
'Specifies the underlying component, which can be either one or many and consists in
either equity, index or convertible bond component, or a combination of these.'

<notional> NonNegativeMoney </notional> [0..1]
'The notional amount.'

<equityExercise> EquityExerciseValuationSettlement </equityExercise> [1]
'The parameters for defining how the equity option can be exercised, how it is valued and
how it is settled.

Start Group: Feature.model [0..1]
<feature> OptionFeatures </feature> [0..1]
'Asian, Barrier, Knock and Pass Through features.'

<fxFeature> FxFeature </fxFeature> [0..1]
'Quanto, Composite, or Cross Currency FX features.

End Group: Feature.model
<strategyFeature> StrategyFeature </strategyFeature> [0..1]
'A equity option simple strategy feature.'

<dividendConditions> DividendConditions </dividendConditions> [0..1]
<methodOfAdjustment> MethodOfAdjustmentEnum </methodOfAdjustment> [1]
'Defines how adjustments will be made to the contract should one or more of the
extraordinary events occur.

<extraordinaryEvents> ExtraordinaryEvents </extraordinaryEvents> [1]
'Where the underlying is shares, specifies events affecting the issuer of those shares that
may require the terms of the transaction to be adjusted.

<strike> EquityStrike </strike> [0..1]
'Defines whether it is a price or level at which the option has been, or will be, struck.'

```

```

<spotPrice> NonNegativeDecimal </spotPrice> [0..1]
'The price per share, index or basket observed on the trade or effective date.'

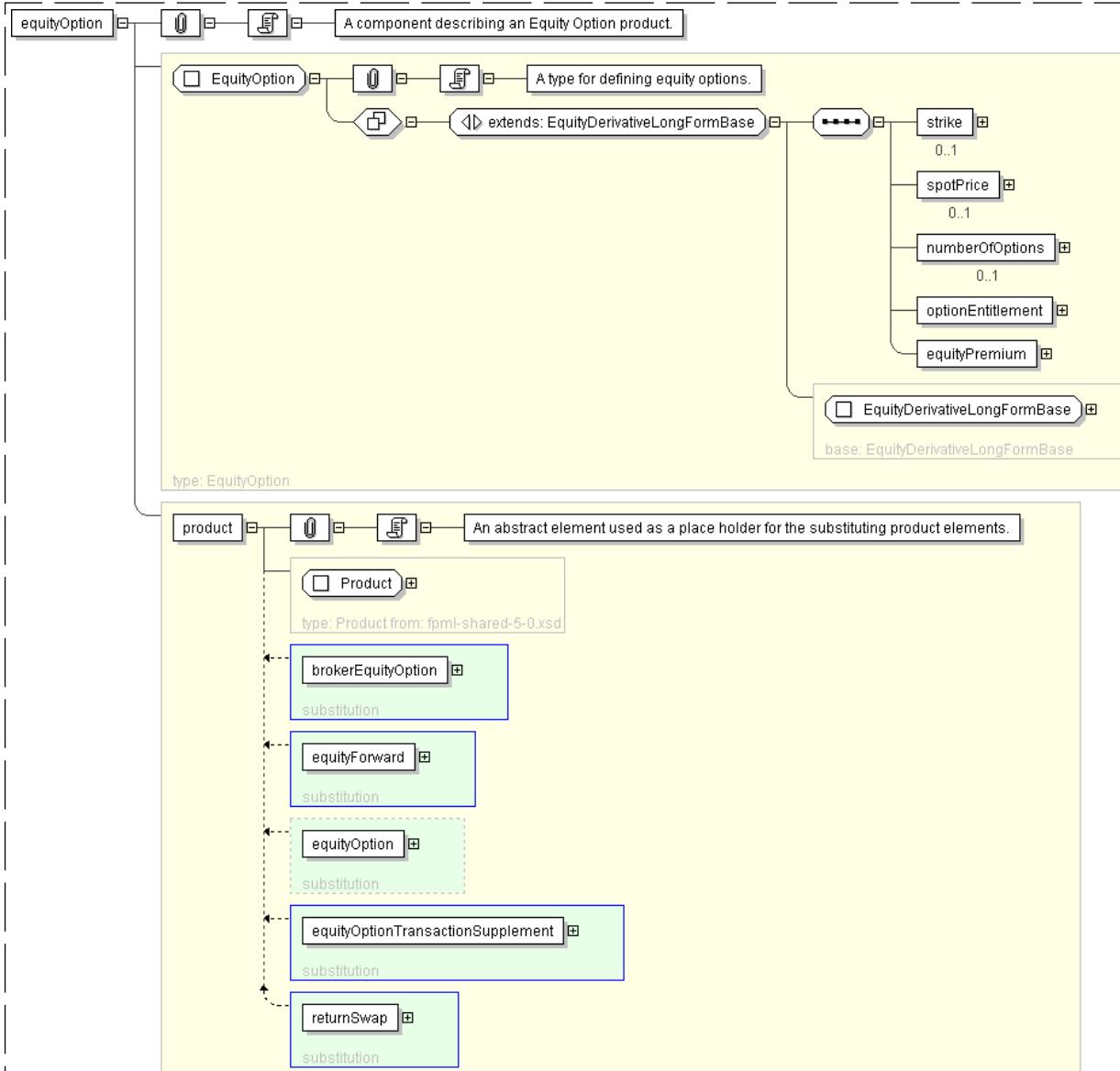
<numberOfOptions> NonNegativeDecimal </numberOfOptions> [0..1]
'The number of options comprised in the option transaction.'

<optionEntitlement> PositiveDecimal </optionEntitlement> [1]
'The number of shares per option comprised in the option transaction.'

<equityPremium> EquityPremium </equityPremium> [1]
'The equity option premium payable by the buyer to the seller.'

</equityOption>

```

**Diagram**

substitutionGroup: product from: fpml-shared-5-0.xsd

## Schema Component Representation

&lt;xsd:element name="equityOption" type="EquityOption" substitutionGroup="product"/&gt;

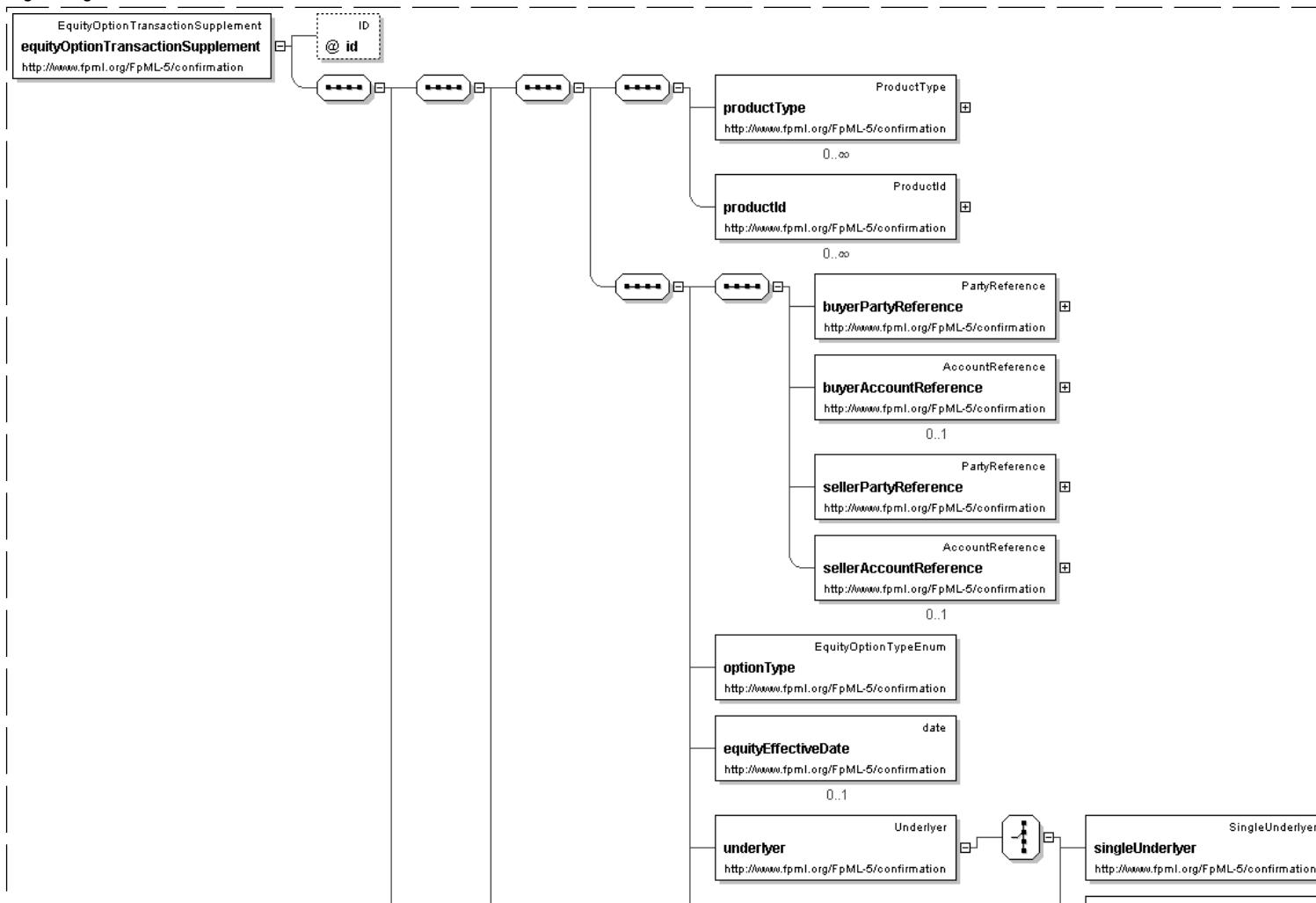
top

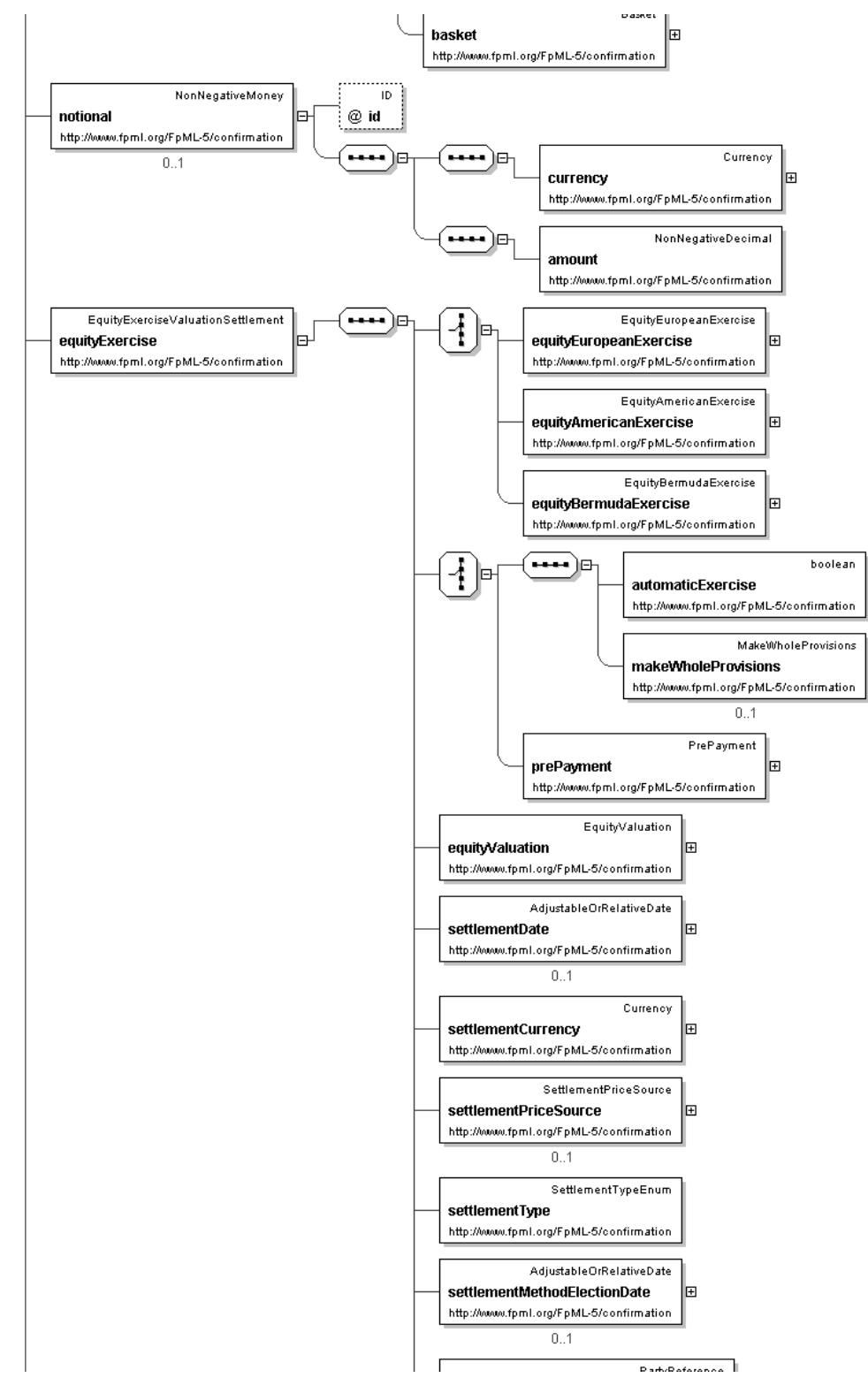
## Element: equityOptionTransactionSupplement

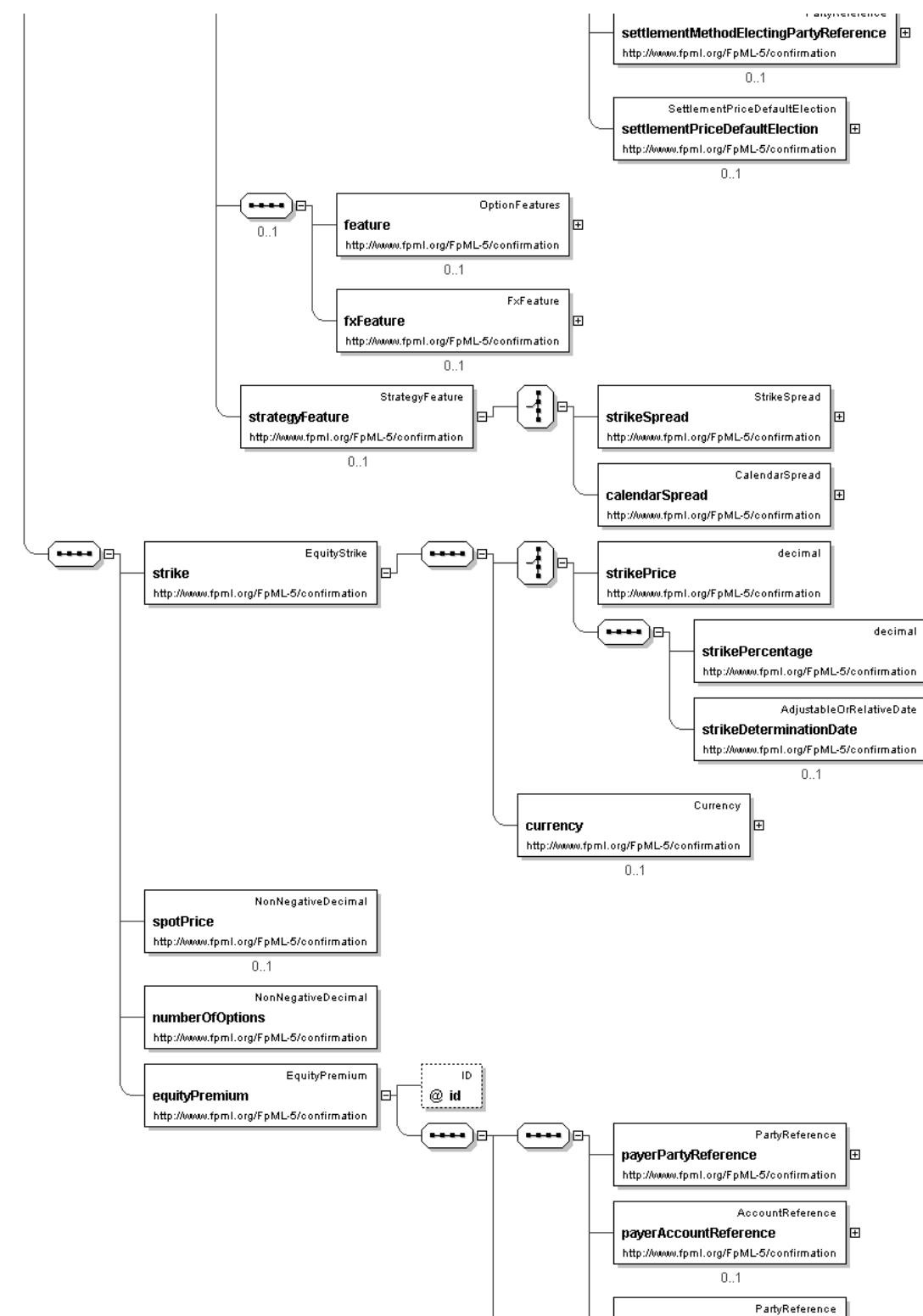
- This element can be used wherever the following element is referenced:
  - product

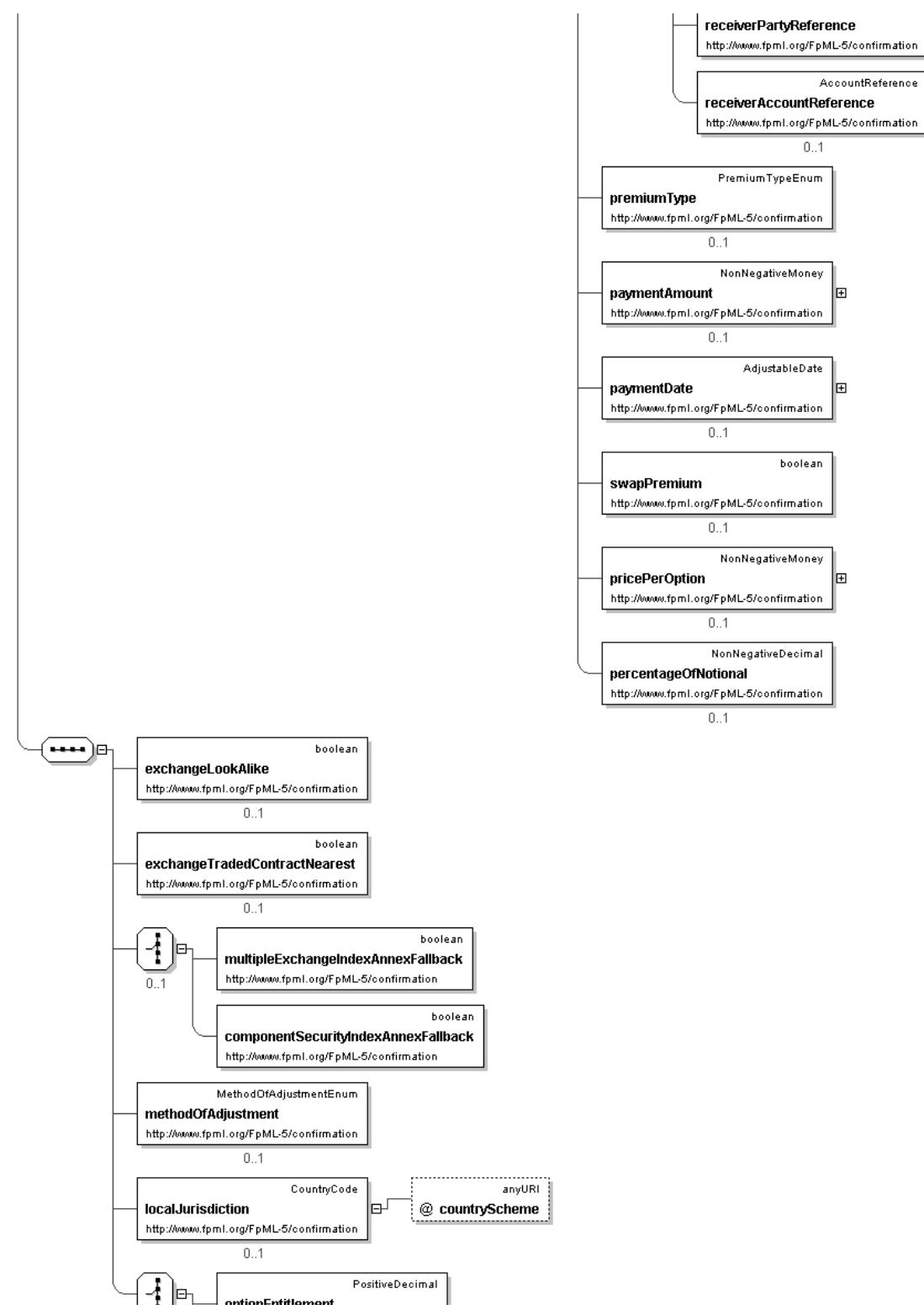
Name	equityOptionTransactionSupplement
Type	<a href="#">EquityOptionTransactionSupplement</a>
Nillable	no
Abstract	no
Documentation	A component describing an Equity Option Transaction Supplement.

## Logical Diagram









**XML Instance Representation**

```

<equityOptionTransactionSupplement
id=" xsd:ID [0..1]>
  <productType> ProductType </productType> [0..*]
  'A classification of the type of product. FpML defines a simple product categorization using
  a coding scheme.'

  <productId> ProductId </productId> [0..*]
  'A product 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.'

  <buyerPartyReference> PartyReference </buyerPartyReference> [1]
  'A reference to the party that buys this instrument, i.e. pays for this instrument and
  receives the rights defined by it. See 2000 ISDA definitions Article 11.1 (b). In the case
  of FRAs this is the fixed rate payer.'

  <buyerAccountReference> AccountReference </buyerAccountReference> [0..1]
  'A reference to the account that buys this instrument.'

  <sellerPartyReference> PartyReference </sellerPartyReference> [1]
  'A reference to the party that sells (\\"writes\\") this instrument, i.e. that grants the
  rights defined by this instrument and in return receives a payment for it. See 2000
  ISDA definitions Article 11.1 (a). In the case of FRAs this is the floating rate payer.'

  <sellerAccountReference> AccountReference </sellerAccountReference> [0..1]
  'A reference to the account that sells this instrument.'

  <optionType> EquityOptionTypeEnum </optionType> [1]
  'The type of option transaction.'

  <equityEffectiveDate> xsd:date </equityEffectiveDate> [0..1]
  'Effective date for a forward starting option.'

  <underlyer> Underlyer </underlyer> [1]
  'Specifies the underlying component, which can be either one or many and consists in
  either equity, index or convertible bond component, or a combination of these.'

  <notional> NonNegativeMoney </notional> [0..1]
  'The notional amount.'

  <equityExercise> EquityExerciseValuationSettlement </equityExercise> [1]
  'The parameters for defining how the equity option can be exercised, how it is valued and
  how it is settled.'

Start Group: Feature.model [0..1]
  <feature> OptionFeatures </feature> [0..1]
  'Asian, Barrier, Knock and Pass Through features.'

  <fxFeature> FxFeature </fxFeature> [0..1]
  'Quanto, Composite, or Cross Currency FX features.'

End Group: Feature.model
  <strategyFeature> StrategyFeature </strategyFeature> [0..1]
  'A equity option simple strategy feature.'

  <strike> EquityStrike </strike> [1]
  'Defines whether it is a price or level at which the option has been, or will be, struck.'

```

```

<spotPrice> NonNegativeDecimal </spotPrice> [0..1]
'The price per share, index or basket observed on the trade or effective date.'

<numberOfOptions> NonNegativeDecimal </numberOfOptions> [1]
'The number of options comprised in the option transaction.'

<equityPremium> EquityPremium </equityPremium> [1]
'The equity option premium payable by the buyer to the seller.'

<exchangeLookAlike> xsd:boolean </exchangeLookAlike> [0..1]
'For a share option transaction, a flag used to indicate whether the transaction is to be treated as an \'exchange look-alike\'. This designation has significance for how share adjustments (arising from corporate actions) will be determined for the transaction. For an \'exchange look-alike\' transaction the relevant share adjustments will follow that for a corresponding designated contract listed on the related exchange (referred to as Options Exchange Adjustment (ISDA defined term)), otherwise the share adjustments will be determined by the calculation agent (referred to as Calculation Agent Adjustment (ISDA defined term)).'

<exchangeTradedContractNearest> xsd:boolean </exchangeTradedContractNearest> [0..1]
'For an index option transaction, a flag used in conjunction with Futures Price Valuation (ISDA defined term) to indicate whether the Nearest Index Contract provision is applicable. The Nearest Index Contract provision is a rule for determining the Exchange-traded Contract (ISDA defined term) without having to explicitly state the actual contract, delivery month and exchange on which it is traded.'

Start Group: IndexAnnexFallback.model [0..1]
Start Choice [1]
  <multipleExchangeIndexAnnexFallback> xsd:boolean </multipleExchangeIndexAnnexFallback> [1]
    'For an index option transaction, a flag to indicate whether a relevant Multiple Exchange Index Annex is applicable to the transaction. This annex defines additional provisions which are applicable where an index is comprised of component securities that are traded on multiple exchanges.'

  <componentSecurityIndexAnnexFallback> xsd:boolean </componentSecurityIndexAnnexFallback> [1]
    'For an index option transaction, a flag to indicate whether a relevant Component Security Index Annex is applicable to the transaction.'

End Choice
End Group: IndexAnnexFallback.model
<methodOfAdjustment> MethodOfAdjustmentEnum </methodOfAdjustment> [0..1]
<localJurisdiction> CountryCode </localJurisdiction> [0..1]
'Local Jurisdiction is a term used in the AEJ Master Confirmation, which is used to determine local taxes, which shall mean taxes, duties, and similar charges imposed by the taxing authority of the Local Jurisdiction If this element is not present Local Jurisdiction is Not Applicable.'

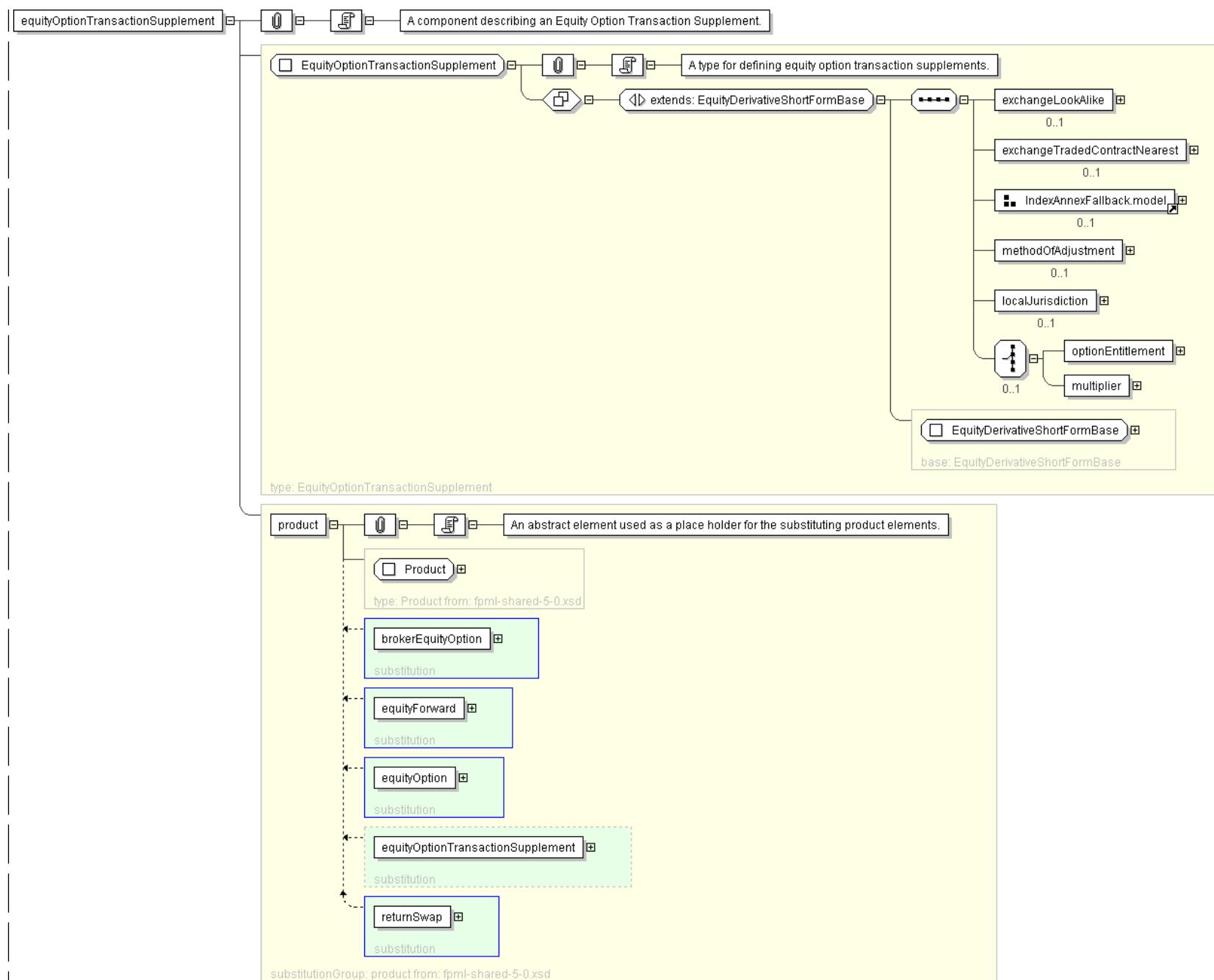
Start Choice [0..1]
  <optionEntitlement> PositiveDecimal </optionEntitlement> [1]
    'The number of shares per option comprised in the option transaction supplement.'

  <multiplier> PositiveDecimal </multiplier> [1]
    'Specifies the contract multiplier that can be associated with an index option.'

End Choice
</equityOptionTransactionSupplement>

```

**Diagram**

**Schema Component Representation**

```
<xsd:element name="equityOptionTransactionSupplement" type="EquityOptionTransactionSupplement" substitutionGroup="product"/>
```

## Global Definitions

### Complex Type: BrokerEquityOption

Super-types:	<a href="#">Product</a> < <a href="#">EquityDerivativeBase</a> (by extension) < <a href="#">EquityDerivativeShortFormBase</a> (by extension) < <b>BrokerEquityOption</b> (by extension)
Sub-types:	None

Name	BrokerEquityOption
Used by (from the same schema document)	Element <a href="#">brokerEquityOption</a>
Abstract	no
Documentation	A type for defining the broker equity options.

#### XML Instance Representation

```
<...>
  id="xsd:ID [0..1]">
    <productType> ProductType </productType> [0..*]
      'A classification of the type of product. FpML defines a simple product categorization using
       a coding scheme.'
    <productId> ProductId </productId> [0..*]
      'A product 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.'
    <buyerPartyReference> PartyReference </buyerPartyReference> [1]
      'A reference to the party that buys this instrument, i.e. pays for this instrument and
       receives the rights defined by it. See 2000 ISDA definitions Article 11.1 (b). In the case
       of FRAs this is the fixed rate payer.'
    <buyerAccountReference> AccountReference </buyerAccountReference> [0..1]
      'A reference to the account that buys this instrument.'
    <sellerPartyReference> PartyReference </sellerPartyReference> [1]
      'A reference to the party that sells ("writes") this instrument, i.e. that grants the
       rights defined by this instrument and in return receives a payment for it. See 2000
       ISDA definitions Article 11.1 (a). In the case of FRAs this is the floating rate payer.'
    <sellerAccountReference> AccountReference </sellerAccountReference> [0..1]
      'A reference to the account that sells this instrument.'
    <optionType> EquityOptionTypeEnum </optionType> [1]
      'The type of option transaction.'
    <equityEffectiveDate> xsd:date </equityEffectiveDate> [0..1]
      'Effective date for a forward starting option.'
    <underlyer> Underlyer </underlyer> [1]
      'Specifies the underlying component, which can be either one or many and consists in
       either equity, index or convertible bond component, or a combination of these.'
    <notional> NonNegativeMoney </notional> [0..1]
      'The notional amount.'
    <equityExercise> EquityExerciseValuationSettlement </equityExercise> [1]
      'The parameters for defining how the equity option can be exercised, how it is valued and
       how it is settled.'
  Start Group: Feature.model [0..1]
    <feature> OptionFeatures </feature> [0..1]
      'Asian, Barrier, Knock and Pass Through features.'
    <fxFeature> FxFeature </fxFeature> [0..1]
      'Quanto, Composite, or Cross Currency FX features.'
```

```

End Group: Feature.model
<strategyFeature> StrategyFeature </strategyFeature> [0..1]
'A equity option simple strategy feature.'

<strike> EquityStrike </strike> [1]
'Defines whether it is a price or level at which the option has been, or will be, struck.'

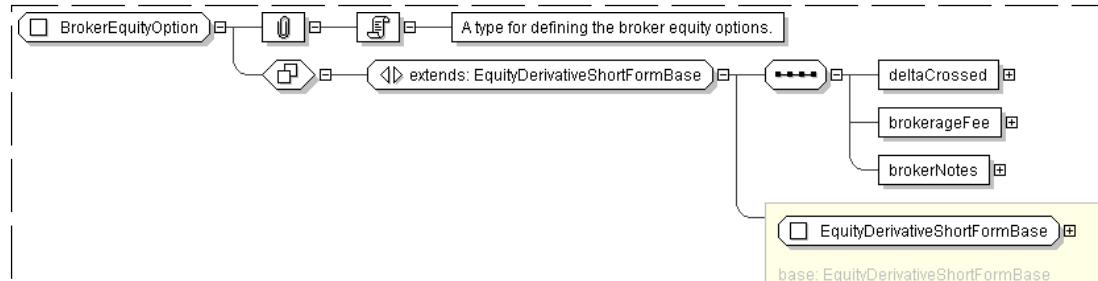
<spotPrice> NonNegativeDecimal </spotPrice> [0..1]
'The price per share, index or basket observed on the trade or effective date.'

<numberOfOptions> NonNegativeDecimal </numberOfOptions> [1]
'The number of options comprised in the option transaction.'

<equityPremium> EquityPremium </equityPremium> [1]
'The equity option premium payable by the buyer to the seller.'

<deltaCrossed> xsd:boolean </deltaCrossed> [1]
<brokerageFee> Money </brokerageFee> [1]
<brokerNotes> xsd:string </brokerNotes> [1]
<...>

```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="BrokerEquityOption">
  <xsd:complexContent>
    <xsd:extension base=" EquityDerivativeShortFormBase ">
      <xsd:sequence>
        <xsd:element name="deltaCrossed" type=" xsd:boolean " />
        <xsd:element name="brokerageFee" type=" Money " />
        <xsd:element name="brokerNotes" type=" xsd:string " />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

top

**Complex Type: EquityAmericanExercise**

**Super-types:** [SharedAmericanExercise](#) < **EquityAmericanExercise** (by extension)  
**Sub-types:** None

<b>Name</b>	EquityAmericanExercise
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">EquityExerciseValuationSettlement</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type for defining exercise procedures associated with an American style exercise of an equity option. This entity inherits from the type SharedAmericanExercise.

**XML Instance Representation**

```

<...
|
```

```

id=" xsd:ID [ 0..1 ]>
<commencementDate> AdjustableOrRelativeDate </commencementDate> [ 1 ]
'The first day of the exercise period for an American style option.'

<expirationDate> AdjustableOrRelativeDate </expirationDate> [ 1 ]
'The last day within an exercise period for an American style option. For a European
style option it is the only day within the exercise period.'

Start Choice [ 0..1 ]
'Choice between latest exercise time expressed as literal time, or using a
determination method.'

<latestExerciseTime> BusinessCenterTime </latestExerciseTime> [ 1 ]
'For a Bermuda or American style option, the latest time on an exercise business day
(excluding the expiration date) within the exercise period that notice can be given by
the 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.'

<latestExerciseTimeDetermination> DeterminationMethod </latestExerciseTimeDetermination> [ 1 ]
'Latest exercise time determination method.'

End Choice
<latestExerciseTimeType> TimeTypeEnum </latestExerciseTimeType> [ 0..1 ]
'The latest time of day at which the equity option can be exercised, for example the
official closing time of the exchange.'

Start Choice [ 1 ]
<equityExpirationTimeType> TimeTypeEnum </equityExpirationTimeType> [ 1 ]
'The time of day at which the equity option expires, for example the official closing time
of the exchange.'

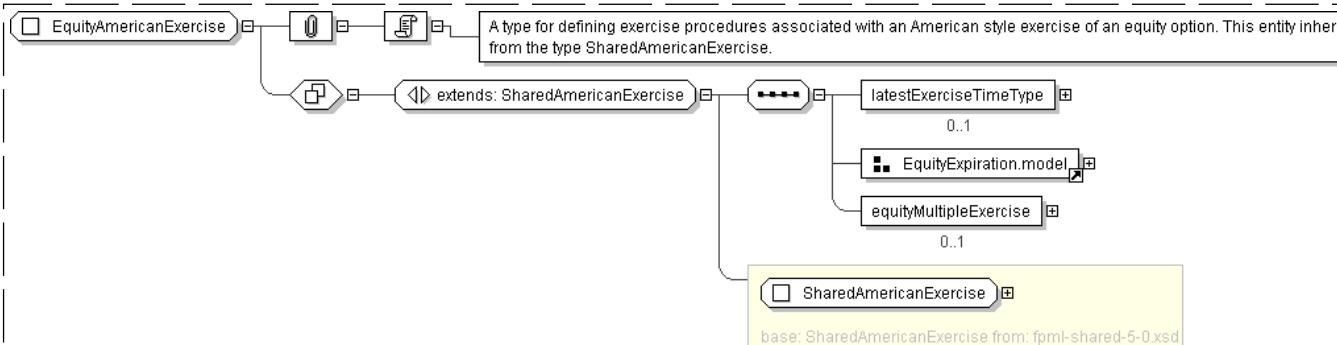
<equityExpirationTime> BusinessCenterTime </equityExpirationTime> [ 0..1 ]
'The specific time of day at which the equity option expires.'

<expirationTimeDetermination> DeterminationMethod </expirationTimeDetermination> [ 1 ]
'Expiration time determination method.'

End Choice
<equityMultipleExercise> EquityMultipleExercise </equityMultipleExercise> [ 0..1 ]
'The presence of this element indicates that the option may be exercised on different days.
It is not applicable to European options.

</...>

```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="EquityAmericanExercise">
<xsd:complexContent>
  <xsd:extension base=" SharedAmericanExercise ">

```

```

<xsd:sequence>
  <xsd:element name="latestExerciseTimeType" type=" TimeTypeEnum " minOccurs="0" />
  <xsd:group ref=" EquityExpiration.model "/>
  <xsd:element name="equityMultipleExercise" type=" EquityMultipleExercise " minOccurs="0" />
</xsd:sequence>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>

```

## Complex Type: **EquityBermudaExercise**

<b>Super-types:</b>	<a href="#">SharedAmericanExercise</a> < <b>EquityBermudaExercise</b> (by extension)
<b>Sub-types:</b>	None

<b>Name</b>	EquityBermudaExercise
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">EquityExerciseValuationSettlement</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type for defining exercise procedures associated with a Bermuda style exercise of an equity option. The term Bermuda is adopted in FpML for consistency with the ISDA Definitions.

### XML Instance Representation

```

<...
  id=" xsd:ID [0..1]">
    <commencementDate> AdjustableOrRelativeDate </commencementDate> [1]
    'The first day of the exercise period for an American style option.'

    <expirationDate> AdjustableOrRelativeDate </expirationDate> [1]
    'The last day within an exercise period for an American style option. For a European
    style option it is the only day within the exercise period.'

  Start Choice [0..1]
  'Choice between latest exercise time expressed as literal time, or using a
  determination method.'

    <latestExerciseTime> BusinessCenterTime </latestExerciseTime> [1]
    'For a Bermuda or American style option, the latest time on an exercise business day
    (excluding the expiration date) within the exercise period that notice can be given by
    the 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.'

    <latestExerciseTimeDetermination> DeterminationMethod </latestExerciseTimeDetermination> [1]
    'Latest exercise time determination method.'

  End Choice
    <bermudaExerciseDates> DateList </bermudaExerciseDates> [1]
    'List of Exercise Dates for a Bermuda option.'

    <latestExerciseTimeType> TimeTypeEnum </latestExerciseTimeType> [0..1]
    'The latest time of day at which the equity option can be exercised, for example the
    official closing time of the exchange.'

  Start Choice [1]
    <equityExpirationTimeType> TimeTypeEnum </equityExpirationTimeType> [1]
    'The time of day at which the equity option expires, for example the official closing time
    of the exchange.'

    <equityExpirationTime> BusinessCenterTime </equityExpirationTime> [0..1]
    'The specific time of day at which the equity option expires.'

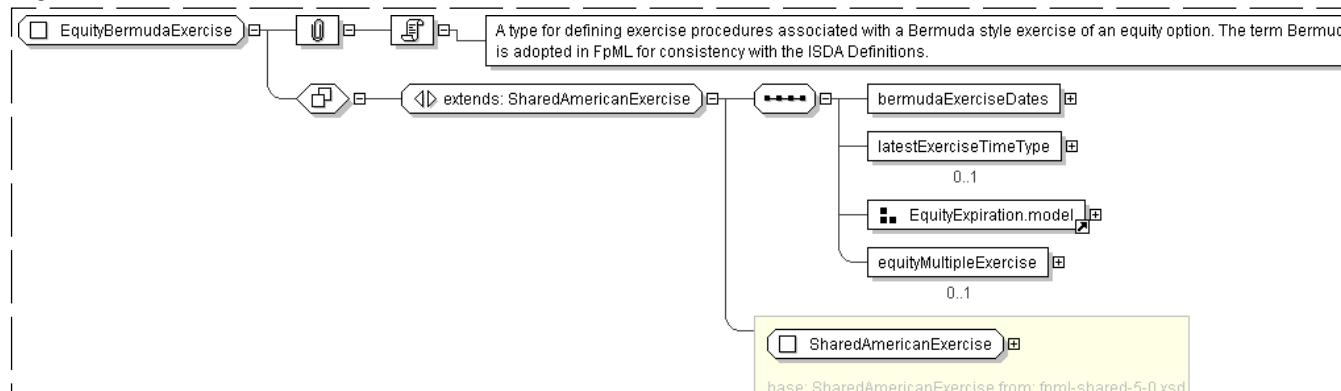
    <expirationTimeDetermination> DeterminationMethod </expirationTimeDetermination> [1]
    'Expiration time determination method.'

```

End Choice

<equityMultipleExercise> **EquityMultipleExercise** </equityMultipleExercise> [0..1]'The presence of this element indicates that the option may be exercised on different days.  
It is not applicable to European options.'

&lt;/...&gt;

**Diagram****Schema Component Representation**

```

<xsd:complexType name="EquityBermudaExercise">
  <xsd:complexContent>
    <xsd:extension base=" SharedAmericanExercise ">
      <xsd:sequence>
        <xsd:element name="bermudaExerciseDates" type=" DateList " />
        <xsd:element name="latestExerciseTimeType" type=" TimeTypeEnum " minOccurs="0" />
        <xsd:group ref=" EquityExpiration.model "/>
        <xsd:element name="equityMultipleExercise" type=" EquityMultipleExercise " minOccurs="0" />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
  
```

top

**Complex Type: EquityDerivativeBase**

Super-types:

[Product](#) < **EquityDerivativeBase** (by extension)

Sub-types:

- [EquityDerivativeLongFormBase](#) (by extension)
  - [EquityForward](#) (by extension)
  - [EquityOption](#) (by extension)
- [EquityDerivativeShortFormBase](#) (by extension)
  - [BrokerEquityOption](#) (by extension)
  - [EquityOptionTransactionSupplement](#) (by extension)

**Name**

EquityDerivativeBase

**Abstract**

yes

**Documentation**

A type for defining the common features of equity derivatives.

**XML Instance Representation**

```

<...
  id=" xsd:ID [0..1]">
  <productType> ProductType </productType> [0..*]
  'A classification of the type of product. FpML defines a simple product categorization using a coding scheme.'
  <productId> ProductId </productId> [0..*]
  
```

'A product 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.'

<buyerPartyReference> PartyReference </buyerPartyReference> [1]

'A reference to the party that buys this instrument, i.e. pays for this instrument and receives the rights defined by it. See 2000 ISDA definitions Article 11.1 (b). In the case of FRAs this is the fixed rate payer.'

<buyerAccountReference> AccountReference </buyerAccountReference> [0..1]

'A reference to the account that buys this instrument.'

<sellerPartyReference> PartyReference </sellerPartyReference> [1]

'A reference to the party that sells ("writes") this instrument, i.e. that grants the rights defined by this instrument and in return receives a payment for it. See 2000 ISDA definitions Article 11.1 (a). In the case of FRAs this is the floating rate payer.'

<sellerAccountReference> AccountReference </sellerAccountReference> [0..1]

'A reference to the account that sells this instrument.'

<optionType> EquityOptionTypeEnum </optionType> [1]

'The type of option transaction.'

<equityEffectiveDate> xsd:date </equityEffectiveDate> [0..1]

'Effective date for a forward starting option.'

<underlyer> Underlyer </underlyer> [1]

'Specifies the underlying component, which can be either one or many and consists in either equity, index or convertible bond component, or a combination of these.'

<notional> NonNegativeMoney </notional> [0..1]

'The notional amount.'

<equityExercise> EquityExerciseValuationSettlement </equityExercise> [1]

'The parameters for defining how the equity option can be exercised, how it is valued and how it is settled.'

Start Group: Feature.model [0..1]

<feature> OptionFeatures </feature> [0..1]

'Asian, Barrier, Knock and Pass Through features.'

<fxFeature> FxFeature </fxFeature> [0..1]

'Quanto, Composite, or Cross Currency FX features.'

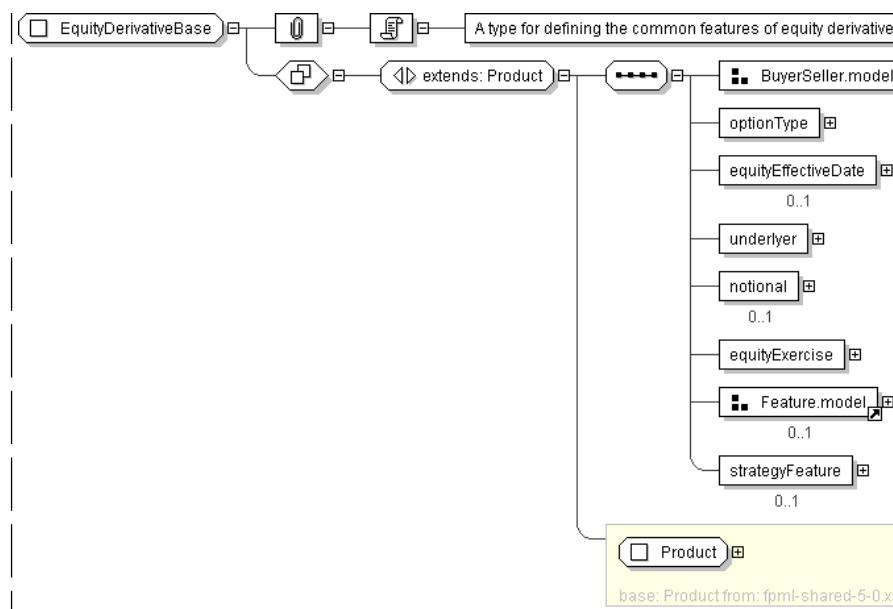
End Group: Feature.model

<strategyFeature> StrategyFeature </strategyFeature> [0..1]

'A equity option simple strategy feature.'

</...>

**Diagram**

**Schema Component Representation**

```

<xsd:complexType name="EquityDerivativeBase" abstract="true">
  <xsd:complexContent>
    <xsd:extension base="#Product">
      <xsd:sequence>
        <xsd:group ref="BuyerSeller.model"/>
        <xsd:element name="optionType" type="EquityOptionTypeEnum"/>
        <xsd:element name="equityEffectiveDate" type="xsd:date" minOccurs="0"/>
        <xsd:element name="underlyer" type="Underlyer"/>
        <xsd:element name="notional" type="NonNegativeMoney" minOccurs="0"/>
        <xsd:element name="equityExercise" type="EquityExerciseValuationSettlement"/>
        <xsd:group ref="Feature.model" minOccurs="0"/>
        <xsd:element name="strategyFeature" type="StrategyFeature" minOccurs="0"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

[top](#)**Complex Type: EquityDerivativeLongFormBase**

**Super-types:** [Product](#) < [EquityDerivativeBase](#) (by extension) < [EquityDerivativeLongFormBase](#) (by extension)

**Sub-types:**

- [EquityForward](#) (by extension)
- [EquityOption](#) (by extension)

<b>Name</b>	EquityDerivativeLongFormBase
-------------	------------------------------

<b>Abstract</b>	yes
-----------------	-----

<b>Documentation</b>	type for defining the common features of equity derivatives.
----------------------	--

**XML Instance Representation**

```

<...
  id="xsd:ID" [0..1]>
  <productType> ProductType </productType> [0..*]
  'A classification of the type of product. FpML defines a simple product categorization using
  a coding scheme.'
  <productId> ProductId </productId> [0..*]

```

'A product 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.'

<buyerPartyReference> PartyReference </buyerPartyReference> [1]

'A reference to the party that buys this instrument, i.e. pays for this instrument and receives the rights defined by it. See 2000 ISDA definitions Article 11.1 (b). In the case of FRAs this is the fixed rate payer.'

<buyerAccountReference> AccountReference </buyerAccountReference> [0..1]

'A reference to the account that buys this instrument.'

<sellerPartyReference> PartyReference </sellerPartyReference> [1]

'A reference to the party that sells ("writes") this instrument, i.e. that grants the rights defined by this instrument and in return receives a payment for it. See 2000 ISDA definitions Article 11.1 (a). In the case of FRAs this is the floating rate payer.'

<sellerAccountReference> AccountReference </sellerAccountReference> [0..1]

'A reference to the account that sells this instrument.'

<optionType> EquityOptionTypeEnum </optionType> [1]

'The type of option transaction.'

<equityEffectiveDate> xsd:date </equityEffectiveDate> [0..1]

'Effective date for a forward starting option.'

<underlyer> Underlyer </underlyer> [1]

'Specifies the underlying component, which can be either one or many and consists in either equity, index or convertible bond component, or a combination of these.'

<notional> NonNegativeMoney </notional> [0..1]

'The notional amount.'

<equityExercise> EquityExerciseValuationSettlement </equityExercise> [1]

'The parameters for defining how the equity option can be exercised, how it is valued and how it is settled.'

Start Group: Feature.model [0..1]

<feature> OptionFeatures </feature> [0..1]

'Asian, Barrier, Knock and Pass Through features.'

<fxFeature> FxFeature </fxFeature> [0..1]

'Quanto, Composite, or Cross Currency FX features.'

End Group: Feature.model

<strategyFeature> StrategyFeature </strategyFeature> [0..1]

'A equity option simple strategy feature.'

<dividendConditions> DividendConditions </dividendConditions> [0..1]

<methodOfAdjustment> MethodOfAdjustmentEnum </methodOfAdjustment> [1]

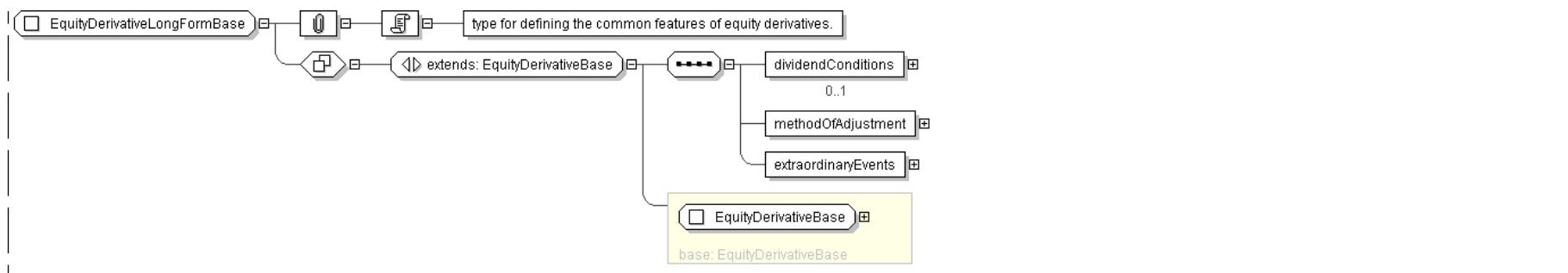
'Defines how adjustments will be made to the contract should one or more of the extraordinary events occur.'

<extraordinaryEvents> ExtraordinaryEvents </extraordinaryEvents> [1]

'Where the underlying is shares, specifies events affecting the issuer of those shares that may require the terms of the transaction to be adjusted.'

</...>

## Diagram

**Schema Component Representation**

```

<xsd:complexType name="EquityDerivativeLongFormBase" abstract="true">
  <xsd:complexContent>
    <xsd:extension base=" EquityDerivativeBase ">
      <xsd:sequence>
        <xsd:element name="dividendConditions" type=" DividendConditions " minOccurs="0"/>
        <xsd:element name="methodOfAdjustment" type=" MethodOfAdjustmentEnum "/>
        <xsd:element name="extraordinaryEvents" type=" ExtraordinaryEvents "/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
  
```

top

**Complex Type: [EquityDerivativeShortFormBase](#)**

<b>Super-types:</b>	<a href="#">Product</a> < <a href="#">EquityDerivativeBase</a> (by extension) < <b>EquityDerivativeShortFormBase</b> (by extension)
<b>Sub-types:</b>	<ul style="list-style-type: none"> <li>• <a href="#">BrokerEquityOption</a> (by extension)</li> <li>• <a href="#">EquityOptionTransactionSupplement</a> (by extension)</li> </ul>

<b>Name</b>	<a href="#">EquityDerivativeShortFormBase</a>
<b>Abstract</b>	yes
<b>Documentation</b>	A type for defining short form equity option basic features.

**XML Instance Representation**

```

<...
  id=" xsd:ID [0..1]">
  <productType> ProductType </productType> [0..*]
  'A classification of the type of product. FpML defines a simple product categorization using
  a coding scheme.'
  <productId> ProductId </productId> [0..*]
  'A product 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.'
  <buyerPartyReference> PartyReference </buyerPartyReference> [1]
  'A reference to the party that buys this instrument, i.e. pays for this instrument and
  receives the rights defined by it. See 2000 ISDA definitions Article 11.1 (b). In the case
  of FRAs this is the fixed rate payer.'
  <buyerAccountReference> AccountReference </buyerAccountReference> [0..1]
  'A reference to the account that buys this instrument.'
  <sellerPartyReference> PartyReference </sellerPartyReference> [1]
  'A reference to the party that sells ("writes") this instrument, i.e. that grants the
  rights defined by this instrument and in return receives a payment for it. See 2000
  ISDA definitions Article 11.1 (a). In the case of FRAs this is the floating rate payer.'
  <sellerAccountReference> AccountReference </sellerAccountReference> [0..1]
  
```

'A reference to the account that sells this instrument.'

```
<optionType> EquityOptionTypeEnum </optionType> [1]
```

'The type of option transaction.'

```
<equityEffectiveDate> xsd:date </equityEffectiveDate> [0..1]
```

'Effective date for a forward starting option.'

```
<underlyer> Underlyer </underlyer> [1]
```

'Specifies the underlying component, which can be either one or many and consists in either equity, index or convertible bond component, or a combination of these.'

```
<notional> NonNegativeMoney </notional> [0..1]
```

'The notional amount.'

```
<equityExercise> EquityExerciseValuationSettlement </equityExercise> [1]
```

'The parameters for defining how the equity option can be exercised, how it is valued and how it is settled.'

Start Group: Feature.model [0..1]

```
<feature> OptionFeatures </feature> [0..1]
```

'Asian, Barrier, Knock and Pass Through features.'

```
<fxFeature> FxFeature </fxFeature> [0..1]
```

'Quanto, Composite, or Cross Currency FX features.'

End Group: Feature.model

```
<strategyFeature> StrategyFeature </strategyFeature> [0..1]
```

'A equity option simple strategy feature.'

```
<strike> EquityStrike </strike> [1]
```

'Defines whether it is a price or level at which the option has been, or will be, struck.'

```
<spotPrice> NonNegativeDecimal </spotPrice> [0..1]
```

'The price per share, index or basket observed on the trade or effective date.'

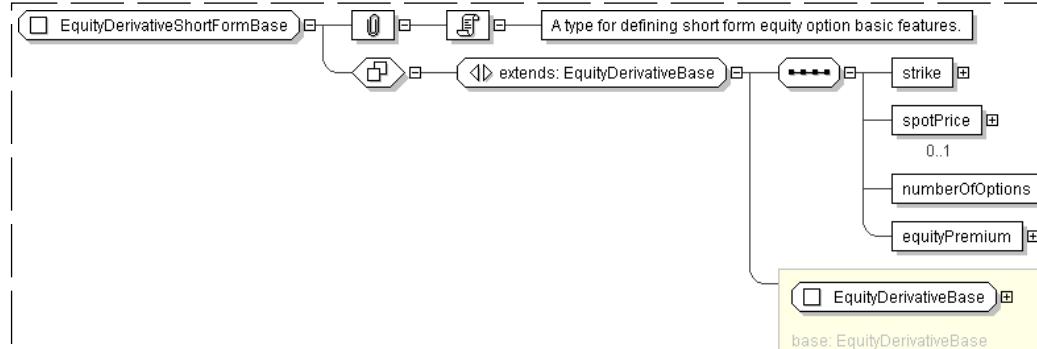
```
<numberOfOptions> NonNegativeDecimal </numberOfOptions> [1]
```

'The number of options comprised in the option transaction.'

```
<equityPremium> EquityPremium </equityPremium> [1]
```

'The equity option premium payable by the buyer to the seller.'

```
<...>
```

**Diagram****Schema Component Representation**

```
<xsd:complexType name="EquityDerivativeShortFormBase" abstract="true">
  <xsd:complexContent>
```

```

<xsd:extension base=" EquityDerivativeBase ">
  <xsd:sequence>
    <xsd:element name="strike" type=" EquityStrike " />
    <xsd:element name="spotPrice" type=" NonNegativeDecimal " minOccurs="0" />
    <xsd:element name="numberOfOptions" type=" NonNegativeDecimal " />
    <xsd:element name="equityPremium" type=" EquityPremium " />
  </xsd:sequence>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>

```

[top](#)

## Complex Type: [EquityEuropeanExercise](#)

Super-types:	<a href="#">Exercise</a> < <b>EquityEuropeanExercise</b> > (by extension)
Sub-types:	None

Name	EquityEuropeanExercise
Used by (from the same schema document)	Complex Type <a href="#">EquityExerciseValuationSettlement</a>
Abstract	no
Documentation	A type for defining exercise procedures associated with a European style exercise of an equity option.

### XML Instance Representation

```

<...
  id=" xsd:ID [0..1]">
  <expirationDate> AdjustableOrRelativeDate </expirationDate> [1]
  'The last day within an exercise period for an American style option. For a European
  style option it is the only day within the exercise period.'

Start Choice [1]
  <equityExpirationTimeType> TimeTypeEnum </equityExpirationTimeType> [1]
  'The time of day at which the equity option expires, for example the official closing time
  of the exchange.'

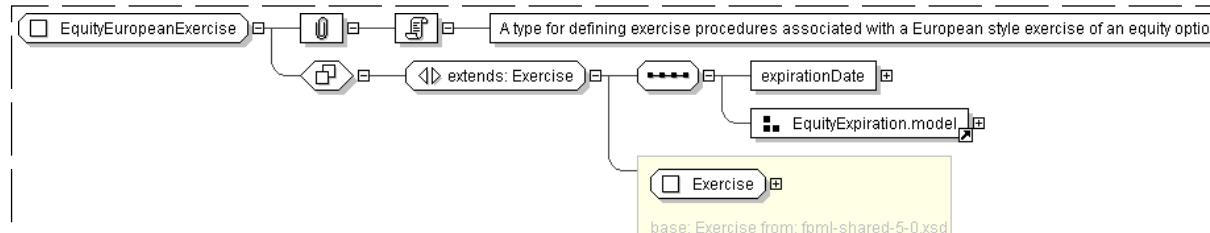
  <equityExpirationTime> BusinessCenterTime </equityExpirationTime> [0..1]
  'The specific time of day at which the equity option expires.'

  <expirationTimeDetermination> DeterminationMethod </expirationTimeDetermination> [1]
  'Expiration time determination method.'

End Choice
</...>

```

### Diagram



### Schema Component Representation

```

<xsd:complexType name="EquityEuropeanExercise">
  <xsd:complexContent>
    <xsd:extension base=" Exercise ">
      <xsd:sequence>
        <xsd:element name="expirationDate" type=" AdjustableOrRelativeDate " />
        <xsd:group ref=" EquityExpiration.model " />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

```
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>
```

## Complex Type: [EquityExerciseValuationSettlement](#)

<b>Super-types:</b>	None
<b>Sub-types:</b>	None
<b>Name</b>	<a href="#">EquityExerciseValuationSettlement</a>
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">EquityDerivativeBase</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type for defining exercise procedures for equity options.

### XML Instance Representation

```
<...>
Start Choice [1]
'The parameters for defining how the equity option can be exercised, how it is valued and
how it is settled.'

<equityEuropeanExercise> EquityEuropeanExercise </equityEuropeanExercise> [1]
'The parameters for defining the expiration date and time for a European style equity option.'

<equityAmericanExercise> EquityAmericanExercise </equityAmericanExercise> [1]
'The parameters for defining the exercise period for an American style equity option
together with the rules governing the quantity of the underlying that can be exercised on
any given exercise date.'

<equityBermudaExercise> EquityBermudaExercise </equityBermudaExercise> [1]
'The parameters for defining the exercise period for an Bermuda style equity option
together with the rules governing the quantity of the underlying that can be exercised on
any given exercise date.'

End Choice
Start Choice [1]
<automaticExercise> xsd:boolean </automaticExercise> [1]
'If true then each option not previously exercised will be deemed to be exercised at
the expiration time on the expiration date without service of notice unless the buyer
notifies the seller that it no longer wishes this to occur.'

<makeWholeProvisions> MakeWholeProvisions </makeWholeProvisions> [0..1]
'Provisions covering early exercise of option.'

<prePayment> PrePayment </prePayment> [1]
'Prepayment features for Forward.'

End Choice
<equityValuation> EquityValuation </equityValuation> [1]
'The parameters for defining when valuation of the underlying takes place.'

<settlementDate> AdjustableOrRelativeDate </settlementDate> [0..1]
'Date on which settlement of option premiums will occur.'

<settlementCurrency> Currency </settlementCurrency> [1]
'The currency in which a cash settlement for non-deliverable forward and non-
deliverable options.'

<settlementPriceSource> SettlementPriceSource </settlementPriceSource> [0..1]
<settlementType> SettlementTypeEnum </settlementType> [1]
'How the option will be settled.'

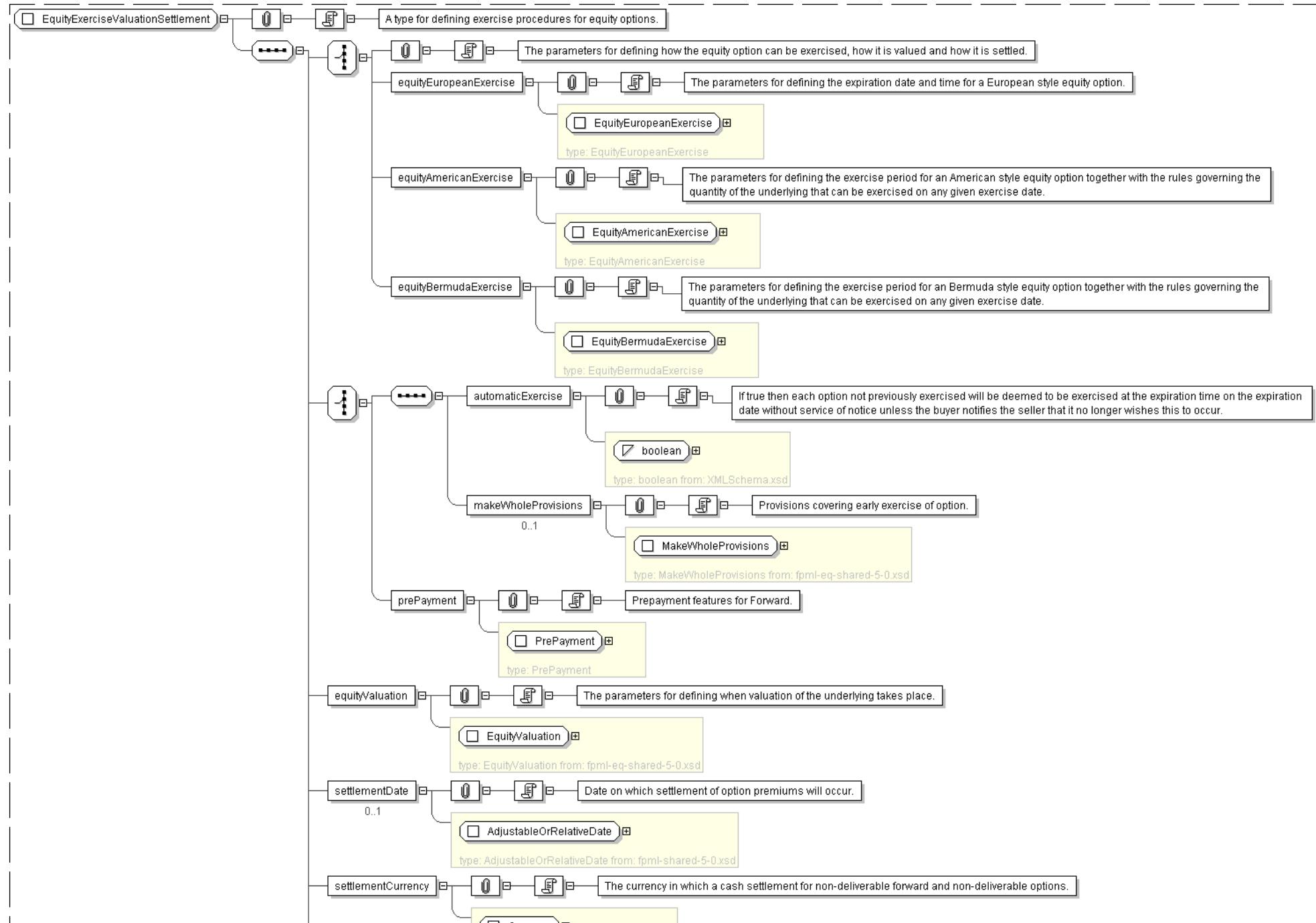
<settlementMethodElectionDate> AdjustableOrRelativeDate </settlementMethodElectionDate> [0..1]
```

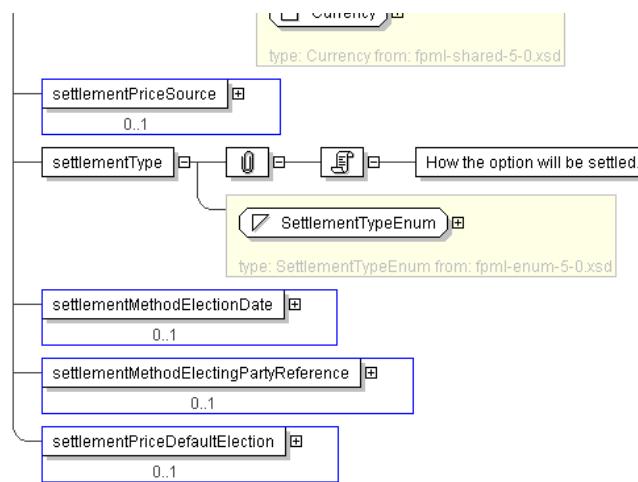
```

<settlementMethodElectingPartyReference> PartyReference
</settlementMethodElectingPartyReference> [0..1]
<settlementPriceDefaultElection> SettlementPriceDefaultElection
</settlementPriceDefaultElection> [0..1]
<...>

```

## Diagram





#### Schema Component Representation

```

<xsd:complexType name="EquityExerciseValuationSettlement">
  <xsd:sequence>
    <xsd:choice>
      <xsd:element name="equityEuropeanExercise" type="EquityEuropeanExercise" />
      <xsd:element name="equityAmericanExercise" type="EquityAmericanExercise" />
      <xsd:element name="equityBermudaExercise" type="EquityBermudaExercise" />
    </xsd:choice>
    <xsd:choice>
      <xsd:sequence>
        <xsd:element name="automaticExercise" type="xsd:boolean" />
        <xsd:element name="makeWholeProvisions" type="MakeWholeProvisions" minOccurs="0" />
      </xsd:sequence>
      <xsd:element name="prePayment" type="PrePayment" />
    </xsd:choice>
    <xsd:element name="equityValuation" type="EquityValuation" />
    <xsd:element name="settlementDate" type="AdjustableOrRelativeDate" minOccurs="0" />
    <xsd:element name="settlementCurrency" type="Currency" />
    <xsd:element name="settlementPriceSource" type="SettlementPriceSource" minOccurs="0" />
    <xsd:element name="settlementType" type="SettlementTypeEnum" />
    <xsd:element name="settlementMethodElectionDate" type="AdjustableOrRelativeDate"
      minOccurs="0" />
    <xsd:element name="settlementMethodElectingPartyReference" type="PartyReference"
      minOccurs="0" />
    <xsd:element name="settlementPriceDefaultElection" type="SettlementPriceDefaultElection"
      minOccurs="0" />
  </xsd:sequence>
</xsd:complexType>
  
```

top

#### Complex Type: **EquityForward**

Super-types:	<a href="#">Product</a> < <a href="#">EquityDerivativeBase</a> (by extension) < <a href="#">EquityDerivativeLongFormBase</a> (by extension) < <b>EquityForward</b> (by extension)
Sub-types:	None

Name	EquityForward
Used by (from the same schema document)	Element <a href="#">equityForward</a>
Abstract	no
Documentation	A type for defining equity forwards.

#### XML Instance Representation

```

<...
| id="xsd:ID [0..1]">
  
```

<productType> ProductType </productType> [0..\*]  
 'A classification of the type of product. FpML defines a simple product categorization using a coding scheme.'

<productId> ProductId </productId> [0..\*]  
 'A product 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.'

<buyerPartyReference> PartyReference </buyerPartyReference> [1]  
 'A reference to the party that buys this instrument, i.e. pays for this instrument and receives the rights defined by it. See 2000 ISDA definitions Article 11.1 (b). In the case of FRAs this is the fixed rate payer.'

<buyerAccountReference> AccountReference </buyerAccountReference> [0..1]  
 'A reference to the account that buys this instrument.'

<sellerPartyReference> PartyReference </sellerPartyReference> [1]  
 'A reference to the party that sells ("writes") this instrument, i.e. that grants the rights defined by this instrument and in return receives a payment for it. See 2000 ISDA definitions Article 11.1 (a). In the case of FRAs this is the floating rate payer.'

<sellerAccountReference> AccountReference </sellerAccountReference> [0..1]  
 'A reference to the account that sells this instrument.'

<optionType> EquityOptionTypeEnum </optionType> [1]  
 'The type of option transaction.'

<equityEffectiveDate> xsd:date </equityEffectiveDate> [0..1]  
 'Effective date for a forward starting option.'

<underlyer> Underlyer </underlyer> [1]  
 'Specifies the underlying component, which can be either one or many and consists in either equity, index or convertible bond component, or a combination of these.'

<notional> NonNegativeMoney </notional> [0..1]  
 'The notional amount.'

<equityExercise> EquityExerciseValuationSettlement </equityExercise> [1]  
 'The parameters for defining how the equity option can be exercised, how it is valued and how it is settled.'

Start Group: Feature.model [0..1]  
 <feature> OptionFeatures </feature> [0..1]  
 'Asian, Barrier, Knock and Pass Through features.'  
 <fxFeature> FxFeature </fxFeature> [0..1]  
 'Quanto, Composite, or Cross Currency FX features.'

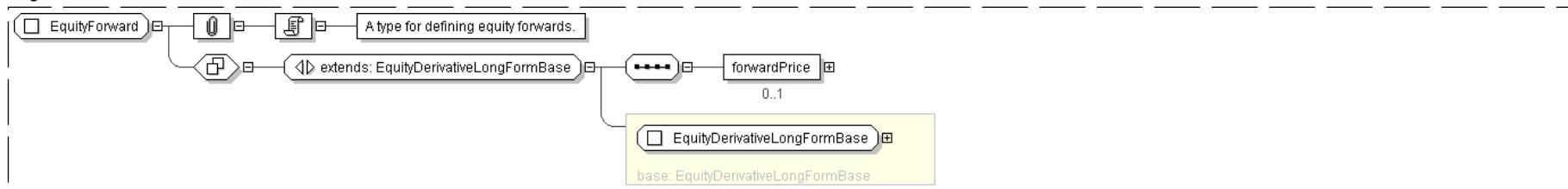
End Group: Feature.model  
 <strategyFeature> StrategyFeature </strategyFeature> [0..1]  
 'A equity option simple strategy feature.'

<dividendConditions> DividendConditions </dividendConditions> [0..1]  
 <methodOfAdjustment> MethodOfAdjustmentEnum </methodOfAdjustment> [1]  
 'Defines how adjustments will be made to the contract should one or more of the extraordinary events occur.'

<extraordinaryEvents> ExtraordinaryEvents </extraordinaryEvents> [1]  
 'Where the underlying is shares, specifies events affecting the issuer of those shares that may require the terms of the transaction to be adjusted.'

<forwardPrice> NonNegativeMoney </forwardPrice> [0..1]  
 'The forward price per share, index or basket.'

&lt;/...&gt;

**Diagram****Schema Component Representation**

```

<xsd:complexType name="EquityForward">
  <xsd:complexContent>
    <xsd:extension base=" EquityDerivativeLongFormBase ">
      <xsd:sequence>
        <xsd:element name="forwardPrice" type=" NonNegativeMoney " minOccurs="0"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
  
```

top

**Complex Type: `EquityMultipleExercise`**

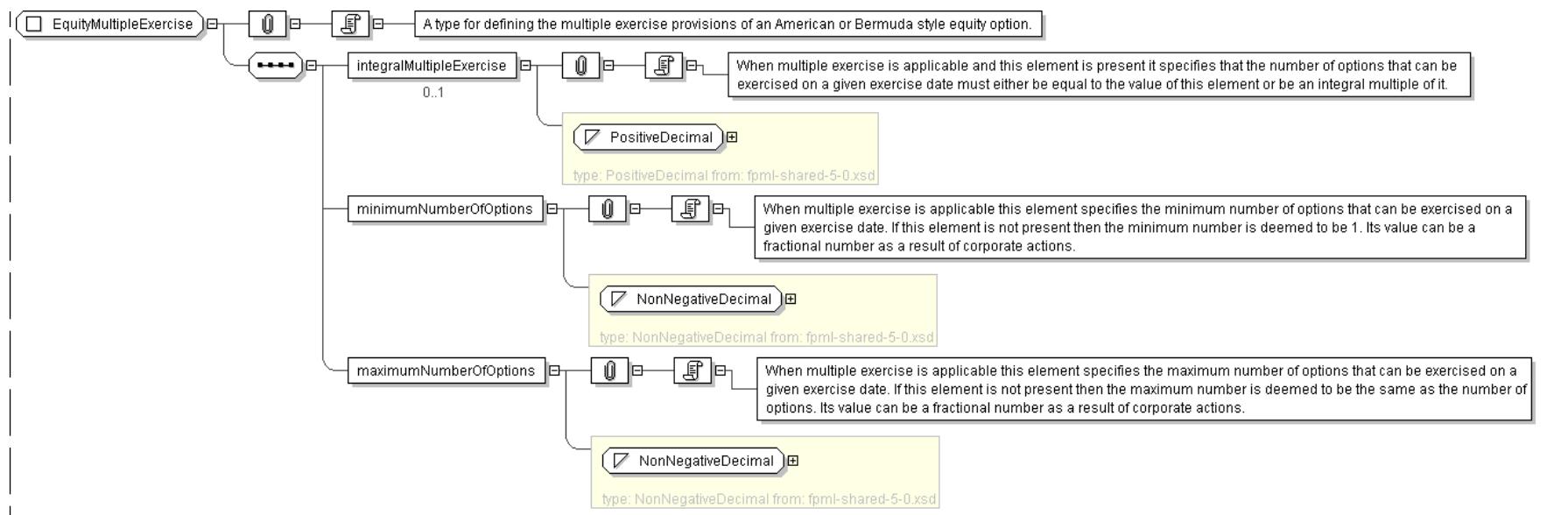
<b>Super-types:</b>	None
<b>Sub-types:</b>	None
<b>Name</b>	<code>EquityMultipleExercise</code>
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">EquityAmericanExercise</a> , Complex Type <a href="#">EquityBermudaExercise</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type for defining the multiple exercise provisions of an American or Bermuda style equity option.

**XML Instance Representation**

```

<....>
<integralMultipleExercise> PositiveDecimal </integralMultipleExercise> [0..1]
  'When multiple exercise is applicable and this element is present it specifies that the
  number of options that can be exercised on a given exercise date must either be equal to
  the value of this element or be an integral multiple of it.'
<minimumNumberOfOptions> NonNegativeDecimal </minimumNumberOfOptions> [1]
  'When multiple exercise is applicable this element specifies the minimum number of options
  that can be exercised on a given exercise date. If this element is not present then the
  minimum number is deemed to be 1. Its value can be a fractional number as a result of
  corporate actions.'
<maximumNumberOfOptions> NonNegativeDecimal </maximumNumberOfOptions> [1]
  'When multiple exercise is applicable this element specifies the maximum number of options
  that can be exercised on a given exercise date. If this element is not present then the
  maximum number is deemed to be the same as the number of options. Its value can be a
  fractional number as a result of corporate actions.'
</....>
  
```

**Diagram**

**Schema Component Representation**

```

<xsd:complexType name="EquityMultipleExercise">
  <xsd:sequence>
    <xsd:element name="integralMultipleExercise" type=" PositiveDecimal " minOccurs="0" />
    <xsd:element name="minimumNumberOfOptions" type=" NonNegativeDecimal " />
    <xsd:element name="maximumNumberOfOptions" type=" NonNegativeDecimal " />
  </xsd:sequence>
</xsd:complexType>
  
```

top

**Complex Type: `EquityOption`**

<b>Super-types:</b>	<code>Product</code> < <code>EquityDerivativeBase</code> (by extension) < <code>EquityDerivativeLongFormBase</code> (by extension) < <b>EquityOption</b> (by extension)
<b>Sub-types:</b>	None

<b>Name</b>	<code>EquityOption</code>
<b>Used by (from the same schema document)</b>	Element <a href="#">equityOption</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type for defining equity options.

**XML Instance Representation**

```

<...
  id=" xsd:ID [0..1]">
  <productType> ProductType </productType> [0..*]
  'A classification of the type of product. FpML defines a simple product categorization using a coding scheme.'
  <productId> ProductId </productId> [0..*]
  'A product 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.'
  <buyerPartyReference> PartyReference </buyerPartyReference> [1]
  'A reference to the party that buys this instrument, ie. pays for this instrument and receives the rights defined by it. See 2000 ISDA definitions Article 11.1 (b). In the case of FRAs this the fixed rate payer.'
  
```

```

<buyerAccountReference> AccountReference </buyerAccountReference> [0..1]
'A reference to the account that buys this instrument.'

<sellerPartyReference> PartyReference </sellerPartyReference> [1]
'A reference to the party that sells ("writes") this instrument, i.e. that grants the
rights defined by this instrument and in return receives a payment for it. See 2000
ISDA definitions Article 11.1 (a). In the case of FRAs this is the floating rate payer.'

<sellerAccountReference> AccountReference </sellerAccountReference> [0..1]
'A reference to the account that sells this instrument.'

<optionType> EquityOptionTypeEnum </optionType> [1]
'The type of option transaction.'

<equityEffectiveDate> xsd:date </equityEffectiveDate> [0..1]
'Effective date for a forward starting option.'

<underlyer> Underlyer </underlyer> [1]
'Specifies the underlying component, which can be either one or many and consists in
either equity, index or convertible bond component, or a combination of these.'

<notional> NonNegativeMoney </notional> [0..1]
'The notional amount.'

<equityExercise> EquityExerciseValuationSettlement </equityExercise> [1]
'The parameters for defining how the equity option can be exercised, how it is valued and
how it is settled.

Start Group: Feature.model [0..1]
<feature> OptionFeatures </feature> [0..1]
'Asian, Barrier, Knock and Pass Through features.'

<fxFeature> FxFeature </fxFeature> [0..1]
'Quanto, Composite, or Cross Currency FX features.'

End Group: Feature.model
<strategyFeature> StrategyFeature </strategyFeature> [0..1]
'A equity option simple strategy feature.'

<dividendConditions> DividendConditions </dividendConditions> [0..1]
<methodOfAdjustment> MethodOfAdjustmentEnum </methodOfAdjustment> [1]
'Defines how adjustments will be made to the contract should one or more of the
extraordinary events occur.'

<extraordinaryEvents> ExtraordinaryEvents </extraordinaryEvents> [1]
'Where the underlying is shares, specifies events affecting the issuer of those shares that
may require the terms of the transaction to be adjusted.'

<strike> EquityStrike </strike> [0..1]
'Defines whether it is a price or level at which the option has been, or will be, struck.'

<spotPrice> NonNegativeDecimal </spotPrice> [0..1]
'The price per share, index or basket observed on the trade or effective date.'

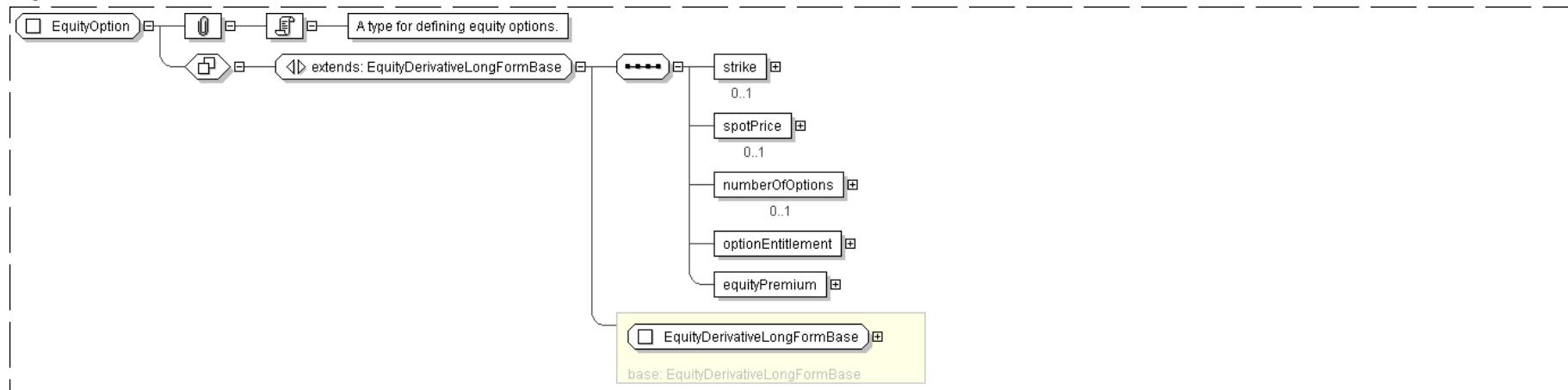
<numberOfOptions> NonNegativeDecimal </numberOfOptions> [0..1]
'The number of options comprised in the option transaction.'

<optionEntitlement> PositiveDecimal </optionEntitlement> [1]
'The number of shares per option comprised in the option transaction.'

<equityPremium> EquityPremium </equityPremium> [1]
'The equity option premium payable by the buyer to the seller.'

</...>

```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="EquityOption">
  <xsd:complexContent>
    <xsd:extension base=" EquityDerivativeLongFormBase ">
      <xsd:sequence>
        <xsd:element name="strike" type=" EquityStrike " minOccurs="0"/>
        <xsd:element name="spotPrice" type=" NonNegativeDecimal " minOccurs="0"/>
        <xsd:element name="numberOfOptions" type=" NonNegativeDecimal " minOccurs="0"/>
        <xsd:element name="optionEntitlement" type=" PositiveDecimal "/>
        <xsd:element name="equityPremium" type=" EquityPremium "/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

[top](#)**Complex Type: [EquityOptionTermination](#)**

**Super-types:** None  
**Sub-types:** None

<b>Name</b>	EquityOptionTermination
<b>Abstract</b>	no
<b>Documentation</b>	A type for defining Equity Option Termination.

**XML Instance Representation**

```

<...>
  <settlementAmountPaymentDate> AdjustableDate </settlementAmountPaymentDate> [1]
  <settlementAmount> NonNegativeMoney </settlementAmount> [1]
</...>

```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="EquityOptionTermination">

```

```

<xsd:sequence>
  <xsd:element name="settlementAmountPaymentDate" type=" AdjustableDate " />
  <xsd:element name="settlementAmount" type=" NonNegativeMoney " />
</xsd:sequence>
</xsd:complexType>

```

## Complex Type: [EquityOptionTransactionSupplement](#)

<b>Super-types:</b>	<a href="#">Product</a> < <a href="#">EquityDerivativeBase</a> (by extension) < <a href="#">EquityDerivativeShortFormBase</a> (by extension) < <b>EquityOptionTransactionSupplement</b>
<b>Sub-types:</b>	None

<b>Name</b>	EquityOptionTransactionSupplement
<b>Used by (from the same schema document)</b>	Element <a href="#">equityOptionTransactionSupplement</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type for defining equity option transaction supplements.

### XML Instance Representation

```

<...
  id=" xsd:ID [0..1]">
  <productType> ProductType </productType> [0..*]
    'A classification of the type of product. FpML defines a simple product categorization using
     a coding scheme.'
  <productId> ProductId </productId> [0..*]
    'A product 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.'
  <buyerPartyReference> PartyReference </buyerPartyReference> [1]
    'A reference to the party that buys this instrument, i.e. pays for this instrument and
     receives the rights defined by it. See 2000 ISDA definitions Article 11.1 (b). In the case
     of FRAs this is the fixed rate payer.'
  <buyerAccountReference> AccountReference </buyerAccountReference> [0..1]
    'A reference to the account that buys this instrument.'
  <sellerPartyReference> PartyReference </sellerPartyReference> [1]
    'A reference to the party that sells ("writes") this instrument, i.e. that grants the
     rights defined by this instrument and in return receives a payment for it. See 2000
     ISDA definitions Article 11.1 (a). In the case of FRAs this is the floating rate payer.'
  <sellerAccountReference> AccountReference </sellerAccountReference> [0..1]
    'A reference to the account that sells this instrument.'
  <optionType> EquityOptionTypeEnum </optionType> [1]
    'The type of option transaction.'
  <equityEffectiveDate> xsd:date </equityEffectiveDate> [0..1]
    'Effective date for a forward starting option.'
  <underlyer> Underlyer </underlyer> [1]
    'Specifies the underlying component, which can be either one or many and consists in
     either equity, index or convertible bond component, or a combination of these.'
  <notional> NonNegativeMoney </notional> [0..1]
    'The notional amount.'
  <equityExercise> EquityExerciseValuationSettlement </equityExercise> [1]
    'The parameters for defining how the equity option can be exercised, how it is valued and
     how it is settled.'

```

```

Start Group: Feature.model [0..1]
<feature> OptionFeatures </feature> [0..1]
'Asian, Barrier, Knock and Pass Through features.'

<fxFeature> FxFeature </fxFeature> [0..1]
'Quanto, Composite, or Cross Currency FX features.'

End Group: Feature.model
<strategyFeature> StrategyFeature </strategyFeature> [0..1]
'A equity option simple strategy feature.'

<strike> EquityStrike </strike> [1]
'Defines whether it is a price or level at which the option has been, or will be, struck.'

<spotPrice> NonNegativeDecimal </spotPrice> [0..1]
'The price per share, index or basket observed on the trade or effective date.'

<numberOfOptions> NonNegativeDecimal </numberOfOptions> [1]
'The number of options comprised in the option transaction.'

<equityPremium> EquityPremium </equityPremium> [1]
'The equity option premium payable by the buyer to the seller.'

<exchangeLookAlike> xsd:boolean </exchangeLookAlike> [0..1]
'For a share option transaction, a flag used to indicate whether the transaction is to be treated as an \'exchange look-alike\'. This designation has significance for how share adjustments (arising from corporate actions) will be determined for the transaction. For an \'exchange look-alike\' transaction the relevant share adjustments will follow that for a corresponding designated contract listed on the related exchange (referred to as Options Exchange Adjustment (ISDA defined term)), otherwise the share adjustments will be determined by the calculation agent (referred to as Calculation Agent Adjustment (ISDA defined term)).'

<exchangeTradedContractNearest> xsd:boolean </exchangeTradedContractNearest> [0..1]
'For an index option transaction, a flag used in conjunction with Futures Price Valuation (ISDA defined term) to indicate whether the Nearest Index Contract provision is applicable. The Nearest Index Contract provision is a rule for determining the Exchange-traded Contract (ISDA defined term) without having to explicitly state the actual contract, delivery month and exchange on which it is traded.'

```

Start Group: IndexAnnexFallback.model [0..1]

Start Choice [1]

- <multipleExchangeIndexAnnexFallback> xsd:boolean </multipleExchangeIndexAnnexFallback> [1]
 'For an index option transaction, a flag to indicate whether a relevant Multiple Exchange Index Annex is applicable to the transaction. This annex defines additional provisions which are applicable where an index is comprised of component securities that are traded on multiple exchanges.'
- <componentSecurityIndexAnnexFallback> xsd:boolean </componentSecurityIndexAnnexFallback> [1]
 'For an index option transaction, a flag to indicate whether a relevant Component Security Index Annex is applicable to the transaction.'

End Choice

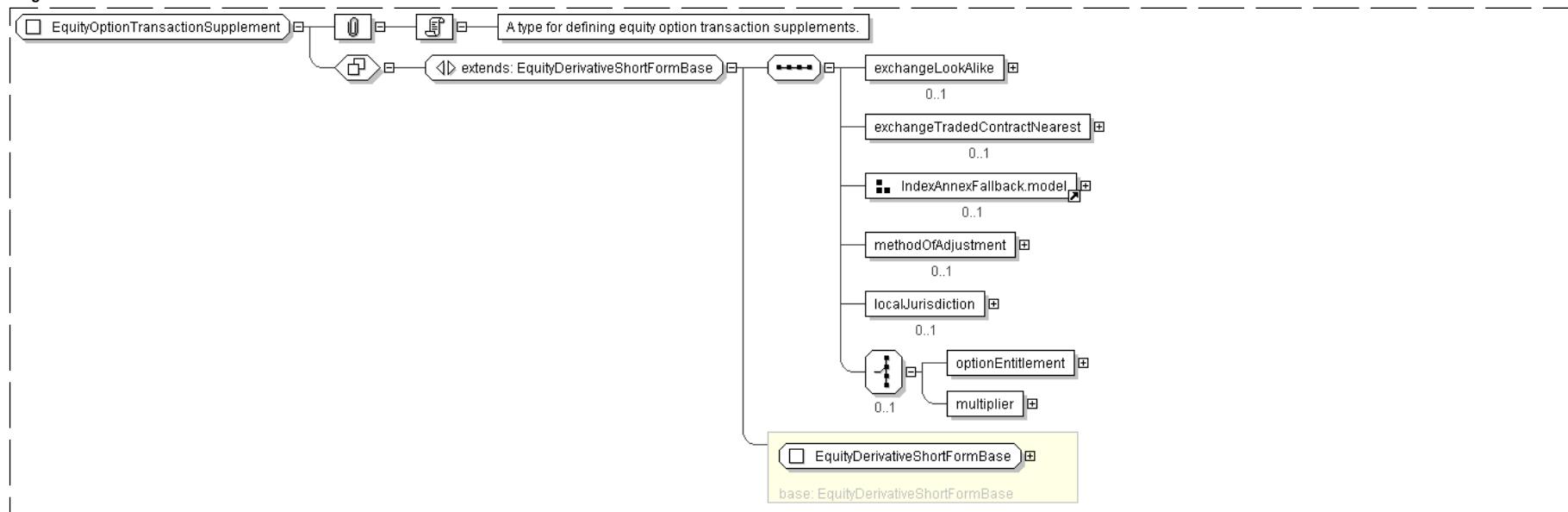
End Group: IndexAnnexFallback.model

- <methodOfAdjustment> MethodOfAdjustmentEnum </methodOfAdjustment> [0..1]
 <localJurisdiction> CountryCode </localJurisdiction> [0..1]
 'Local Jurisdiction is a term used in the AEJ Master Confirmation, which is used to determine local taxes, which shall mean taxes, duties, and similar charges imposed by the taxing authority of the Local Jurisdiction If this element is not present Local Jurisdiction is Not Applicable.'

Start Choice [0..1]

- <optionEntitlement> PositiveDecimal </optionEntitlement> [1]
 'The number of shares per option comprised in the option transaction supplement.'
- <multiplier> PositiveDecimal </multiplier> [1]
 'Specifies the contract multiplier that can be associated with an index option.'

```
| End Choice
| </...>
```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="EquityOptionTransactionSupplement">
  <xsd:complexContent>
    <xsd:extension base=" EquityDerivativeShortFormBase ">
      <xsd:sequence>
        <xsd:element name="exchangeLookAlike" type=" xsd:boolean " minOccurs="0"/>
        <xsd:element name="exchangeTradedContractNearest" type=" xsd:boolean " minOccurs="0"/>
        <xsd:group ref=" IndexAnnexFallback.model " minOccurs="0"/>
        <xsd:element name="methodOfAdjustment" type=" MethodofAdjustmentEnum " minOccurs="0"/>
        <xsd:element name="localJurisdiction" type=" CountryCode " minOccurs="0"/>
        <xsd:choice minOccurs="0">
          <xsd:element name="optionEntitlement" type=" PositiveDecimal "/>
          <xsd:element name="multiplier" type=" PositiveDecimal "/>
        </xsd:choice>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

top

**Complex Type: PrePayment**

**Super-types:** [PaymentBase](#) < **PrePayment** (by extension)  
**Sub-types:** None

<b>Name</b>	PrePayment
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">EquityExerciseValuationSettlement</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type for defining PrePayment.

**XML Instance Representation**

```
<...
  id=" xsd:ID [0..1]">
```

```

<payerPartyReference> PartyReference </payerPartyReference> [1]
'A reference to the party responsible for making the payments defined by this structure.'

<payerAccountReference> AccountReference </payerAccountReference> [0..1]
'A reference to the account responsible for making the payments defined by this structure.'

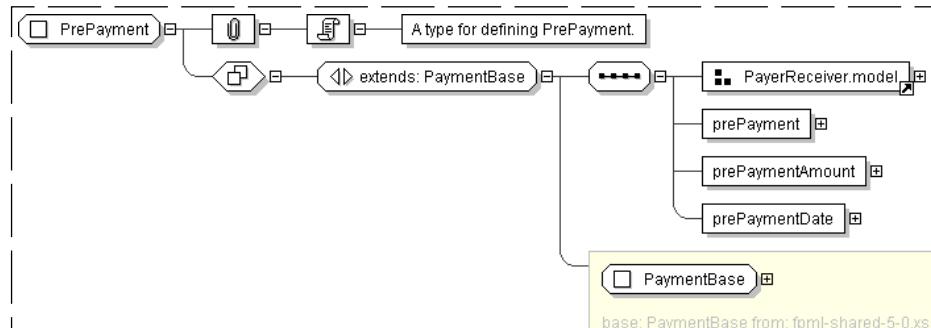
<receiverPartyReference> PartyReference </receiverPartyReference> [1]
'A reference to the party that receives the payments corresponding to this structure.'

<receiverAccountReference> AccountReference </receiverAccountReference> [0..1]
'A reference to the account that receives the payments corresponding to this structure.'

<prePayment> xsd:boolean </prePayment> [1]
<prePaymentAmount> NonNegativeMoney </prePaymentAmount> [1]
<prePaymentDate> AdjustableDate </prePaymentDate> [1]
</...>

```

## Diagram



## Schema Component Representation

```

<xsd:complexType name="PrePayment">
  <xsd:complexContent>
    <xsd:extension base=" PaymentBase ">
      <xsd:sequence>
        <xsd:group ref=" PayerReceiver.model ">
          <xsd:element name="prePayment" type=" xsd:boolean " />
          <xsd:element name="prePaymentAmount" type=" NonNegativeMoney " />
          <xsd:element name="prePaymentDate" type=" AdjustableDate " />
        </xsd:sequence>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>

```

top

Model Group: [EquityExpiration.model](#)

Name	<a href="#">EquityExpiration.model</a>
Used by (from the same schema document)	Complex Type <a href="#">EquityAmericanExercise</a> , Complex Type <a href="#">EquityBermudaExercise</a> , Complex Type <a href="#">EquityEuropeanExercise</a>
Documentation	Choice between expiration expressed as symbolic and optional literal time, or using a determination method.

## XML Instance Representation

```

Start Choice [1]
<equityExpirationTimeType> TimeTypeEnum </equityExpirationTimeType> [1]
'The time of day at which the equity option expires, for example the official closing time
of the exchange.'

<equityExpirationTime> BusinessCenterTime </equityExpirationTime> [0..1]
'The specific time of day at which the equity option expires.'

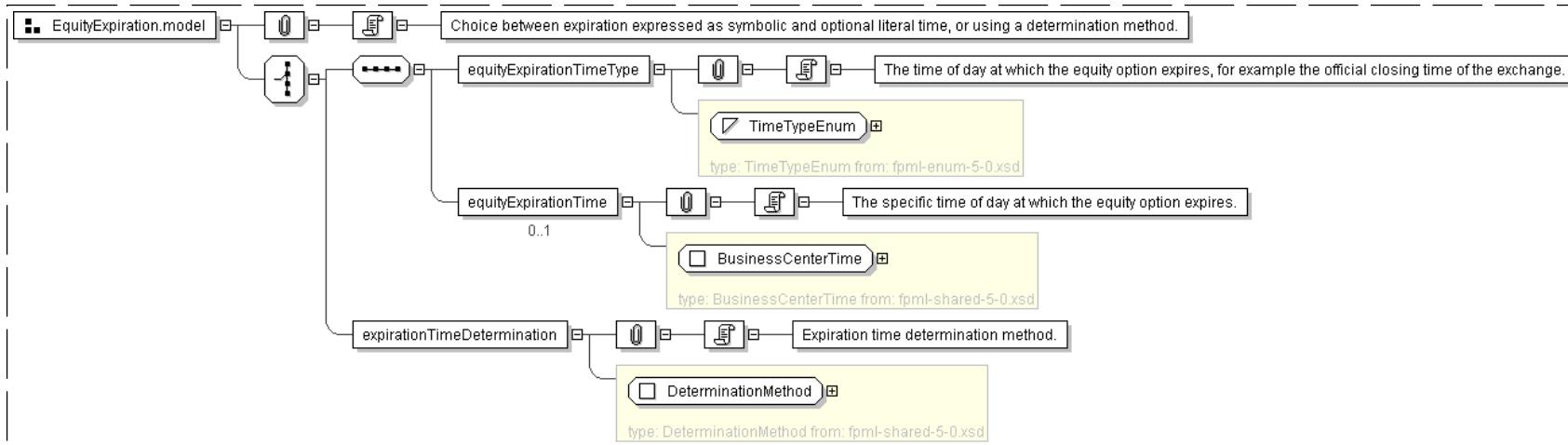
```

```
<expirationTimeDetermination> DeterminationMethod </expirationTimeDetermination> [1]
```

'Expiration time determination method.'

End Choice

#### Diagram



#### Schema Component Representation

```

<xsd:group name="EquityExpiration.model">
  <xsd:choice>
    <xsd:sequence>
      <xsd:element name="equityExpirationTimeType" type="TimeTypeEnum" />
      <xsd:element name="equityExpirationTime" type="BusinessCenterTime" minOccurs="0"/>
    </xsd:sequence>
    <xsd:element name="expirationTimeDetermination" type="DeterminationMethod" />
  </xsd:choice>
</xsd:group>

```

top

#### Legend

**Complex Type:**  
Schema Component Type

**AusAddress**  
Schema Component Name

**Super-types:**  
`Address` < AusAddress (by extension)  
**Sub-types:**  
• `QLDAddress` (by restriction)

If this schema component is a type definition, its type hierarchy is shown in a gray-bordered box.

Name	AusAddress
Abstract	no

The table above displays the properties of this schema component.

#### XML Instance Representation

```

<... country="Australia" >
<unitNo> string </unitNo> [0..1]
<houseNo> string </houseNo> [1]
<street> string </street> [1]
Start Choice [1]
<city> string </city> [1]

```

```

| <town> string </town> [1]
| End Choice
|<state> AusStates </state> [1]
|<postcode> string <><pattern = "[1-9][0-9]{3}>> </postcode> [1]
|</...>

```

The XML Instance Representation table above shows the schema component's content as an XML instance.

- The minimum and maximum occurrence of elements and attributes are provided in square brackets, e.g. [0..1].
- Model group information are shown in gray, e.g. Start Choice ... End Choice.
- For type derivations, the elements and attributes that have been added to or changed from the base type's content are shown in **bold**.
- If an element/attribute has a fixed value, the fixed value is shown in green, e.g. country="Australia".
- Otherwise, the type of the element/attribute is displayed.
  - If the element/attribute's type is in the schema, a link is provided to it.
  - For local simple type definitions, the constraints are displayed in angle brackets, e.g. <><pattern = "[1-9][0-9]{3}>>.

### Schema Component Representation

```

<complexType name="AusAddress">
<complexContent>
<extension base=" Address ">
<sequence>
<element name="state" type=" AusStates " />
<element name="postcode">
<simpleType>
<restriction base=" string " >
<pattern value="[1-9][0-9]{3}" />
</restriction>
</simpleType>
</element>
</sequence>
<attribute name="country" type=" string " fixed="Australia" />
</extension>
</complexContent>
</complexType>

```

The Schema Component Representation table above displays the underlying XML representation of the schema component. (Annotations are not shown.)

[top](#)

## Glossary

**Abstract** (Applies to complex type definitions and element declarations). An abstract element or complex type cannot be used to validate an element instance. If there is a reference to an abstract element, only element declarations that can substitute the abstract element can be used to validate the instance. For references to abstract type definitions, only derived types can be used.

**All Model Group** Child elements can be provided *in any order* in instances. See: <http://www.w3.org/TR/xmlschema-1/#element-all>.

**Choice Model Group** Only one from the list of child elements and model groups can be provided in instances. See: <http://www.w3.org/TR/xmlschema-1/#element-choice>.

**Collapse Whitespace Policy** Replace tab, line feed, and carriage return characters with space character (Unicode character 32). Then, collapse contiguous sequences of space characters into single space character, and remove leading and trailing space characters.

**Disallowed Substitutions** (Applies to element declarations). If *substitution* is specified, then *substitution group* members cannot be used in place of the given element declaration to validate element instances. If *derivation methods*, e.g. extension, restriction, are specified, then the given element declaration will not validate element instances that have types derived from the element declaration's type using the specified derivation methods. Normally, element instances can override their declaration's type by specifying an *xsi:type* attribute.

**Key Constraint** Like [Uniqueness Constraint](#), but additionally requires that the specified value(s) must be provided. See: [http://www.w3.org/TR/xmlschema-1/#clidentity-constraint\\_Definitions](http://www.w3.org/TR/xmlschema-1/#clidentity-constraint_Definitions).

**Key Reference Constraint** Ensures that the specified value(s) must match value(s) from a [Key Constraint](#) or [Uniqueness Constraint](#). See: [http://www.w3.org/TR/xmlschema-1/#clidentity-constraint\\_Definitions](http://www.w3.org/TR/xmlschema-1/#clidentity-constraint_Definitions).

**Model Group** Groups together element content, specifying the order in which the element content can occur and the number of times the group of element content may be repeated. See: [http://www.w3.org/TR/xmlschema-1/#Model\\_Groups](http://www.w3.org/TR/xmlschema-1/#Model_Groups).

**Nillable** (Applies to element declarations). If an element declaration is nillable, instances can use the *xsi:nil* attribute. The *xsi:nil* attribute is

the boolean attribute, *nil*, from the <http://www.w3.org/2001/XMLSchema-instance> namespace. If an element instance has an *xsi:nil* attribute set to true, it can be left empty, even though its element declaration may have required content.

**Notation** A notation is used to identify the format of a piece of data. Values of elements and attributes that are of type, NOTATION, must come from the names of declared notations. See: [http://www.w3.org/TR/xmlschema-1/#cNotation\\_Declarations](http://www.w3.org/TR/xmlschema-1/#cNotation_Declarations).

**Preserve Whitespace Policy** Preserve whitespaces exactly as they appear in instances.

**Prohibited Derivations** (Applies to type definitions). Derivation methods that cannot be used to create sub-types from a given type definition.

**Prohibited Substitutions** (Applies to complex type definitions). Prevents sub-types that have been derived using the specified derivation methods from validating element instances in place of the given type definition.

**Replace Whitespace Policy** Replace tab, line feed, and carriage return characters with space character (Unicode character 32).

**Sequence Model Group** Child elements and model groups must be provided *in the specified order* in instances. See: <http://www.w3.org/TR/xmlschema-1/#element-sequence>.

**Substitution Group** Elements that are *members* of a substitution group can be used wherever the *head* element of the substitution group is referenced.

**Substitution Group Exclusions** (Applies to element declarations). Prohibits element declarations from nominating themselves as being able to substitute a given element declaration, if they have types that are derived from the original element's type using the specified derivation methods.

**Target Namespace** The target namespace identifies the namespace that components in this schema belongs to. If no target namespace is provided, then the schema components do not belong to any namespace.

**Uniqueness Constraint** Ensures uniqueness of an element/attribute value, or a combination of values, within a specified scope. See: [http://www.w3.org/TR/xmlschema-1/#clidentity-constraint\\_Definitions](http://www.w3.org/TR/xmlschema-1/#clidentity-constraint_Definitions).

[top](#)

---

Generated by [oXygen XML Editor](#) using a modified version of [xs3p](#) that adds schema diagrams and chunking support.

# XML Schema Documentation

## Table of Contents

- [Schema Document Properties](#)
- [Global Declarations](#)
  - [Element: interestLeg](#)
  - [Element: returnLeg](#)
  - [Element: returnSwap](#)
  - [Element: returnSwapLeg](#)
- [Global Definitions](#)
  - [Complex Type: AdditionalDisruptionEvents](#)
  - [Complex Type: AdditionalPaymentAmount](#)
  - [Complex Type: AdjustableDateOrRelativeDateSequence](#)
  - [Complex Type: BoundedCorrelation](#)
  - [Complex Type: BoundedVariance](#)
  - [Complex Type: CalculatedAmount](#)
  - [Complex Type: CalculationFromObservation](#)
  - [Complex Type: Compounding](#)
  - [Complex Type: CompoundingRate](#)
  - [Complex Type: Correlation](#)
  - [Complex Type: DirectionalLeg](#)
  - [Complex Type: DirectionalLegUnderlyer](#)
  - [Complex Type: DirectionalLegUnderlyerValuation](#)
  - [Complex Type: DividendAdjustment](#)
  - [Complex Type: DividendConditions](#)
  - [Complex Type: DividendPaymentDate](#)
  - [Complex Type: DividendPeriod](#)
  - [Complex Type: DividendPeriodDividend](#)
  - [Complex Type: EquityCorporateEvents](#)
  - [Complex Type: EquityPremium](#)
  - [Complex Type: EquityStrike](#)
  - [Complex Type: EquityValuation](#)
  - [Complex Type: ExtraordinaryEvents](#)
  - [Complex Type: FloatingRateCalculationReference](#)
  - [Complex Type: IndexAdjustmentEvents](#)
  - [Complex Type: InterestCalculation](#)
  - [Complex Type: InterestLeg](#)
  - [Complex Type: InterestLegCalculationPeriodDates](#)
  - [Complex Type: InterestLegCalculationPeriodDatesReference](#)
  - [Complex Type: InterestLegResetDates](#)
  - [Complex Type: LegAmount](#)
  - [Complex Type: LegId](#)
  - [Complex Type: LegIdentifier](#)
  - [Complex Type: MakeWholeProvisions](#)
  - [Complex Type: NettedSwapBase](#)
  - [Complex Type: OptionFeatures](#)
  - [Complex Type: PrincipalExchangeAmount](#)
  - [Complex Type: PrincipalExchangeDescriptions](#)
  - [Complex Type: PrincipalExchangeFeatures](#)
  - [Complex Type: Representations](#)
  - [Complex Type: Return](#)
  - [Complex Type: ReturnLeg](#)
  - [Complex Type: ReturnLegValuation](#)
  - [Complex Type: ReturnLegValuationPrice](#)
  - [Complex Type: ReturnSwap](#)
  - [Complex Type: ReturnSwapAdditionalPayment](#)
  - [Complex Type: ReturnSwapAmount](#)
  - [Complex Type: ReturnSwapBase](#)
  - [Complex Type: ReturnSwapEarlyTermination](#)
  - [Complex Type: ReturnSwapLegUnderlyer](#)
  - [Complex Type: ReturnSwapNotional](#)
  - [Complex Type: ReturnSwapPaymentDates](#)

- Complex Type: [StartingDate](#)
- Complex Type: [StubCalculationPeriod](#)
- Complex Type: [Variance](#)
- Model Group: [CurrencyAndDeterminationMethod.model](#)
- Model Group: [DeclaredCashAndCashEquivalentDividendPercentage.model](#)
- Model Group: [Dividends.model](#)
- Model Group: [EquityUnderlyerProvisions.model](#)
- Model Group: [Feature.model](#)
- Model Group: [IndexAnnexFallback.model](#)
- Model Group: [MutualOrOptionalEarlyTermination.model](#)

- [Legend](#)

- [Glossary](#)

[top](#)

## Schema Document Properties

<b>Target Namespace</b>	<a href="http://www.fpml.org/FpML-5/confirmation">http://www.fpml.org/FpML-5/confirmation</a>
<b>Version</b>	\$Revision: 2654 \$
<b>Element and Attribute Namespaces</b>	<ul style="list-style-type: none"> <li>◦ Global element and attribute declarations belong to this schema's target namespace.</li> <li>◦ By default, local element declarations belong to this schema's target namespace.</li> <li>◦ By default, local attribute declarations have no namespace.</li> </ul>
<b>Schema Composition</b>	<ul style="list-style-type: none"> <li>◦ This schema includes components from the following schema document(s):           <ul style="list-style-type: none"> <li>◦ <a href="#">fpml-option-shared-5-0.xsd</a></li> </ul> </li> </ul>

## Declared Namespaces

Prefix	Namespace
Default namespace	<a href="http://www.fpml.org/FpML-5/confirmation">http://www.fpml.org/FpML-5/confirmation</a>
xml	<a href="http://www.w3.org/XML/1998/namespace">http://www.w3.org/XML/1998/namespace</a>
fpml-annotation	<a href="http://www.fpml.org/annotation">http://www.fpml.org/annotation</a>
xsd	<a href="http://www.w3.org/2001/XMLSchema">http://www.w3.org/2001/XMLSchema</a>
ecore	<a href="http://www.eclipse.org/emf/2002/Ecore">http://www.eclipse.org/emf/2002/Ecore</a>

## Schema Component Representation

```

<xsd:schema targetNamespace="http://www.fpml.org/FpML-5/confirmation"
  documentRoot="FpML" nsPrefix="conf" package="org.fpml.confirmation" version="$Revision: 2654 "
  $ elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xsd:include schemaLocation="fpml-option-shared-5-0.xsd"/>
  ...
</xsd:schema>
  
```

[top](#)

## Global Declarations

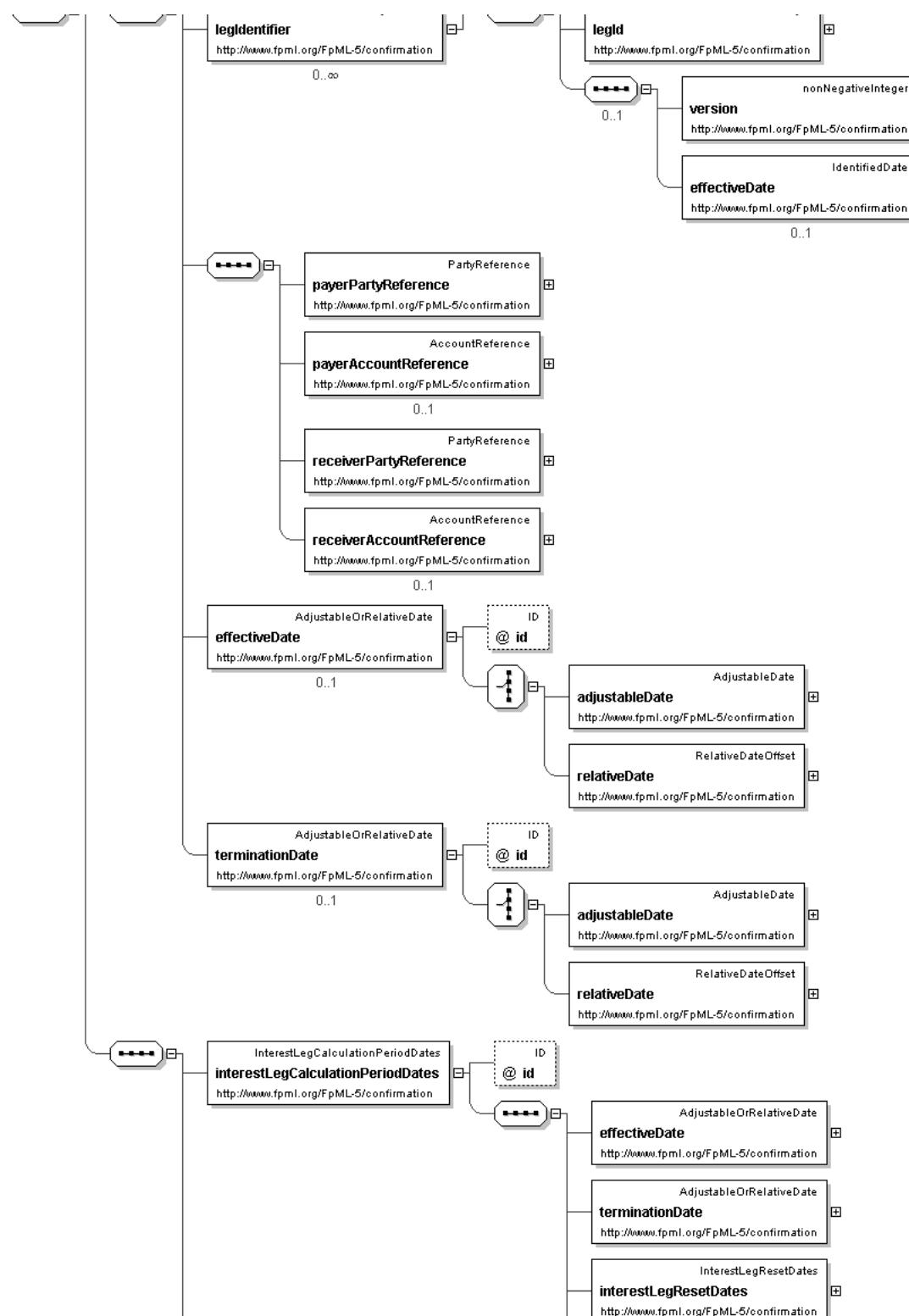
### Element: interestLeg

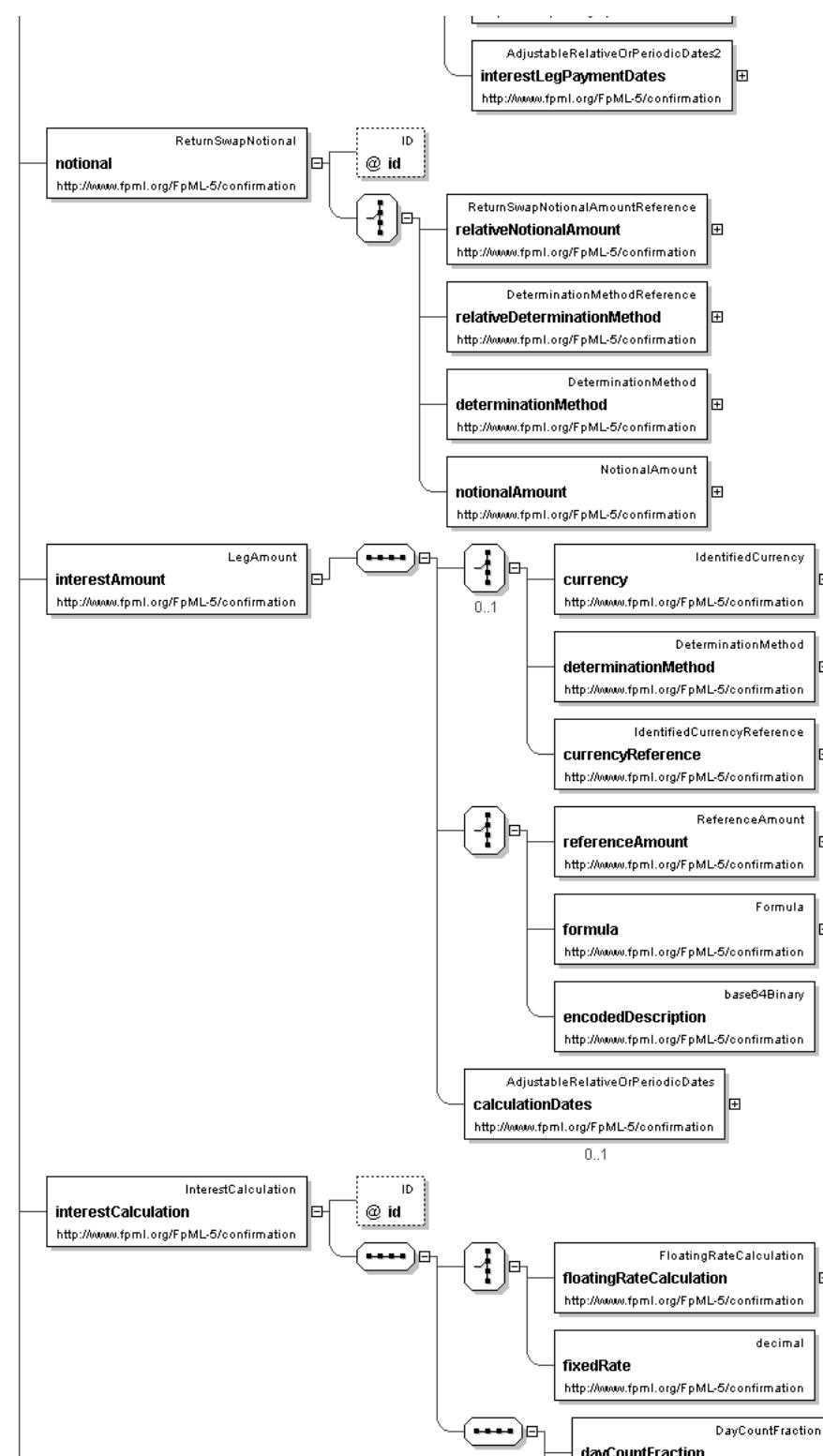
- This element can be used wherever the following element is referenced:
  - [returnSwapLeg](#)

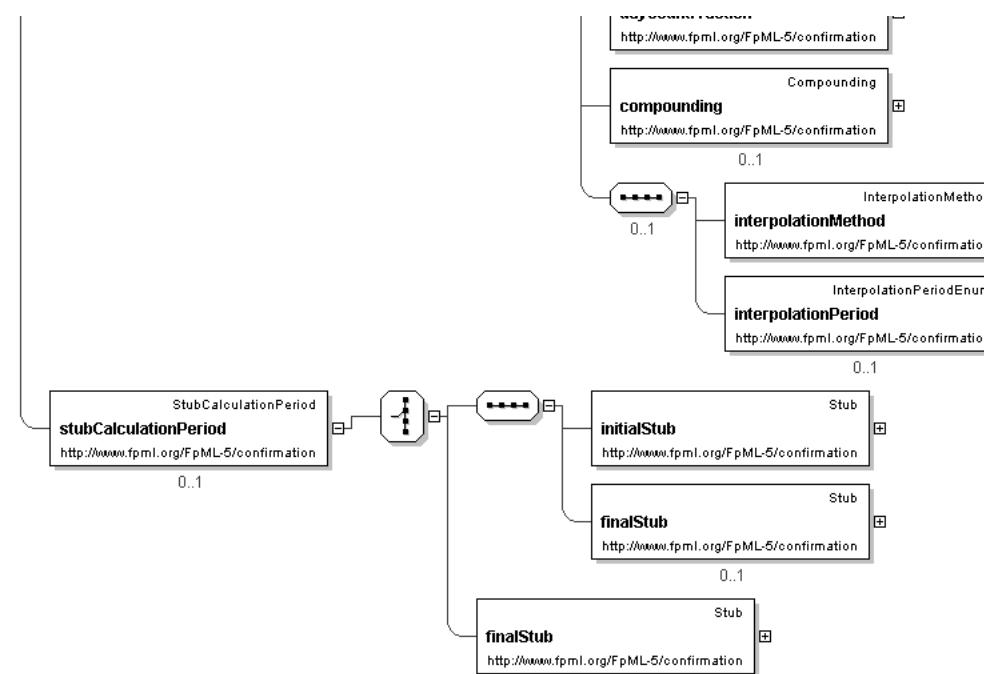
<b>Name</b>	interestLeg
<b>Type</b>	<a href="#">InterestLeg</a>
<b>Nillable</b>	no
<b>Abstract</b>	no
<b>Documentation</b>	The fixed income amounts of the return type swap.

## Logical Diagram







**XML Instance Representation**

```

<interestLeg
id=" xsd:ID [0..1]">
  <legIdentifier> LegIdentifier </legIdentifier> [0..*]
  'Version aware identification of this leg.'

  <payerPartyReference> PartyReference </payerPartyReference> [1]
  'A reference to the party responsible for making the payments defined by this structure.'

  <payerAccountReference> AccountReference </payerAccountReference> [0..1]
  'A reference to the account responsible for making the payments defined by this structure.'

  <receiverPartyReference> PartyReference </receiverPartyReference> [1]
  'A reference to the party that receives the payments corresponding to this structure.'

  <receiverAccountReference> AccountReference </receiverAccountReference> [0..1]
  'A reference to the account that receives the payments corresponding to this structure.'

  <effectiveDate> AdjustableOrRelativeDate </effectiveDate> [0..1]
  'Specifies the effective date of this leg of the swap. When defined in relation to a date specified somewhere else in the document (through the relativeDate component), this element will typically point to the effective date of the other leg of the swap.'

  <terminationDate> AdjustableOrRelativeDate </terminationDate> [0..1]
  'Specifies the termination date of this leg of the swap. When defined in relation to a date specified somewhere else in the document (through the relativeDate component), this element will typically point to the termination date of the other leg of the swap.'

  <interestLegCalculationPeriodDates> InterestLegCalculationPeriodDates
  </interestLegCalculationPeriodDates> [1]
  'Component that holds the various dates used to specify the interest leg of the equity swap. It is used to define the InterestPeriodDates identifier.'

  <notional> ReturnSwapNotional </notional> [1]
  'Specifies the notional of a return type swap. When used in the equity leg, the definition
  
```

will typically combine the actual amount (using the notional component defined by the FpML industry group) and the determination method. When used in the interest leg, the definition will typically point to the definition of the equity leg.'

<interestAmount> LegAmount </interestAmount> [1]

'Specifies, in relation to each Interest Payment Date, the amount to which the Interest Payment Date relates. Unless otherwise specified, this term has the meaning defined in the ISDA 2000 ISDA Definitions.'

<interestCalculation> InterestCalculation </interestCalculation> [1]

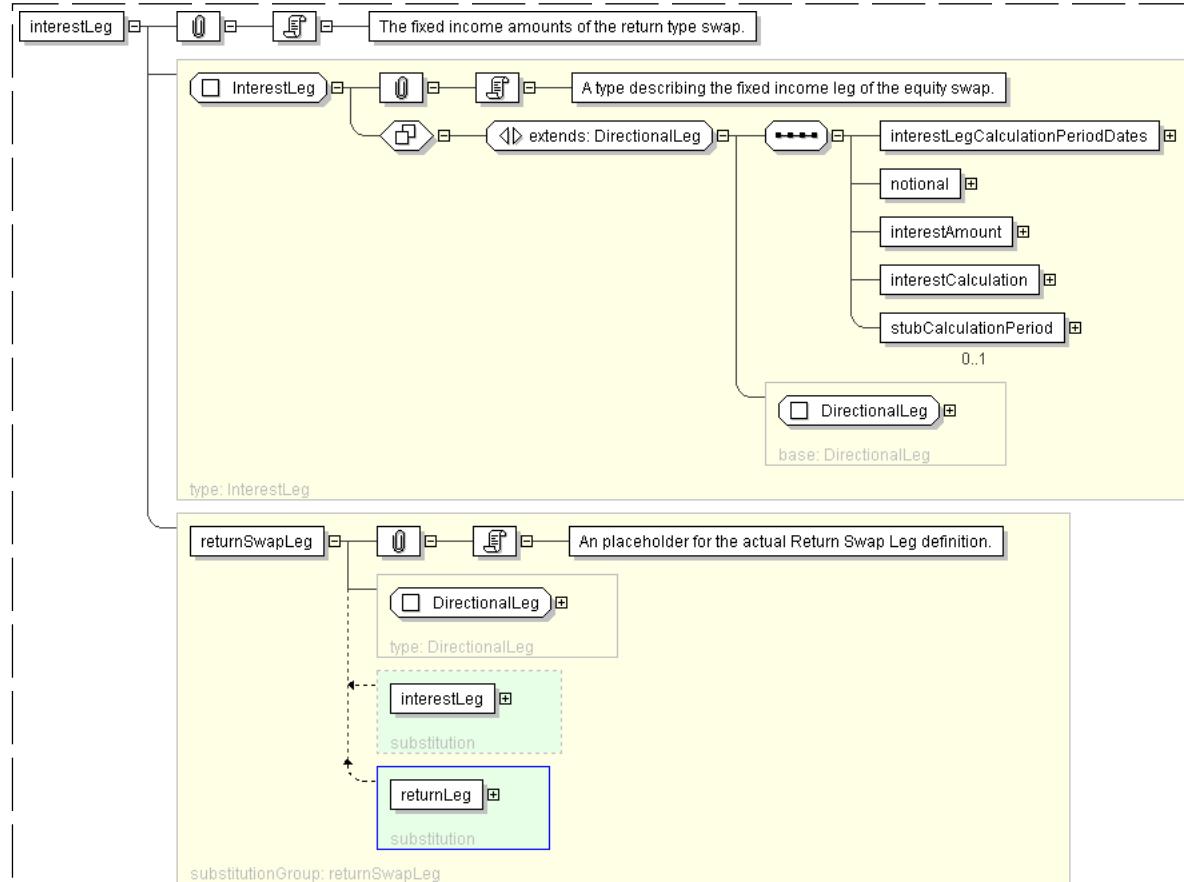
'Specifies the calculation method of the interest rate leg of the equity swap. Includes the floating or fixed rate calculation definitions, along with the determination of the day count fraction.'

<stubCalculationPeriod> StubCalculationPeriod </stubCalculationPeriod> [0..1]

'Specifies the stub calculation period.'

</interestLeg>

#### Diagram



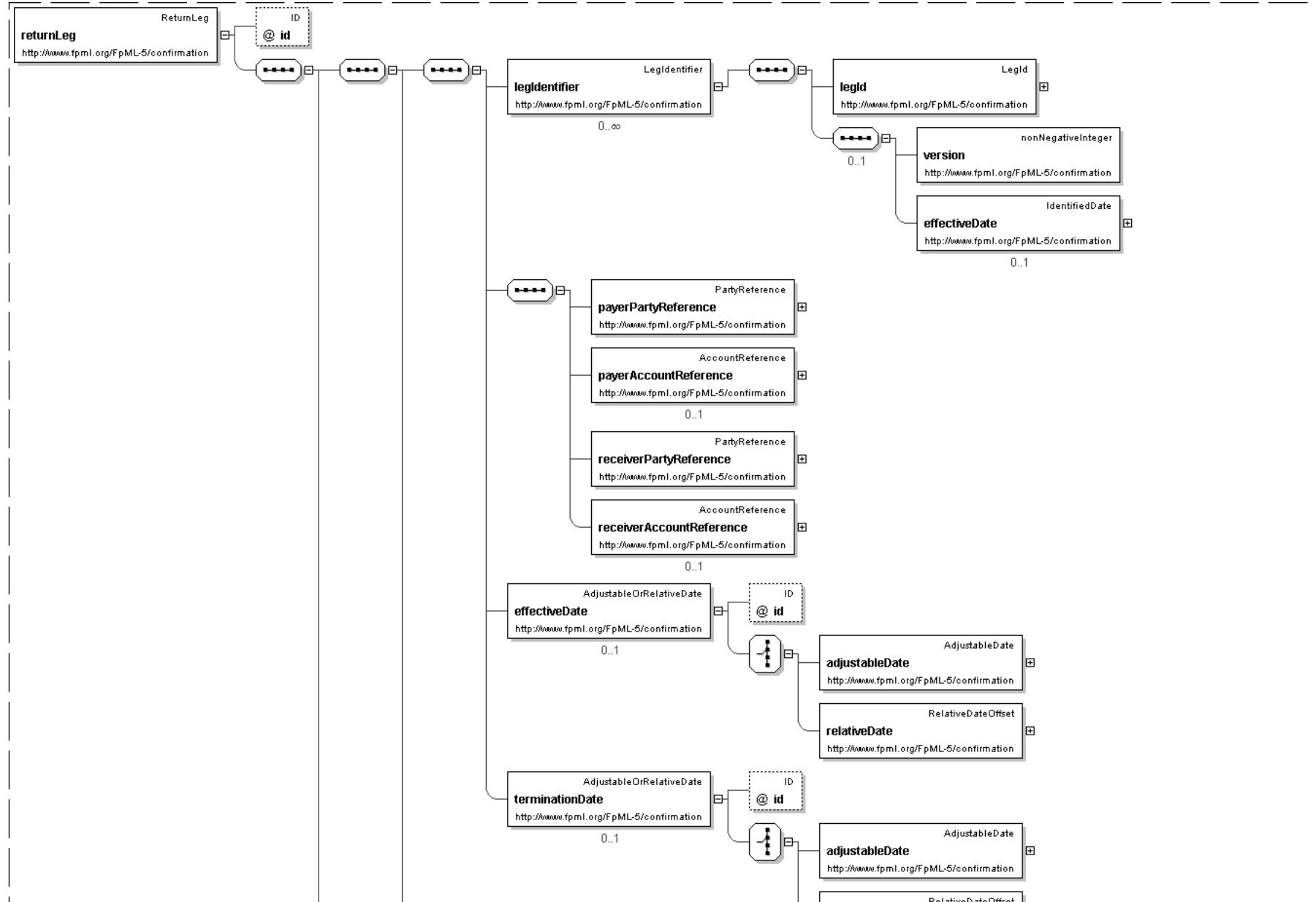
#### Schema Component Representation

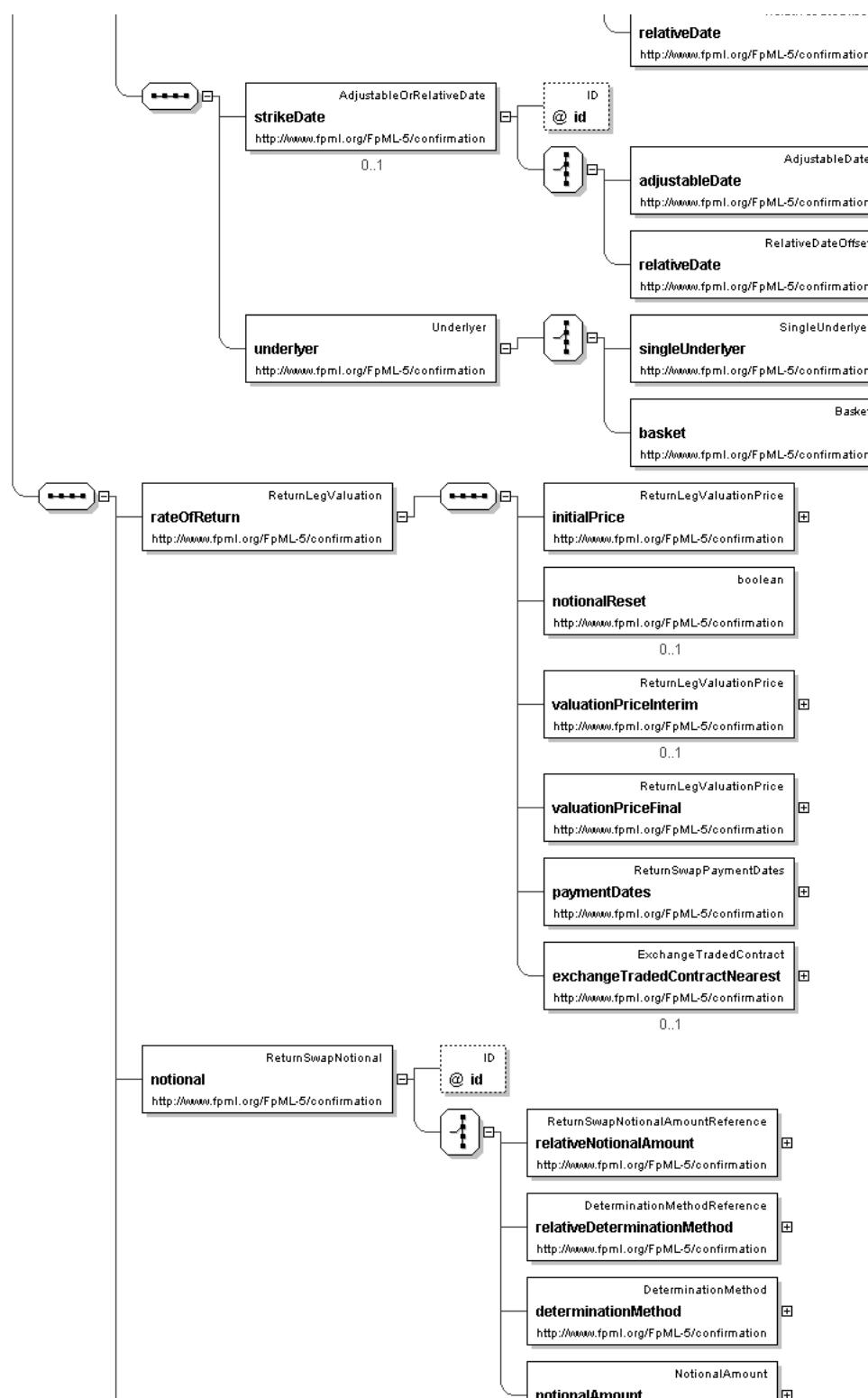
```
<xsd:element name="interestLeg" type="#_InterestLeg" substitutionGroup="returnSwapLeg"/>
```

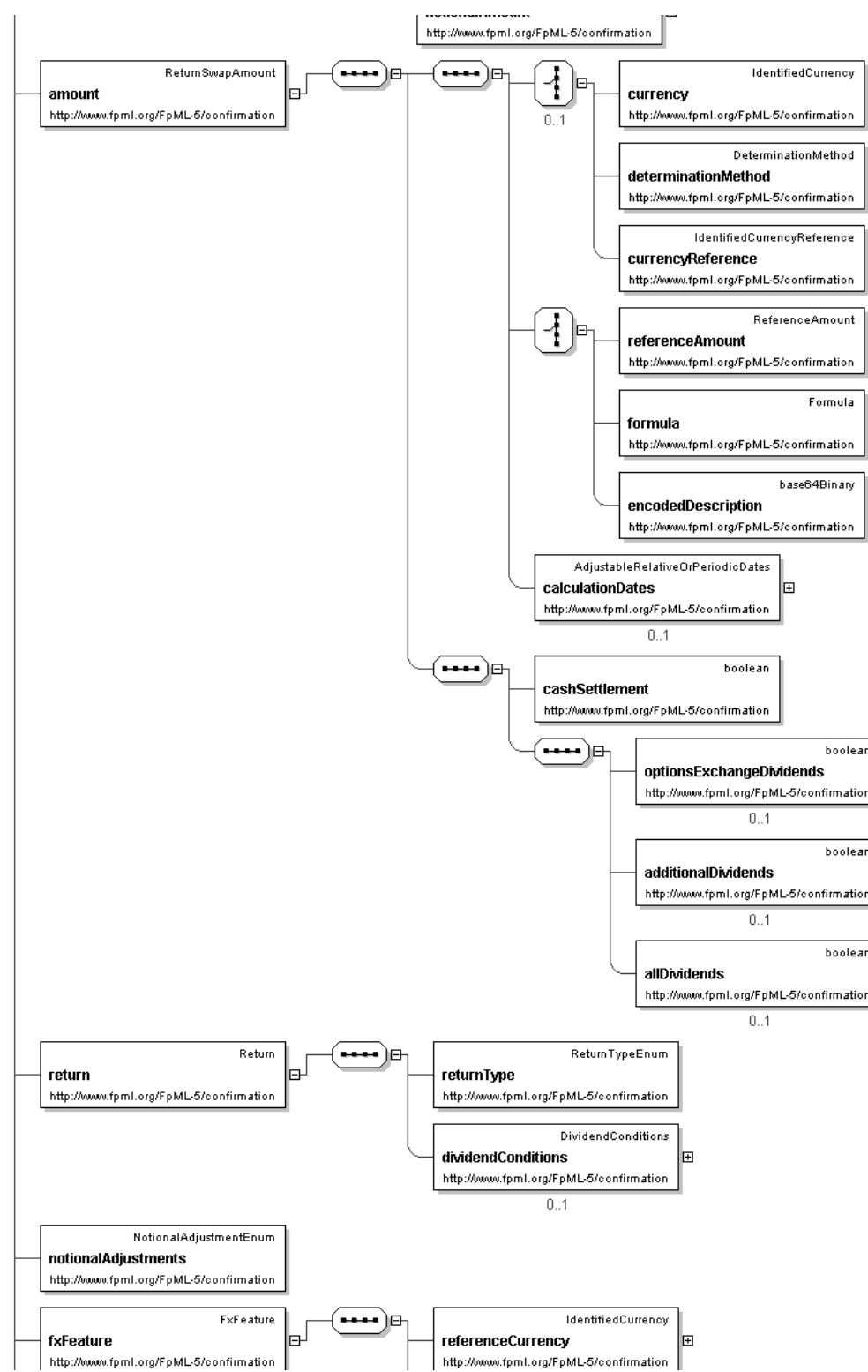
**Element: returnLeg**

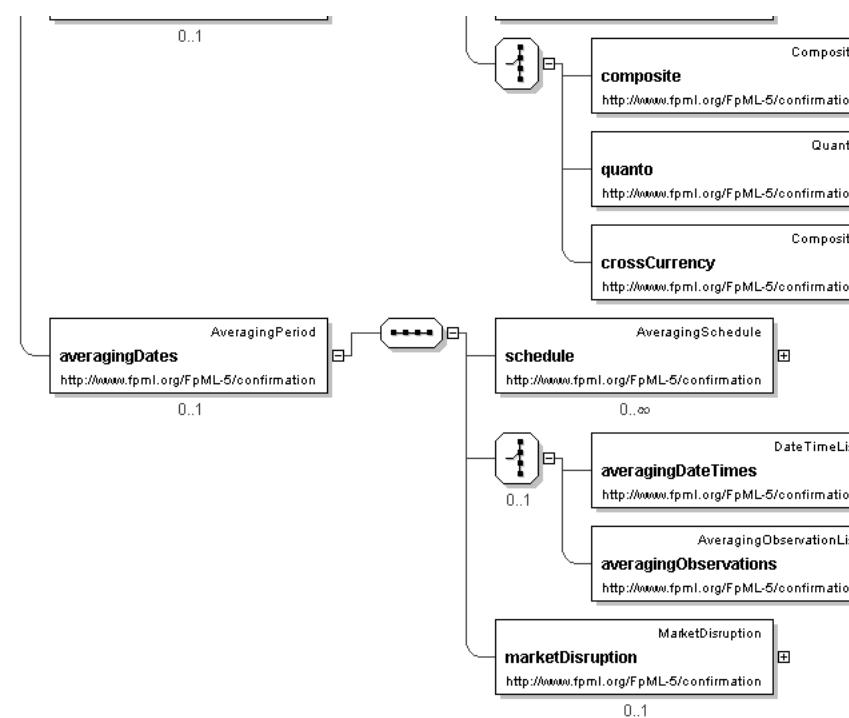
- This element can be used wherever the following element is referenced:
  - [returnSwapLeg](#)

Name	returnLeg
Type	<a href="#">ReturnLeg</a>
Nillable	no
Abstract	no
Documentation	Return amounts of the return type swap.

**Logical Diagram**





**XML Instance Representation**

```

<returnLeg
id=" xsd:ID [0..1]">
  <legIdentifier> LegIdentifier </legIdentifier> [0..*]
  'Version aware identification of this leg.'

  <payerPartyReference> PartyReference </payerPartyReference> [1]
  'A reference to the party responsible for making the payments defined by this structure.'

  <payerAccountReference> AccountReference </payerAccountReference> [0..1]
  'A reference to the account responsible for making the payments defined by this structure.'

  <receiverPartyReference> PartyReference </receiverPartyReference> [1]
  'A reference to the party that receives the payments corresponding to this structure.'

  <receiverAccountReference> AccountReference </receiverAccountReference> [0..1]
  'A reference to the account that receives the payments corresponding to this structure.'

  <effectiveDate> AdjustableOrRelativeDate </effectiveDate> [0..1]
  'Specifies the effective date of this leg of the swap. When defined in relation to a date specified somewhere else in the document (through the relativeDate component), this element will typically point to the effective date of the other leg of the swap.'

  <terminationDate> AdjustableOrRelativeDate </terminationDate> [0..1]
  'Specifies the termination date of this leg of the swap. When defined in relation to a date specified somewhere else in the document (through the relativeDate component), this element will typically point to the termination date of the other leg of the swap.'

  <strikeDate> AdjustableOrRelativeDate </strikeDate> [0..1]
  'Specifies the strike date of this leg of the swap, used for forward starting swaps. When defined in relation to a date specified somewhere else in the document (through the relativeDate component), this element will typically be relative to the trade date of the swap.'

```

<underlyer> Underlyer </underlyer> [1]  
*'Specifies the underlying component of the leg, which can be either one or many and consists in either equity, index or convertible bond component, or a combination of these.'*

<rateOfReturn> ReturnLegValuation </rateOfReturn> [1]  
*'Specifies the terms of the initial price of the return type swap and of the subsequent valuations of the underlyer.'*

<notional> ReturnSwapNotional </notional> [1]  
*'Specifies the notional of a return type swap. When used in the equity leg, the definition will typically combine the actual amount (using the notional component defined by the FpML industry group) and the determination method. When used in the interest leg, the definition will typically point to the definition of the equity leg.'*

<amount> ReturnSwapAmount </amount> [1]  
*'Specifies, in relation to each Payment Date, the amount to which the Payment Date relates. For return swaps this element is equivalent to the Equity Amount term as defined in the ISDA 2002 Equity Derivatives Definitions.'*

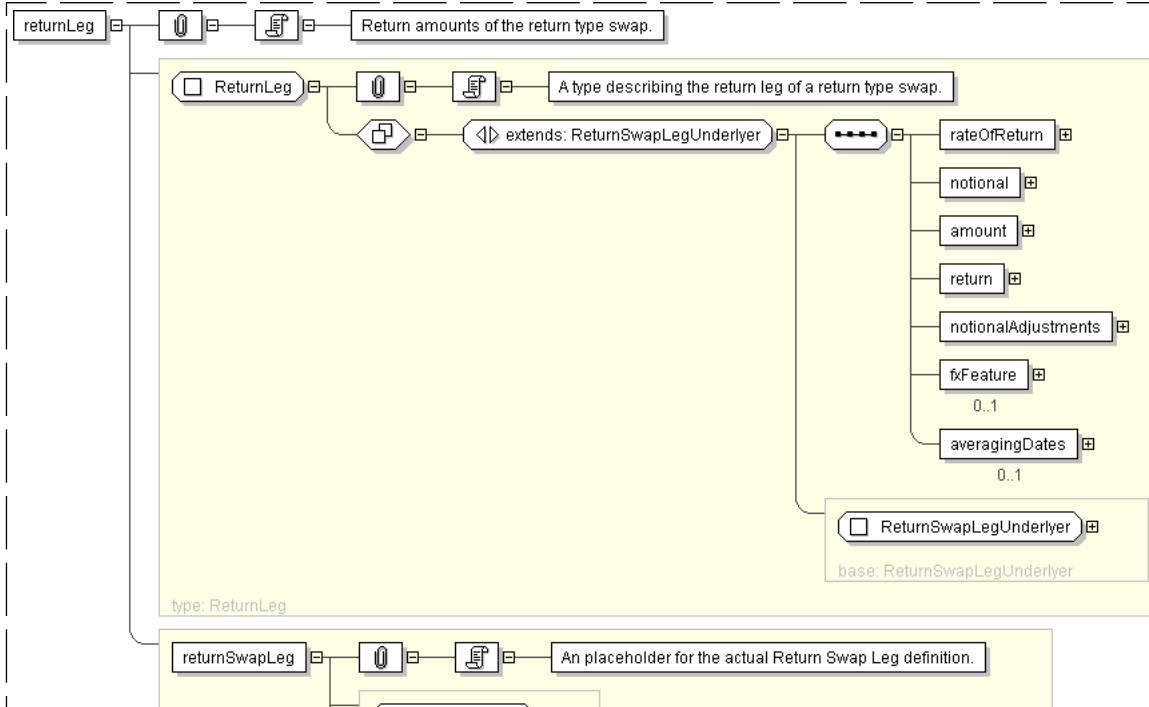
<return> Return </return> [1]  
*'Specifies the conditions under which dividend affecting the underlyer will be paid to the receiver of the amounts.'*

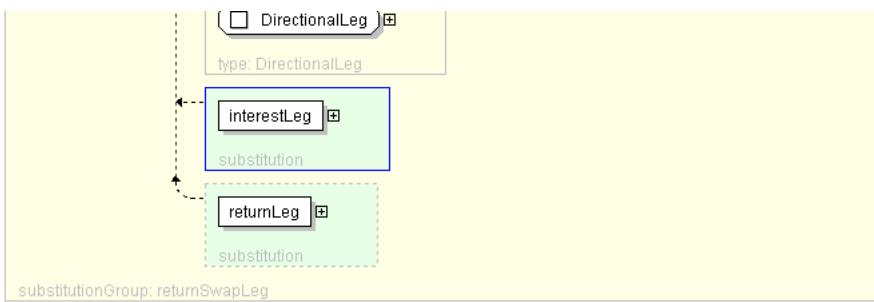
<notionalAdjustments> NotionalAdjustmentEnum </notionalAdjustments> [1]  
*'Specifies the conditions that govern the adjustment to the number of units of the return swap.'*

<fxFeature> FxFeature </fxFeature> [0..1]  
*'A quanto or composite FX feature.'*

<averagingDates> AveragingPeriod </averagingDates> [0..1]  
*'Averaging Dates used in the swap.'*

&lt;/returnLeg&gt;

**Diagram**

**Schema Component Representation**

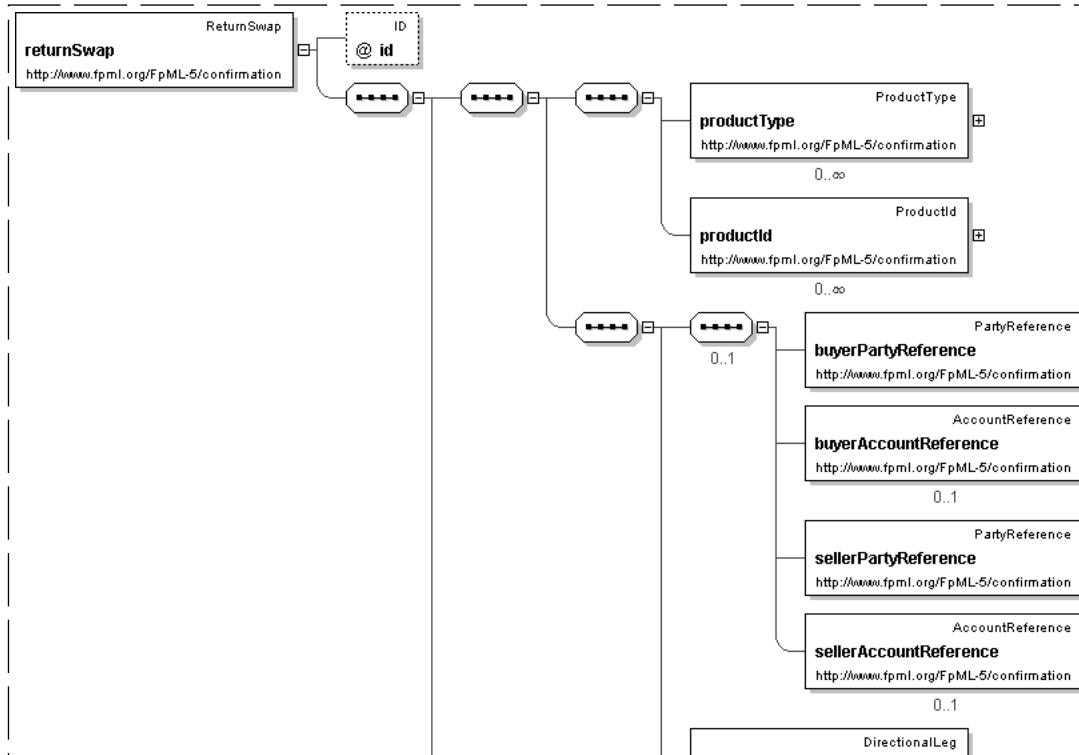
```
<xsd:element name="returnLeg" type="ReturnLeg" substitutionGroup="returnSwapLeg" />
```

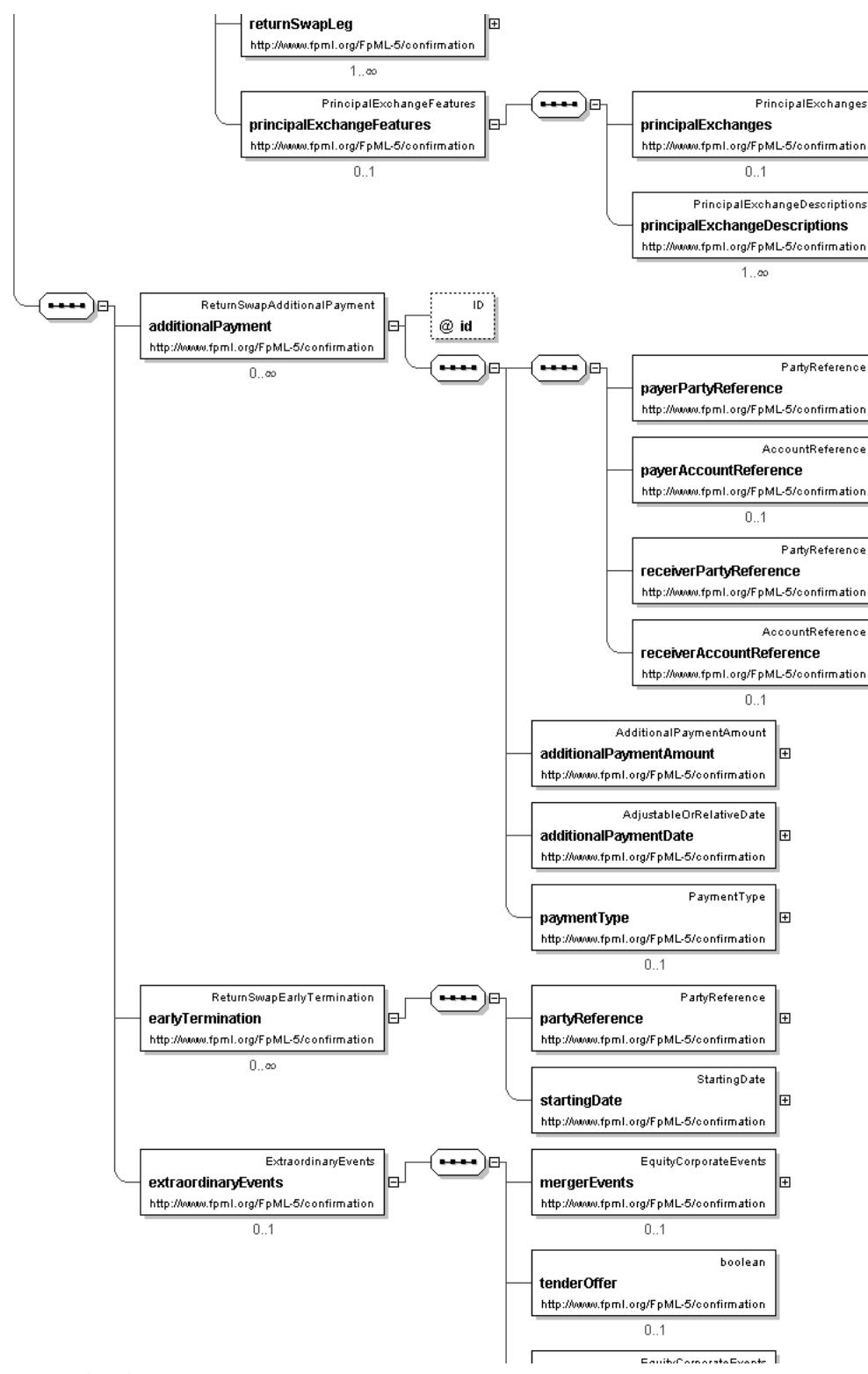
top

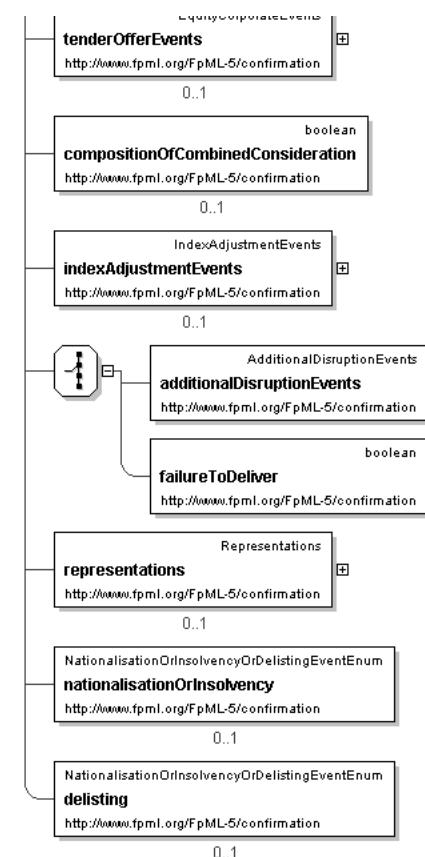
**Element: returnSwap**

- This element can be used wherever the following element is referenced:
  - product

Name	returnSwap
Type	ReturnSwap
Nullable	no
Abstract	no
Documentation	Specifies the structure of a return type swap. It can represent return swaps, total return swaps, variance swaps.

**Logical Diagram**



**XML Instance Representation**

```

<returnSwap
id=" xsd:ID [0..1]">
  <productType> ProductType </productType> [0..*]
  
```

*'A classification of the type of product. FpML defines a simple product categorization using a coding scheme.'*

```

  <productId> ProductId </productId> [0..*]
  
```

*'A product 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.'*

Start Group: BuyerSeller.model [0..1]

*'BuyerSeller.model has been included as an optional child of ReturnSwapBase to support the situation where an implementor wishes to indicate who has manufactured the Swap through representing them as the Seller. It may be removed in future major revisions.'*

```

  <buyerPartyReference> PartyReference </buyerPartyReference> [1]
  
```

*'A reference to the party that buys this instrument, ie. pays for this instrument and receives the rights defined by it. See 2000 ISDA definitions Article 11.1 (b). In the case of FRAs this the fixed rate payer.'*

```

  <buyerAccountReference> AccountReference </buyerAccountReference> [0..1]
  
```

*'A reference to the account that buys this instrument.'*

```

  <sellerPartyReference> PartyReference </sellerPartyReference> [1]
  
```

*'A reference to the party that sells (\"writes\") this instrument, i.e. that grants the*

rights defined by this instrument and in return receives a payment for it. See 2000 ISDA definitions Article 11.1 (a). In the case of FRAs this is the floating rate payer.

```
<sellerAccountReference> AccountReference </sellerAccountReference> [0..1]
  'A reference to the account that sells this instrument.'
```

group: BuyerSeller.model

```
returnSwapLeg> ... </returnSwapLeg> [1..*]
```

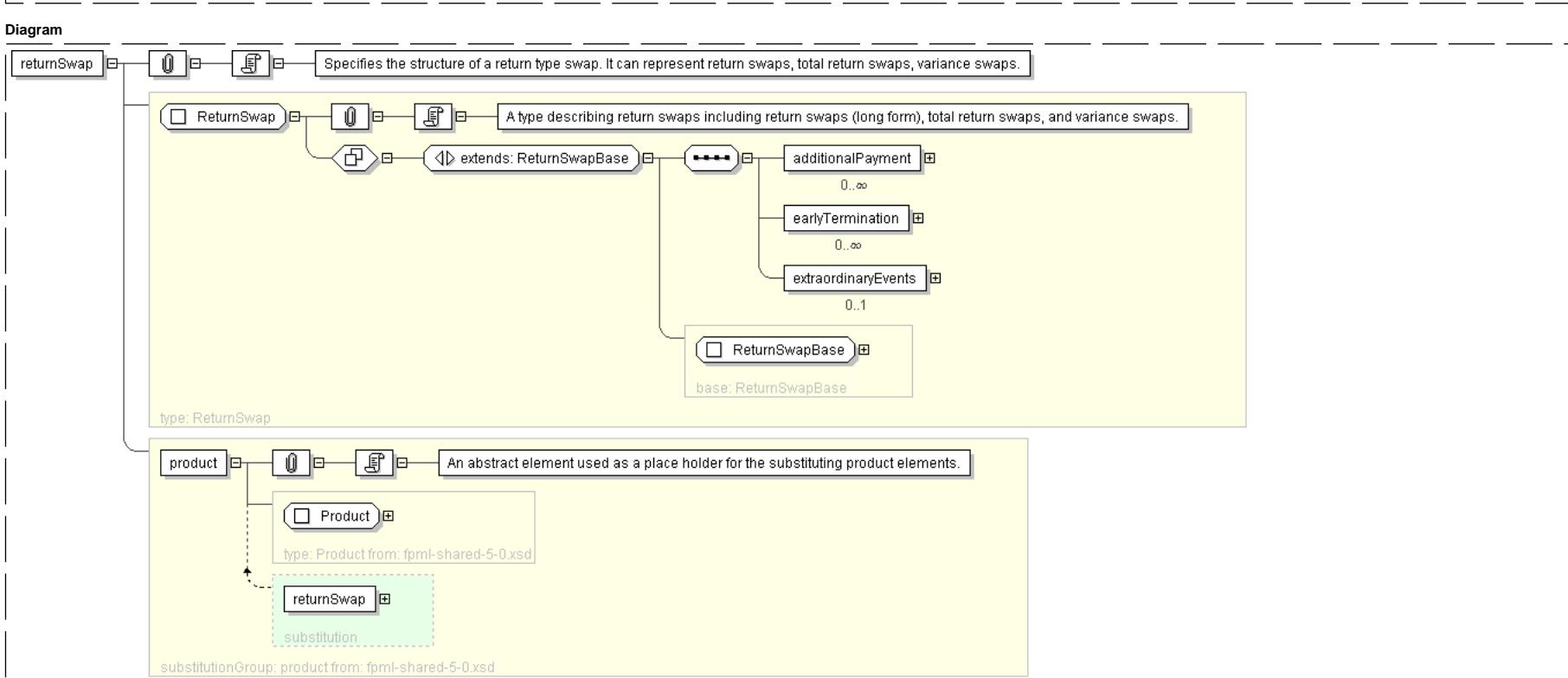
principalExchangeFeatures> PrincipalExchangeFeatures </principalExchangeFeatures> [0..1]

```
<additionalPayment> ReturnSwapAdditionalPayment </additionalPayment> [0..*]  
[ Specifying additional payment(s) between the principal parties to the trade ]
```

`<earlyTermination> ReturnSwapEarlyTermination </earlyTermination> [0..*]`  
*'Specifies, for one or for both the parties to the trade, the date from which it can early terminate it.'*

**<extraordinaryEvents>** **ExtraordinaryEvents** **</extraordinaryEvents>** [0..1]  
*'Where the underlying is shares, specifies events affecting the issuer of those shares that may require the terms of the transaction to be adjusted.'*

```
| </returnSwap>
```



## Schema Component Representation

```
<xsd:element name="returnSwap" type=" ReturnSwap " substitutionGroup="product " />
```

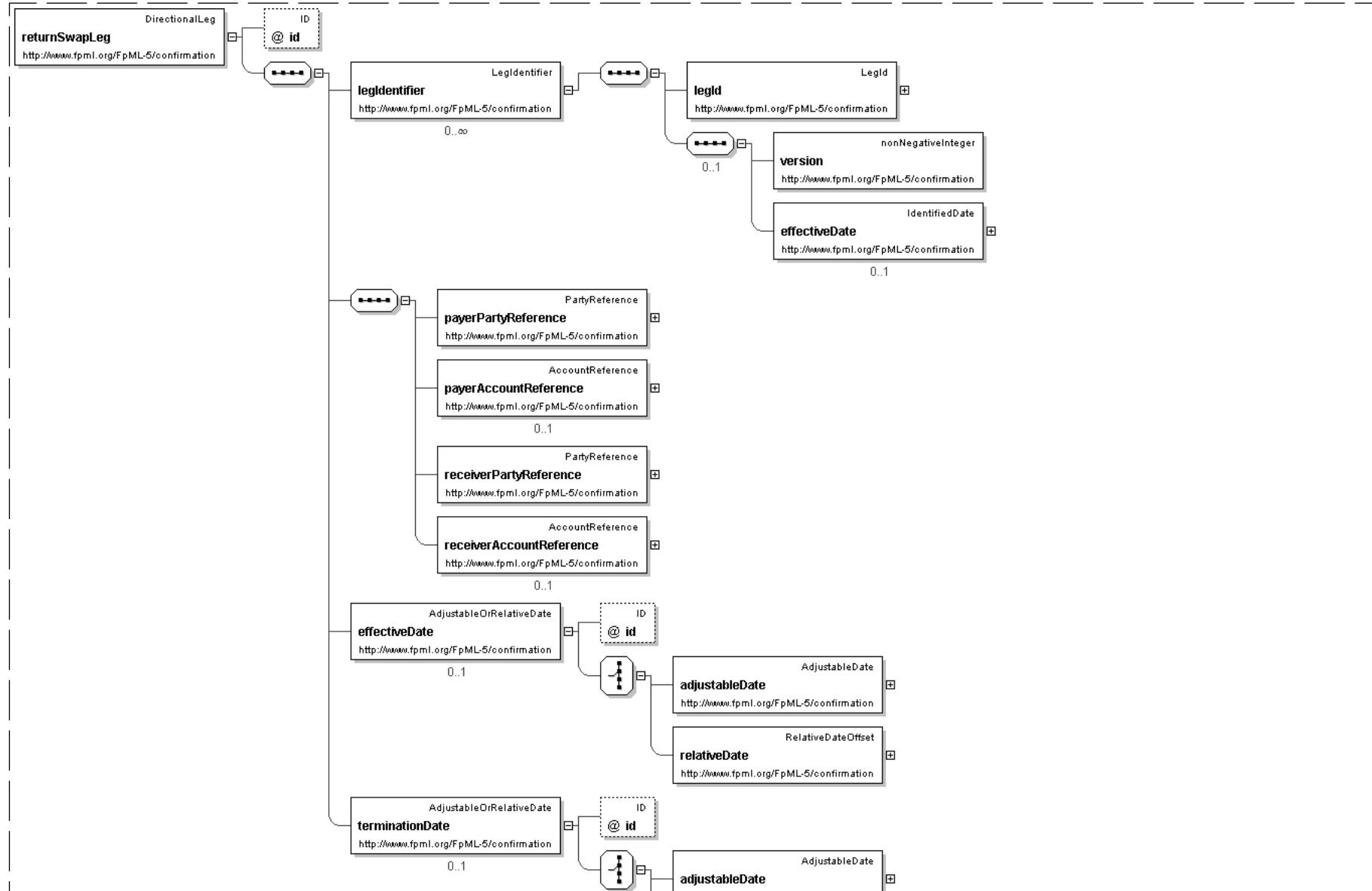
## Element: returnSwapLeg

- The following elements can be used wherever this element is referenced

- interestLeg
  - returnLeg

<b>Name</b>	returnSwapLeg
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">ReturnSwapBase</a>
<b>Type</b>	<a href="#">DirectionalLeg</a>
<b>Nillable</b>	no
<b>Abstract</b>	yes
<b>Documentation</b>	An placeholder for the actual Return Swap Leg definition.

## Logical Diagram



**XML Instance Representation**

```

<returnSwapLeg
id=" xsd:ID [0..1]">
    <legIdentifier> LegIdentifier </legIdentifier> [0..*]
    'Version aware identification of this leg.'

    <payerPartyReference> PartyReference </payerPartyReference> [1]
    'A reference to the party responsible for making the payments defined by this structure.'

    <payerAccountReference> AccountReference </payerAccountReference> [0..1]
    'A reference to the account responsible for making the payments defined by this structure.'

    <receiverPartyReference> PartyReference </receiverPartyReference> [1]
    'A reference to the party that receives the payments corresponding to this structure.'

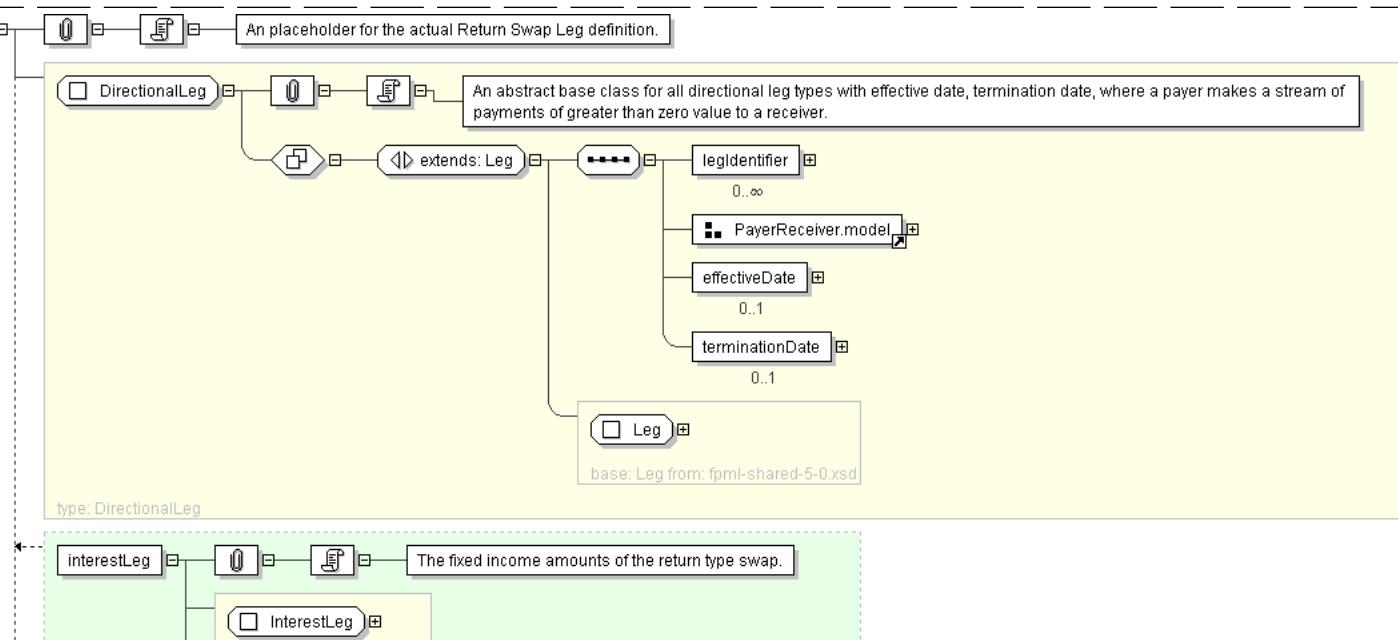
    <receiverAccountReference> AccountReference </receiverAccountReference> [0..1]
    'A reference to the account that receives the payments corresponding to this structure.'

    <effectiveDate> AdjustableOrRelativeDate </effectiveDate> [0..1]
    'Specifies the effective date of this leg of the swap. When defined in relation to a date specified somewhere else in the document (through the relativeDate component), this element will typically point to the effective date of the other leg of the swap.'

    <terminationDate> AdjustableOrRelativeDate </terminationDate> [0..1]
    'Specifies the termination date of this leg of the swap. When defined in relation to a date specified somewhere else in the document (through the relativeDate component), this element will typically point to the termination date of the other leg of the swap.'

</returnSwapLeg>

```

**Diagram**

**Schema Component Representation**

```
<xsd:element name="returnSwapLeg" type="DirectionalLeg" abstract="true"/>
```

top

**Global Definitions****Complex Type: AdditionalDisruptionEvents**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None

<b>Name</b>	AdditionalDisruptionEvents
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">ExtraordinaryEvents</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type for defining ISDA 2002 Equity Derivative Additional Disruption Events.

**XML Instance Representation**

```
<...>
<changeInLaw> xsd:boolean </changeInLaw> [0..1]
  'If true, then change in law is applicable.'

<failureToDeliver> xsd:boolean </failureToDeliver> [0..1]
  'Where the underlying is shares and the transaction is physically settled, then, if true,
  a failure to deliver the shares on the settlement date will not be an event of default for
  the purposes of the master agreement.'

<insolvencyFiling> xsd:boolean </insolvencyFiling> [0..1]
  'If true, then insolvency filing is applicable.'

<hedgingDisruption> xsd:boolean </hedgingDisruption> [0..1]
  'If true, then hedging disruption is applicable.'

<lossOfStockBorrow> xsd:boolean </lossOfStockBorrow> [0..1]
  'If true, then loss of stock borrow is applicable.'

<maximumStockLoanRate> RestrictedPercentage </maximumStockLoanRate> [0..1]
  'Specifies the maximum stock loan rate for Loss of Stock Borrow.'

<increasedCostOfStockBorrow> xsd:boolean </increasedCostOfStockBorrow> [0..1]
  'If true, then increased cost of stock borrow is applicable.'
```

```

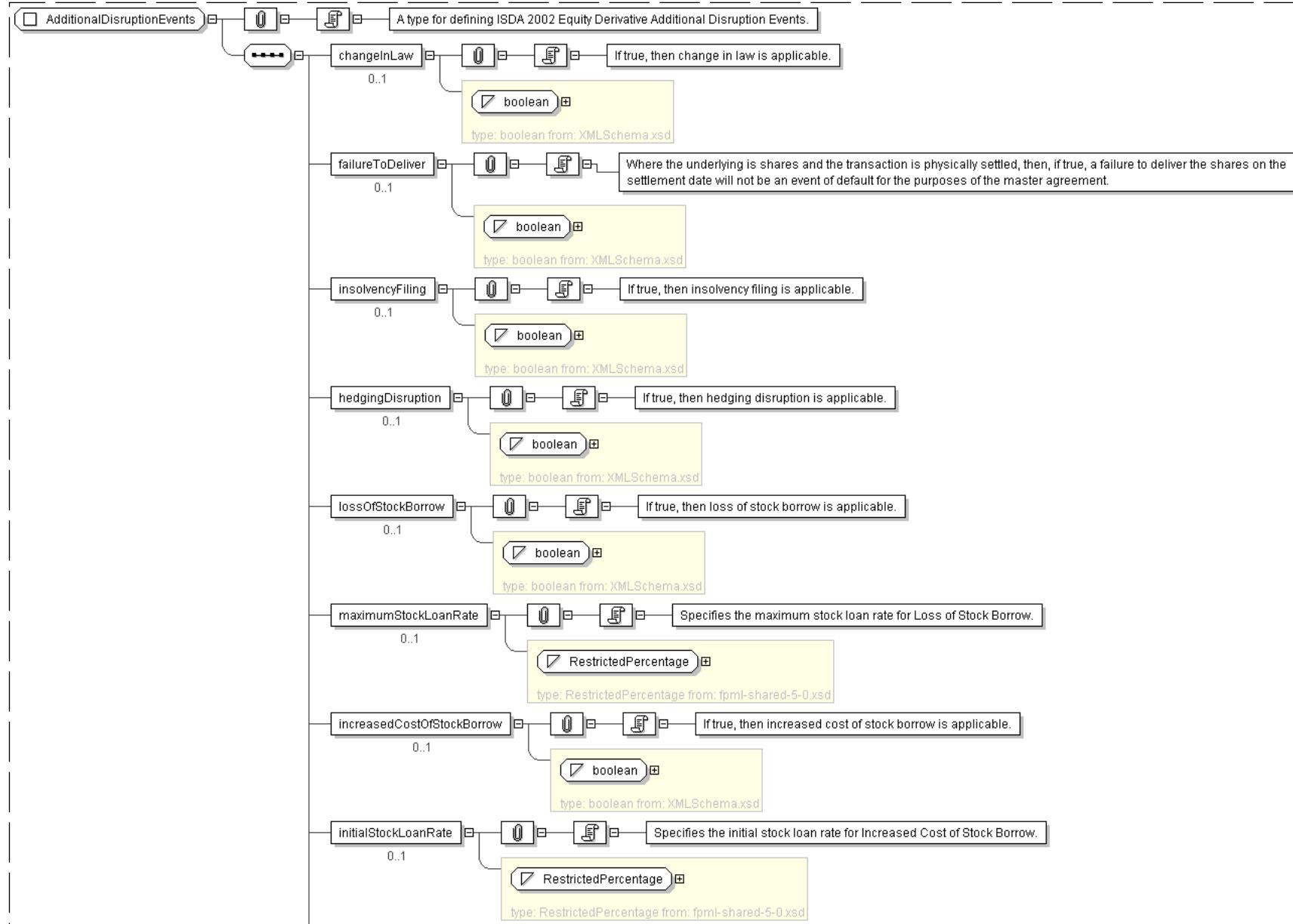
<initialStockLoanRate> RestrictedPercentage </initialStockLoanRate> [0..1]
'Specifies the initial stock loan rate for Increased Cost of Stock Borrow.'

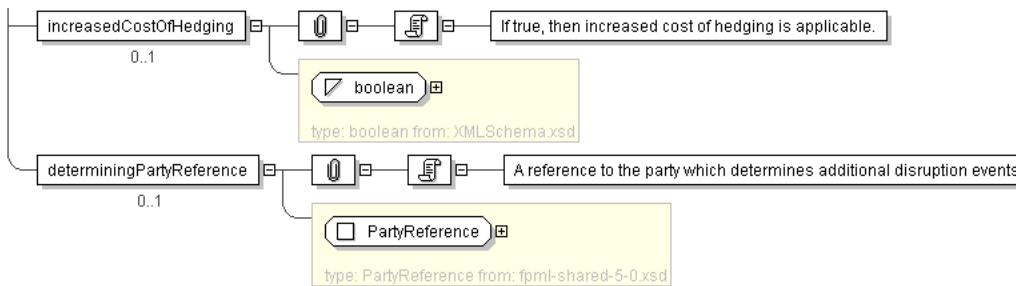
<increasedCostOfHedging> xsd:boolean </increasedCostOfHedging> [0..1]
'If true, then increased cost of hedging is applicable.'

<determiningPartyReference> PartyReference </determiningPartyReference> [0..1]
'A reference to the party which determines additional disruption events.'

</...>

```

**Diagram**

**Schema Component Representation**

```

<xsd:complexType name="AdditionalDisruptionEvents">
  <xsd:sequence>
    <xsd:element name="changeInLaw" type="xsd:boolean" minOccurs="0"/>
    <xsd:element name="failureToDeliver" type="xsd:boolean" minOccurs="0"/>
    <xsd:element name="insolvencyFiling" type="xsd:boolean" minOccurs="0"/>
    <xsd:element name="hedgingDisruption" type="xsd:boolean" minOccurs="0"/>
    <xsd:element name="lossOfStockBorrow" type="xsd:boolean" minOccurs="0"/>
    <xsd:element name="maximumStockLoanRate" type="RestrictedPercentage" minOccurs="0"/>
    <xsd:element name="increasedCostOfStockBorrow" type="xsd:boolean" minOccurs="0"/>
    <xsd:element name="initialStockLoanRate" type="RestrictedPercentage" minOccurs="0"/>
    <xsd:element name="increasedCostOfHedging" type="xsd:boolean" minOccurs="0"/>
    <xsd:element name="determiningPartyReference" type="PartyReference" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>

```

[top](#)**Complex Type: AdditionalPaymentAmount**

**Super-types:** None  
**Sub-types:** None

<b>Name</b>	AdditionalPaymentAmount
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">ReturnSwapAdditionalPayment</a>
<b>Abstract</b>	no
<b>Documentation</b>	Specifies the amount of the fee along with, when applicable, the formula that supports its determination.

**XML Instance Representation**

```

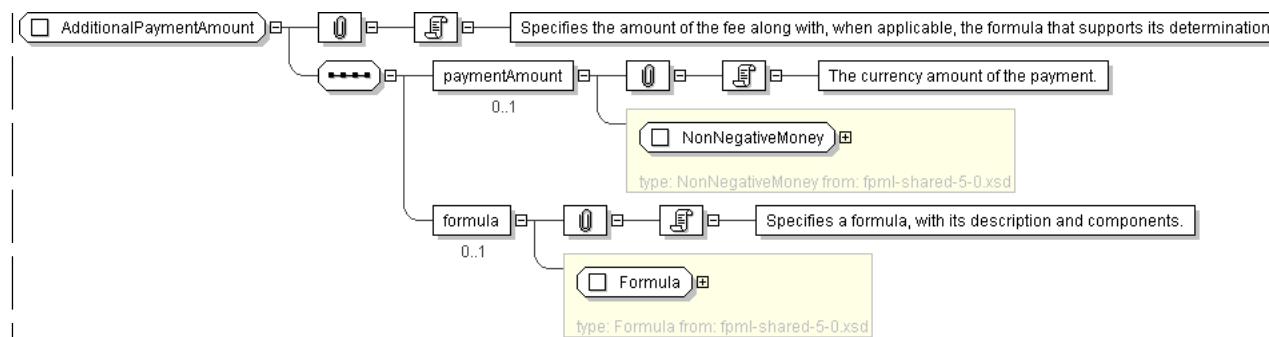
<...>
<paymentAmount> NonNegativeMoney </paymentAmount> [0..1]
  'The currency amount of the payment.'

  <formula> Formula </formula> [0..1]
  'Specifies a formula, with its description and components.'

</...>

```

**Diagram**

**Schema Component Representation**

```

<xsd:complexType name="AdditionalPaymentAmount">
  <xsd:sequence>
    <xsd:element name="paymentAmount" type="NonNegativeMoney" minOccurs="0"/>
    <xsd:element name="formula" type="Formula" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>
  
```

[top](#)**Complex Type: AdjustableDateOrRelativeDateSequence**

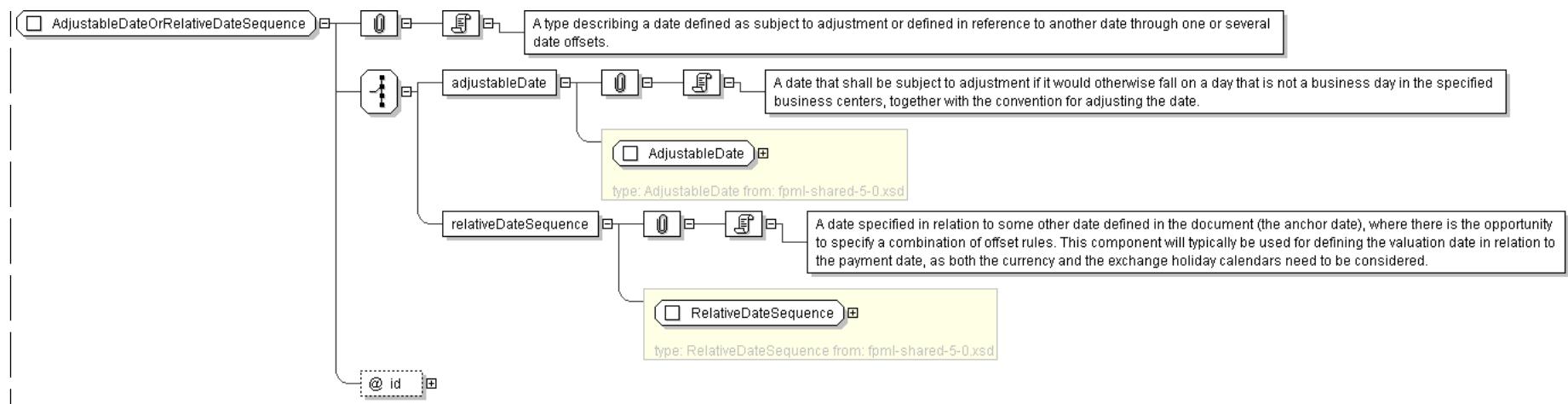
<b>Super-types:</b>	None
<b>Sub-types:</b>	None
<b>Name</b>	AdjustableDateOrRelativeDateSequence
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">EquityValuation</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type describing a date defined as subject to adjustment or defined in reference to another date through one or several date offsets.

**XML Instance Representation**

```

<...
  id=" xsd:ID [0..1]">
  Start Choice [1]
    <adjustableDate> AdjustableDate </adjustableDate> [1]
      '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.'
    <relativeDateSequence> RelativeDateSequence </relativeDateSequence> [1]
      'A date specified in relation to some other date defined in the document (the anchor
       date), where there is the opportunity to specify a combination of offset rules. This
       component will typically be used for defining the valuation date in relation to the
       payment date, as both the currency and the exchange holiday calendars need to be considered.'
  End Choice
</...>
  
```

**Diagram**

**Schema Component Representation**

```

<xsd:complexType name="AdjustableDateOrRelativeDateSequence">
    <xsd:choice>
        <xsd:element name="adjustableDate" type=" AdjustableDate " />
        <xsd:element name="relativeDateSequence" type=" RelativeDateSequence " />
    </xsd:choice>
    <xsd:attribute name="id" type=" xsd:ID " />
</xsd:complexType>

```

top

**Complex Type: BoundedCorrelation**

**Super-types:** None  
**Sub-types:** None

<b>Name</b>	BoundedCorrelation
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">Correlation</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type describing correlation bounds, which form a cap and a floor on the realized correlation.

**XML Instance Representation**

```

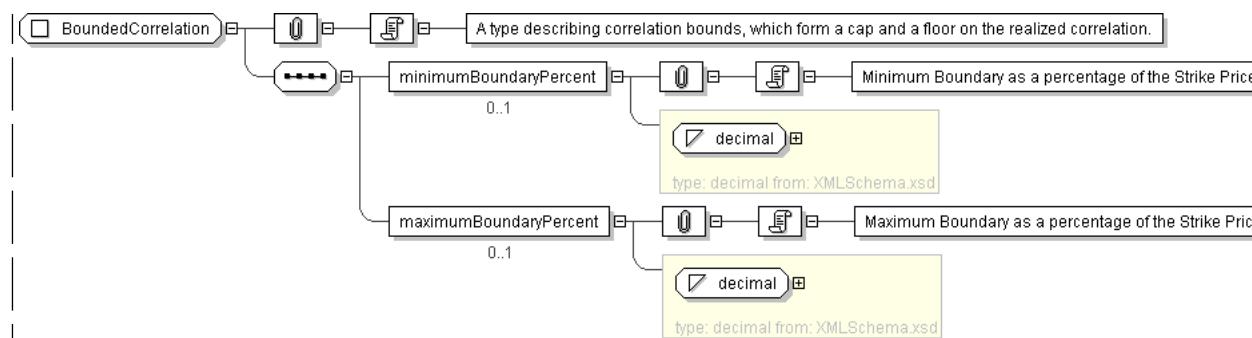
<...>
<minimumBoundaryPercent> xsd:decimal </minimumBoundaryPercent> [0..1]
    'Minimum Boundary as a percentage of the Strike Price.'

<maximumBoundaryPercent> xsd:decimal </maximumBoundaryPercent> [0..1]
    'Maximum Boundary as a percentage of the Strike Price.'

</...>

```

**Diagram**

**Schema Component Representation**

```

<xsd:complexType name="BoundedCorrelation">
  <xsd:sequence>
    <xsd:element name="minimumBoundaryPercent" type="xsd:decimal" minOccurs="0"/>
    <xsd:element name="maximumBoundaryPercent" type="xsd:decimal" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>

```

[top](#)**Complex Type: BoundedVariance**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None
<b>Name</b>	BoundedVariance
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">Variance</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type describing variance bounds, which are used to exclude money price values outside of the specified range. In a Up Conditional Swap Underlyer price must be equal to or higher than Lower Barrier. In a Down Conditional Swap Underlyer price must be equal to or lower than Upper Barrier. In a Corridor Conditional Swap Underlyer price must be equal to or higher than Lower Barrier and must be equal to or lower than Upper Barrier.

**XML Instance Representation**

```

<...>
  <realisedVarianceMethod> RealisedVarianceMethodEnum </realisedVarianceMethod> [1]
  'The contract specifies whether which price must satisfy the boundary condition.'

  <daysInRangeAdjustment> xsd:boolean </daysInRangeAdjustment> [1]
  'The contract specifies whether the notional should be scaled by the Number of Days in Range divided by the Expected N. The number of Days in Ranges refers to the number of returns that contribute to the realized volatility.'

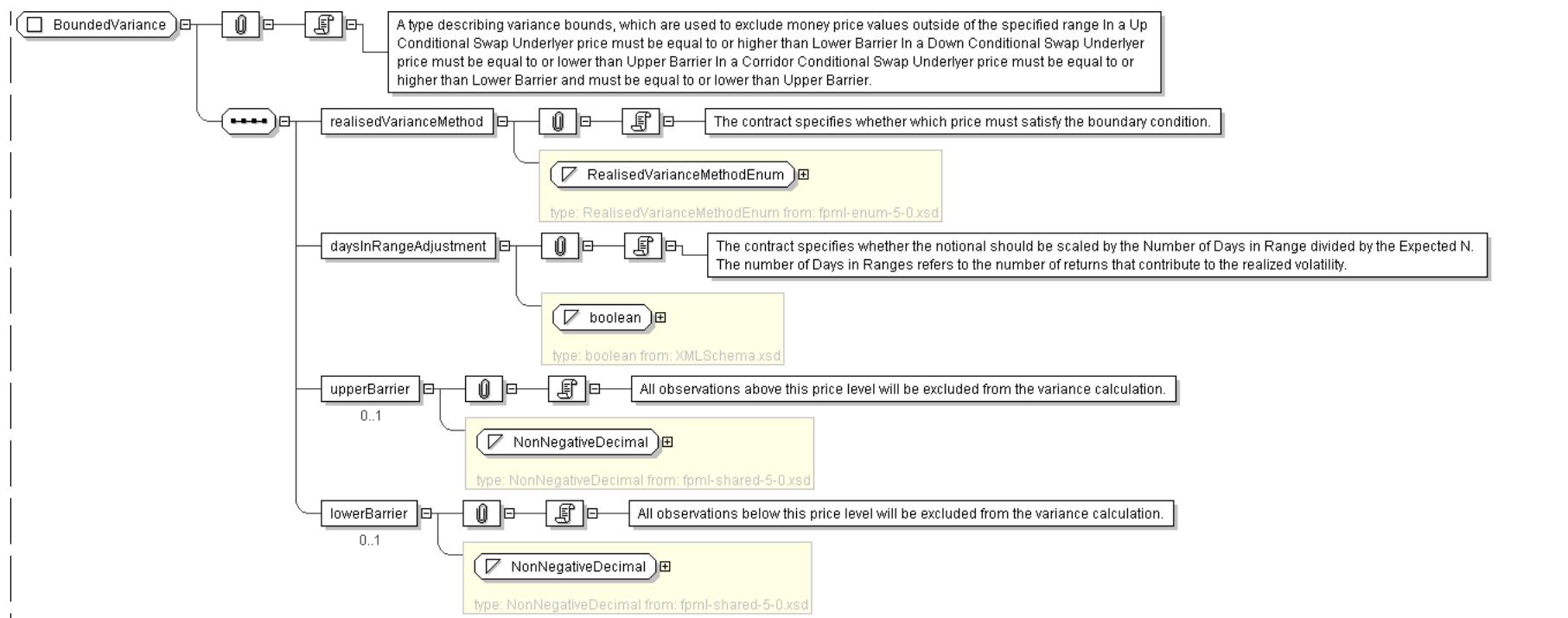
  <upperBarrier> NonNegativeDecimal </upperBarrier> [0..1]
  'All observations above this price level will be excluded from the variance calculation.'

  <lowerBarrier> NonNegativeDecimal </lowerBarrier> [0..1]
  'All observations below this price level will be excluded from the variance calculation.'

</...>

```

**Diagram**

**Schema Component Representation**

```

<xsd:complexType name="BoundedVariance">
  <xsd:sequence>
    <xsd:element name="realisedVarianceMethod" type="RealisedVarianceMethodEnum" />
    <xsd:element name="daysInRangeAdjustment" type="xsd:boolean" />
    <xsd:element name="upperBarrier" type="NonNegativeDecimal" minOccurs="0"/>
    <xsd:element name="lowerBarrier" type="NonNegativeDecimal" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>

```

top

**Complex Type: CalculatedAmount**

Super-types:	None
Sub-types:	None

Name	CalculatedAmount
Abstract	yes
Documentation	An abstract base class for all calculated money amounts, which are in the currency of the cash multiplier of the calculation.

**XML Instance Representation**

```

<...>
<calculationDates> AdjustableRelativeOrPeriodicDates </calculationDates> [0..1]
  Specifies the date on which a calculation or an observation will be performed for the
  purpose of calculating the amount.

<observationStartDate> AdjustableOrRelativeDate </observationStartDate> [0..1]
  The start of the period over which observations are made which are used in the
  calculation Used when the observation start date differs from the trade date such as
  for forward starting swaps.

```

```

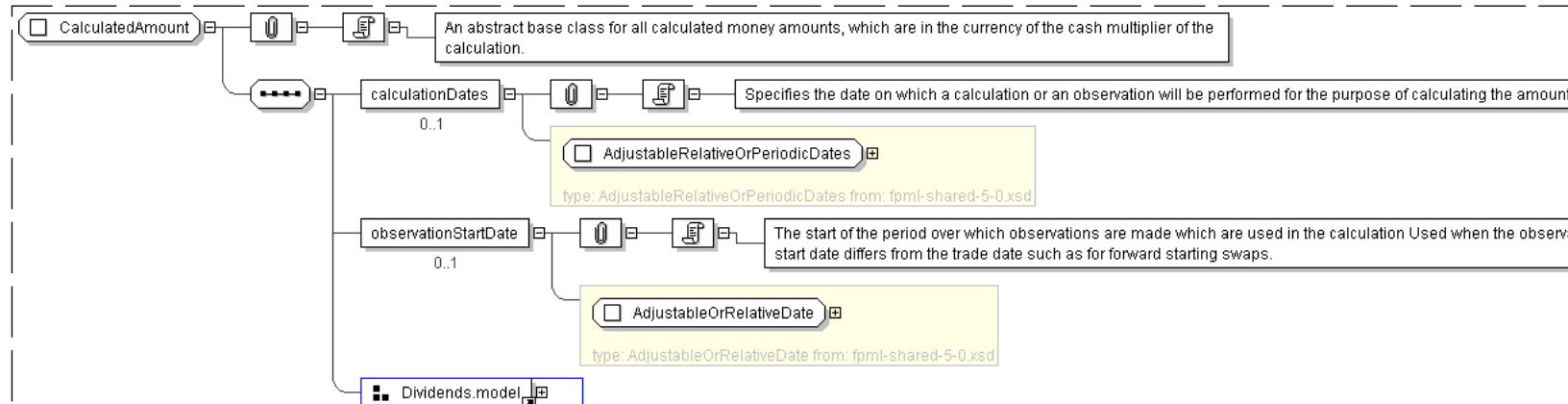
<optionsExchangeDividends> xsd:boolean </optionsExchangeDividends> [0..1]
  'If present and true, then options exchange dividends are applicable.'

<additionalDividends> xsd:boolean </additionalDividends> [0..1]
  'If present and true, then additional dividends are applicable.'

<allDividends> xsd:boolean </allDividends> [0..1]
  'Represents the European Master Confirmation value of \'All Dividends\' which, when
  applicable, signifies that, for a given Ex-Date, the daily observed Share Price for that day
  is adjusted (reduced) by the cash dividend and/or the cash value of any non cash dividend
  per Share (including Extraordinary Dividends) declared by the Issuer.'

<...>

```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="CalculatedAmount" abstract="true">
  <xsd:sequence>
    <xsd:element name="calculationDates" type="#AdjustableRelativeOrPeriodicDates" minOccurs="0"/>
    <xsd:element name="observationStartDate" type="#AdjustableOrRelativeDate" minOccurs="0"/>
    <xsd:group ref="Dividends.model"/>
  </xsd:sequence>
</xsd:complexType>

```

top

**Complex Type: CalculationFromObservation**

**Super-types:** None  
**Sub-types:**

- [Correlation](#) (by extension)
- [Variance](#) (by extension)

<b>Name</b>	CalculationFromObservation
<b>Abstract</b>	yes
<b>Documentation</b>	Abstract base class for all calculation from observed values.

**XML Instance Representation**

```

<...>
Start Choice [1]
  <initialLevel> xsd:decimal </initialLevel> [1]
  'Contract will strike off this initial level.'

  <closingLevel> xsd:boolean </closingLevel> [1]
  'If true this contract will strike off the closing level of the default exchange'

```

```

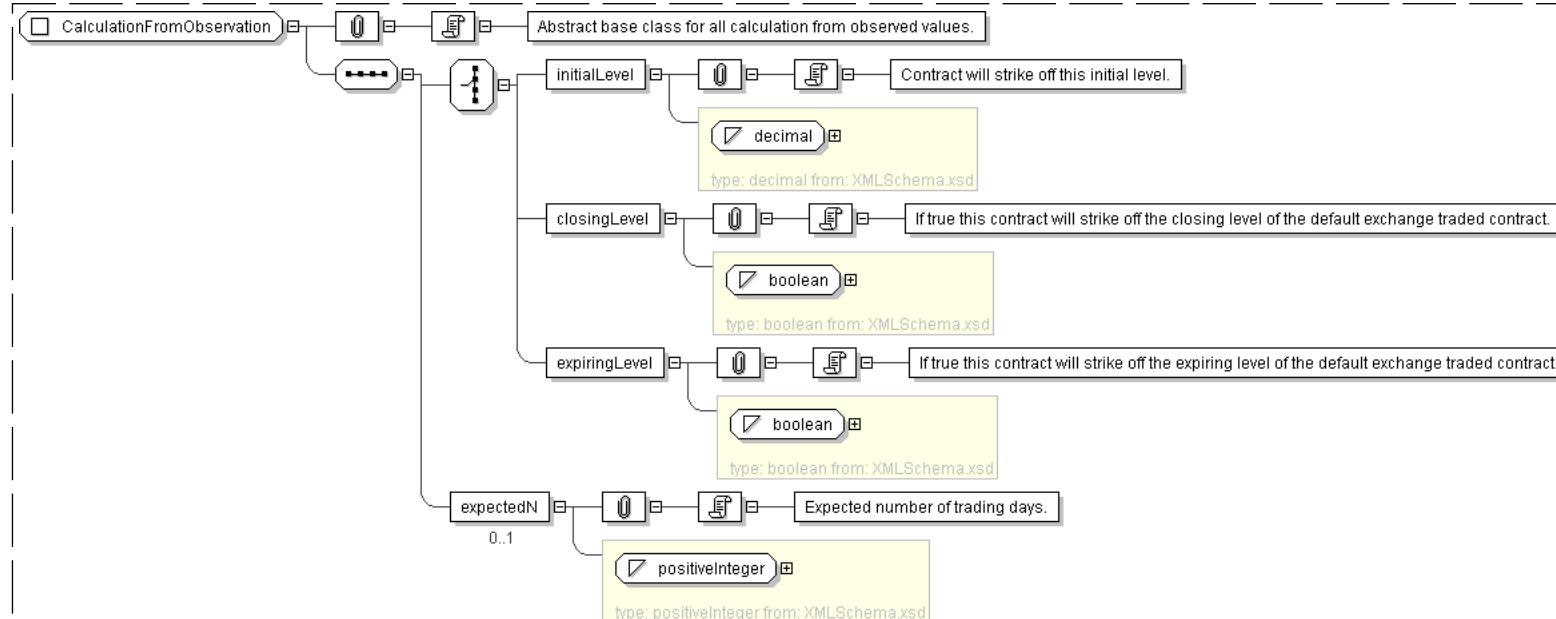
traded contract.

<expiringLevel> xsd:boolean </expiringLevel> [1]
'If true this contract will strike off the expiring level of the default exchange
traded contract.'

End Choice
<expectedN> xsd:positiveInteger </expectedN> [0..1]
'Expected number of trading days.'

</...>

```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="CalculationFromObservation" abstract="true">
  <xsd:sequence>
    <xsd:choice>
      <xsd:element name="initialLevel" type="xsd:decimal" />
      <xsd:element name="closingLevel" type="xsd:boolean" />
      <xsd:element name="expiringLevel" type="xsd:boolean" />
    </xsd:choice>
    <xsd:element name="expectedN" type="xsd:positiveInteger" minOccurs="0" />
  </xsd:sequence>
</xsd:complexType>

```

top

**Complex Type: Compounding**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None

<b>Name</b>	Compounding
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">InterestCalculation</a>
<b>Abstract</b>	no

**Documentation**

Specifies the compounding method and the compounding rate.

**XML Instance Representation**

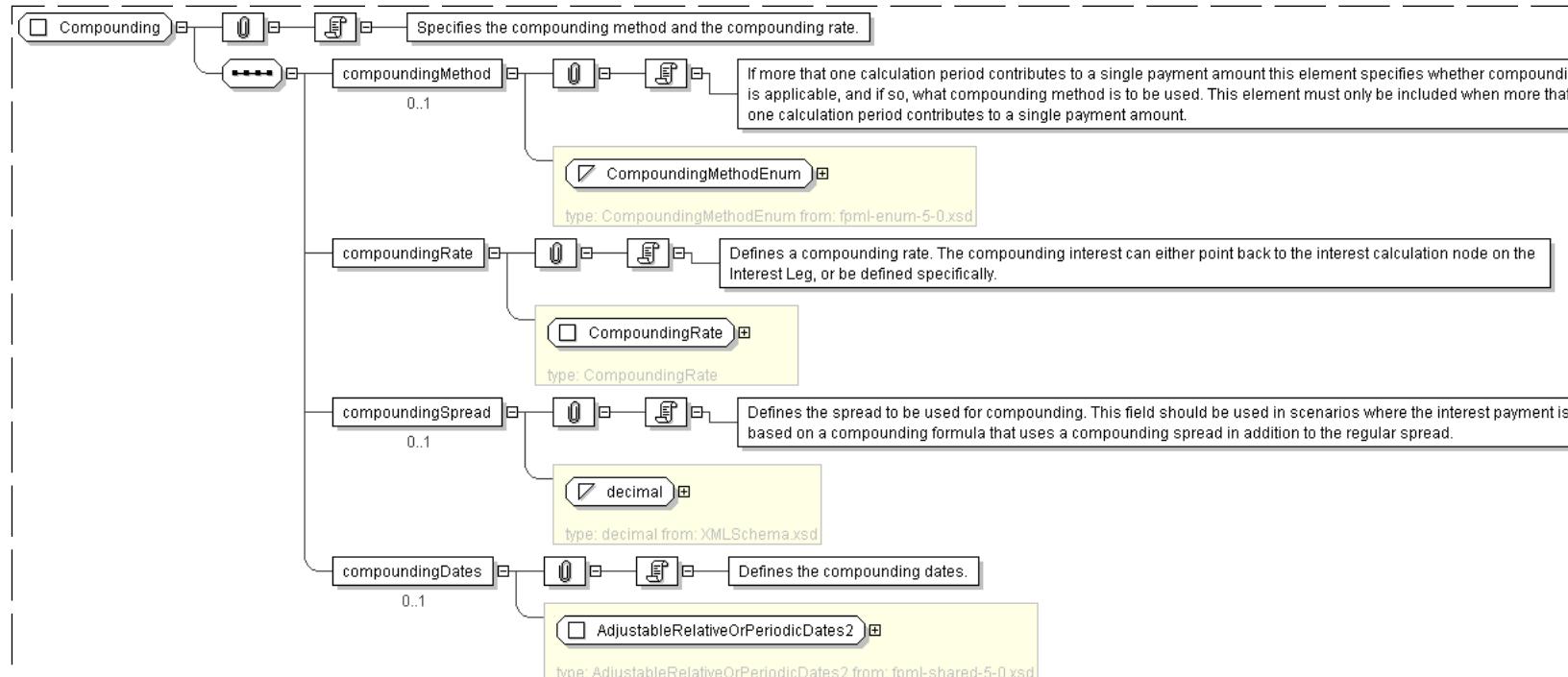
```
<...>
<compoundingMethod> CompoundingMethodEnum </compoundingMethod> [0..1]
'If more than one calculation period contributes to a single payment amount this element specifies whether compounding is applicable, and if so, what compounding method is to be used. This element must only be included when more than one calculation period contributes to a single payment amount.'

<compoundingRate> CompoundingRate </compoundingRate> [1]
'Defines a compounding rate. The compounding interest can either point back to the interest calculation node on the Interest Leg, or be defined specifically.'

<compoundingSpread> xsd:decimal </compoundingSpread> [0..1]
'Defines the spread to be used for compounding. This field should be used in scenarios where the interest payment is based on a compounding formula that uses a compounding spread in addition to the regular spread.'

<compoundingDates> AdjustableRelativeOrPeriodicDates2 </compoundingDates> [0..1]
'Defines the compounding dates.'

</...>
```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="Compounding">
  <xsd:sequence>
    <xsd:element name="compoundingMethod" type=" CompoundingMethodEnum " minOccurs="0" />
    <xsd:element name="compoundingRate" type=" CompoundingRate " />
    <xsd:element name="compoundingSpread" type=" xsd:decimal " minOccurs="0" />
    <xsd:element name="compoundingDates" type=" AdjustableRelativeOrPeriodicDates2 " minOccurs="0" />
  </xsd:sequence>
</xsd:complexType>

```

## Complex Type: CompoundingRate

Super-types:	None
Sub-types:	None

Name	CompoundingRate
Used by (from the same schema document)	Complex Type <a href="#">Compounding</a>
Abstract	no
Documentation	A type defining a compounding rate. The compounding interest can either point back to the floating rate calculation of interest calculation node on the Interest Leg, or be defined specifically.

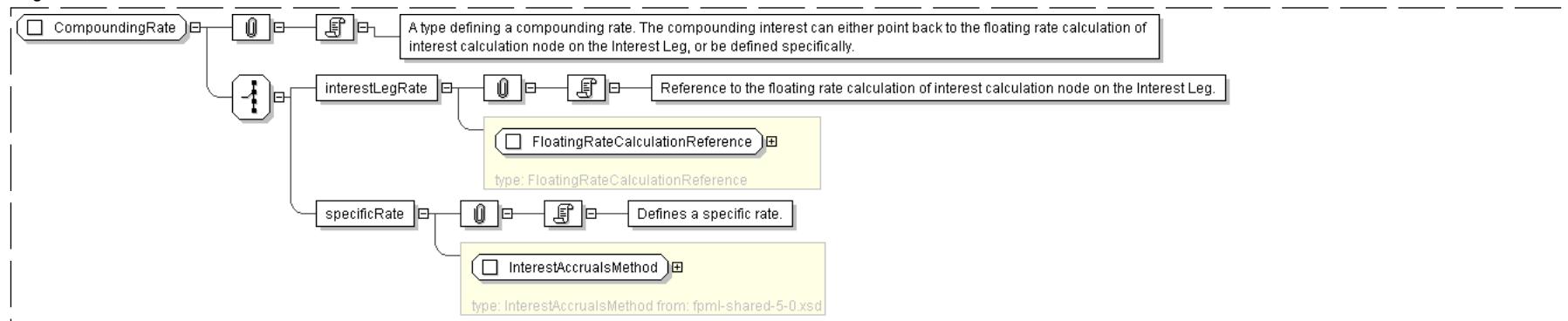
### XML Instance Representation

```
<...>
Start Choice [1]
<interestLegRate> FloatingRateCalculationReference </interestLegRate> [1]
'Reference to the floating rate calculation of interest calculation node on the Interest Leg.'

<specificRate> InterestAccrualsMethod </specificRate> [1]
'Defines a specific rate.'

End Choice
</...>
```

### Diagram



### Schema Component Representation

```
<xsd:complexType name="CompoundingRate">
  <xsd:choice>
    <xsd:element name="interestLegRate" type="FloatingRateCalculationReference" />
    <xsd:element name="specificRate" type="InterestAccrualsMethod" />
  </xsd:choice>
</xsd:complexType>
```

## Complex Type: Correlation

Super-types:	<a href="#">CalculationFromObservation</a> < <b>Correlation</b> (by extension)
Sub-types:	None

Name	Correlation
Abstract	no

**Documentation**

A type describing the correlation amount of a correlation swap.

**XML Instance Representation**

```
<...>
Start Choice [1]
  <initialLevel> xsd:decimal </initialLevel> [1]
  'Contract will strike off this initial level.'

  <closingLevel> xsd:boolean </closingLevel> [1]
  'If true this contract will strike off the closing level of the default exchange
traded contract.'

  <expiringLevel> xsd:boolean </expiringLevel> [1]
  'If true this contract will strike off the expiring level of the default exchange
traded contract.'

End Choice
<expectedN> xsd:positiveInteger </expectedN> [0..1]
'Expected number of trading days.'

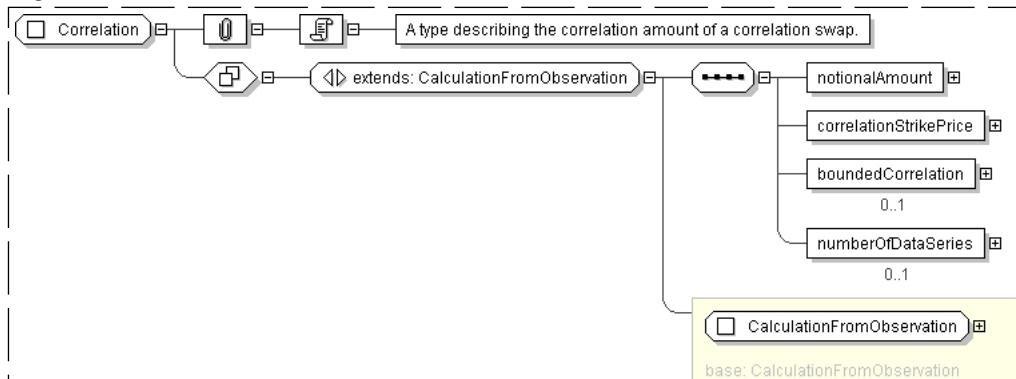
<notionalAmount> NonNegativeMoney </notionalAmount> [1]
'Notional amount, which is a cash multiplier.'

<correlationStrikePrice> CorrelationValue </correlationStrikePrice> [1]
'Correlation Strike Price.'

<boundedCorrelation> BoundedCorrelation </boundedCorrelation> [0..1]
'Bounded Correlation.'

<numberOfDataSeries> xsd:positiveInteger </numberOfDataSeries> [0..1]
'Number of data series, normal market practice is that correlation data sets are drawn
from geographic market areas, such as America, Europe and Asia Pacific, each of
these geographic areas will have its own data series to avoid contagion.'

</...>
```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="Correlation">
  <xsd:complexContent>
    <xsd:extension base=" CalculationFromObservation ">
      <xsd:sequence>
        <xsd:element name="notionalAmount" type=" NonNegativeMoney "/>
        <xsd:element name="correlationStrikePrice" type=" CorrelationValue "/>
        <xsd:element name="boundedCorrelation" type=" BoundedCorrelation " minOccurs="0"/>
        <xsd:element name="numberOfDataSeries" type=" xsd:positiveInteger " minOccurs="0"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
  
```

```

</xsd:extension>
</xsd:complexContent>
</xsd:complexType>

```

## Complex Type: DirectionalLeg

Super-types:

[Leg < DirectionalLeg](#) (by extension)

Sub-types:

- [DirectionalLegUnderlier](#) (by extension)
  - [DirectionalLegUnderlierValuation](#) (by extension)
- [InterestLeg](#) (by extension)
- [ReturnSwapLegUnderlier](#) (by extension)
  - [ReturnLeg](#) (by extension)

Name

DirectionalLeg

Used by (from the same schema document)

Element [returnSwapLeg](#)

Abstract

yes

Documentation

An abstract base class for all directional leg types with effective date, termination date, where a payer makes a stream of payments of greater than zero value to a receiver.

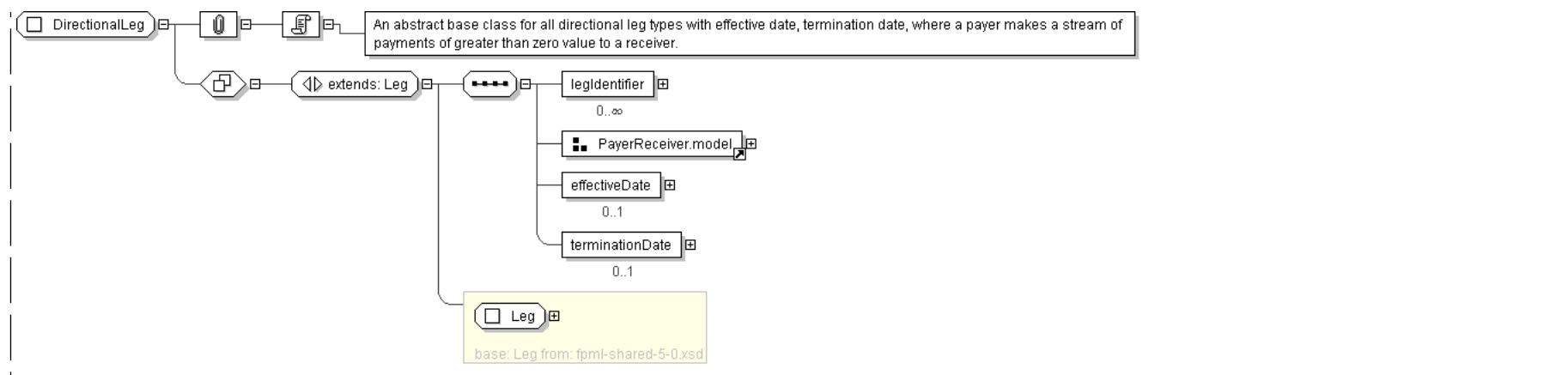
### XML Instance Representation

```

<...>
  id="xsd:ID [0..1]">
    <legIdentifier> LegIdentifier </legIdentifier> [0..*]
    "Version aware identification of this leg."
    <payerPartyReference> PartyReference </payerPartyReference> [1]
    "A reference to the party responsible for making the payments defined by this structure."
    <payerAccountReference> AccountReference </payerAccountReference> [0..1]
    "A reference to the account responsible for making the payments defined by this structure."
    <receiverPartyReference> PartyReference </receiverPartyReference> [1]
    "A reference to the party that receives the payments corresponding to this structure."
    <receiverAccountReference> AccountReference </receiverAccountReference> [0..1]
    "A reference to the account that receives the payments corresponding to this structure."
    <effectiveDate> AdjustableOrRelativeDate </effectiveDate> [0..1]
    "Specifies the effective date of this leg of the swap. When defined in relation to a date specified somewhere else in the document (through the relativeDate component), this element will typically point to the effective date of the other leg of the swap."
    <terminationDate> AdjustableOrRelativeDate </terminationDate> [0..1]
    "Specifies the termination date of this leg of the swap. When defined in relation to a date specified somewhere else in the document (through the relativeDate component), this element will typically point to the termination date of the other leg of the swap."
  </...>

```

### Diagram

**Schema Component Representation**

```

<xsd:complexType name="DirectionalLeg" abstract="true">
  <xsd:complexContent>
    <xsd:extension base="#_Leg">
      <xsd:sequence>
        <xsd:element name="legIdentifier" type="LegIdentifier" minOccurs="0" maxOccurs="unbounded"/>
        <xsd:group ref="PayerReceiver.model"/>
        <xsd:element name="effectiveDate" type="AdjustableOrRelativeDate" minOccurs="0"/>
        <xsd:element name="terminationDate" type="AdjustableOrRelativeDate" minOccurs="0"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
  
```

top

**Complex Type: DirectionalLegUnderlier**

<b>Super-types:</b>	<a href="#">Leg</a> < <a href="#">DirectionalLeg</a> (by extension) < <b>DirectionalLegUnderlier</b> (by extension)
<b>Sub-types:</b>	• <a href="#">DirectionalLegUnderlierValuation</a> (by extension)

<b>Name</b>	DirectionalLegUnderlier
<b>Abstract</b>	yes
<b>Documentation</b>	An abstract base class for all directional leg types with effective date, termination date, and underlyer where a payer makes a stream of payments of greater than zero value to a receiver.

**XML Instance Representation**

```

<...
id="xsd:ID [0..1]">
<legIdentifier> LegIdentifier </legIdentifier> [0..*]
'<i>Version aware identification of this leg.</i>'

<payerPartyReference> PartyReference </payerPartyReference> [1]
'<i>A reference to the party responsible for making the payments defined by this structure.</i>'

<payerAccountReference> AccountReference </payerAccountReference> [0..1]
'<i>A reference to the account responsible for making the payments defined by this structure.</i>'

<receiverPartyReference> PartyReference </receiverPartyReference> [1]
'<i>A reference to the party that receives the payments corresponding to this structure.</i>'

<receiverAccountReference> AccountReference </receiverAccountReference> [0..1]
'<i>A reference to the account that receives the payments corresponding to this structure.</i>'
  
```

```
<effectiveDate> AdjustableOrRelativeDate </effectiveDate> [0..1]
'Specifies the effective date of this leg of the swap. When defined in relation to a
date specified somewhere else in the document (through the relativeDate component),
this element will typically point to the effective date of the other leg of the swap.'
```

```
<terminationDate> AdjustableOrRelativeDate </terminationDate> [0..1]
'Specifies the termination date of this leg of the swap. When defined in relation to a
date specified somewhere else in the document (through the relativeDate component),
this element will typically point to the termination date of the other leg of the swap.'
```

```
<underlyer> Underlyer </underlyer> [1]
'Specifies the underlyer of the leg.'
```

```
<settlementType> SettlementTypeEnum </settlementType> [0..1]
<settlementDate> AdjustableOrRelativeDate </settlementDate> [0..1]
```

Start Group: SettlementAmountOrCurrency.model [0..1]

Start Choice [1]

```
<settlementAmount> Money </settlementAmount> [1]
```

'Settlement Amount'

```
<settlementCurrency> Currency </settlementCurrency> [1]
```

'Settlement Currency for use where the Settlement Amount cannot be known in advance'

End Choice

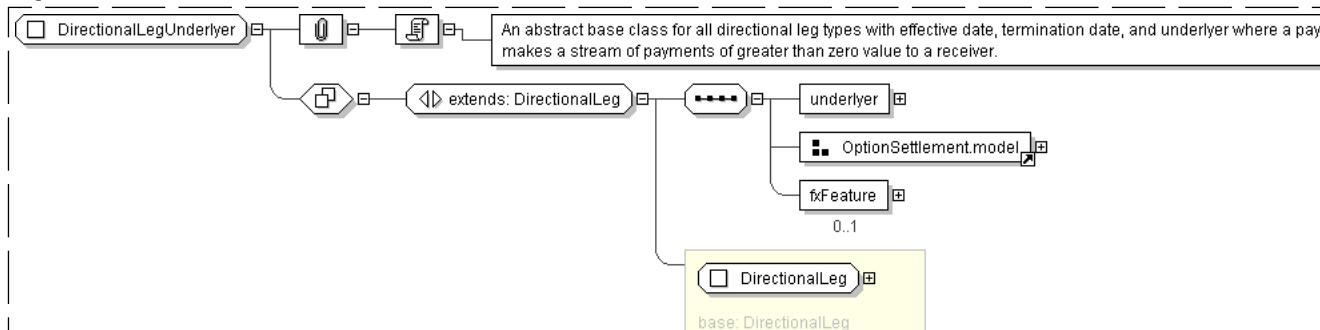
End Group: SettlementAmountOrCurrency.model

```
<fxFeature> FxFeature </fxFeature> [0..1]
```

'Quanto, Composite, or Cross Currency FX features.'

</...>

#### Diagram



#### Schema Component Representation

```

<xsd:complexType name="DirectionalLegUnderlyer" abstract="true">
  <xsd:complexContent>
    <xsd:extension base=" DirectionalLeg ">
      <xsd:sequence>
        <xsd:element name="underlyer" type=" Underlyer "/>
        <xsd:group ref=" OptionSettlement.model ">
          <xsd:element name="fxFeature" type=" FxFeature " minOccurs="0"/>
        </xsd:sequence>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>

```

#### Complex Type: DirectionalLegUnderlyerValuation

Super-types:

[Leg](#) < [DirectionalLeg](#) (by extension) < [DirectionalLegUnderlyer](#) (by extension) < [DirectionalLegUnderlyerValuation](#) (by extension)

<b>Sub-types:</b>	None
<b>Name</b>	DirectionalLegUnderlyerValuation
<b>Abstract</b>	yes
<b>Documentation</b>	An abstract base class for all directional leg types with effective date, termination date, and underlyer, where a payer makes a stream of payments of greater than zero value to a receiver.

**XML Instance Representation**

```

<...
  id=" xsd:ID [0..1]">
  <legIdentifier> LegIdentifier </legIdentifier> [0..*]
    'Version aware identification of this leg.'

  <payerPartyReference> PartyReference </payerPartyReference> [1]
    'A reference to the party responsible for making the payments defined by this structure.'

  <payerAccountReference> AccountReference </payerAccountReference> [0..1]
    'A reference to the account responsible for making the payments defined by this structure.'

  <receiverPartyReference> PartyReference </receiverPartyReference> [1]
    'A reference to the party that receives the payments corresponding to this structure.'

  <receiverAccountReference> AccountReference </receiverAccountReference> [0..1]
    'A reference to the account that receives the payments corresponding to this structure.'

  <effectiveDate> AdjustableOrRelativeDate </effectiveDate> [0..1]
    'Specifies the effective date of this leg of the swap. When defined in relation to a date specified somewhere else in the document (through the relativeDate component), this element will typically point to the effective date of the other leg of the swap.'

  <terminationDate> AdjustableOrRelativeDate </terminationDate> [0..1]
    'Specifies the termination date of this leg of the swap. When defined in relation to a date specified somewhere else in the document (through the relativeDate component), this element will typically point to the termination date of the other leg of the swap.'

  <underlyer> Underlyer </underlyer> [1]
    'Specifies the underlyer of the leg.'

  <settlementType> SettlementTypeEnum </settlementType> [0..1]
  <settlementDate> AdjustableOrRelativeDate </settlementDate> [0..1]
Start Group: SettlementAmountOrCurrency.model [0..1]
Start Choice [1]
  <settlementAmount> Money </settlementAmount> [1]
    'Settlement Amount'

  <settlementCurrency> Currency </settlementCurrency> [1]
    'Settlement Currency for use where the Settlement Amount cannot be known in advance'

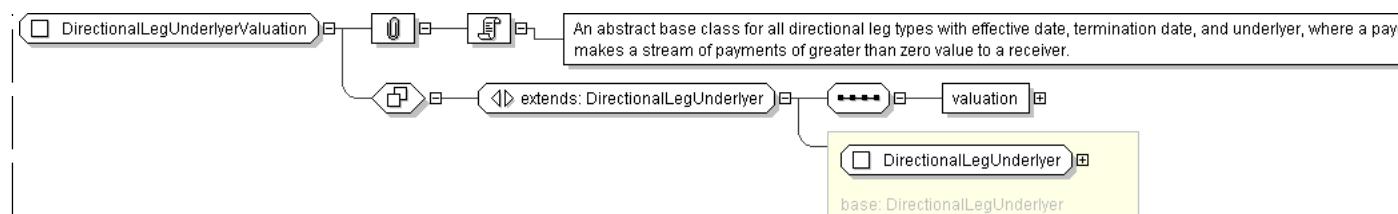
End Choice
End Group: SettlementAmountOrCurrency.model
<fxFeature> FxFeature </fxFeature> [0..1]
  'Quanto, Composite, or Cross Currency FX features.'

  <valuation> EquityValuation </valuation> [1]
    'Valuation of the underlyer.'

</...>

```

**Diagram**

**Schema Component Representation**

```
<xsd:complexType name="DirectionalLegUnderlyerValuation" abstract="true">
  <xsd:complexContent>
    <xsd:extension base="DirectionalLegUnderlyer">
      <xsd:sequence>
        <xsd:element name="valuation" type="EquityValuation" />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

[top](#)**Complex Type: DividendAdjustment**

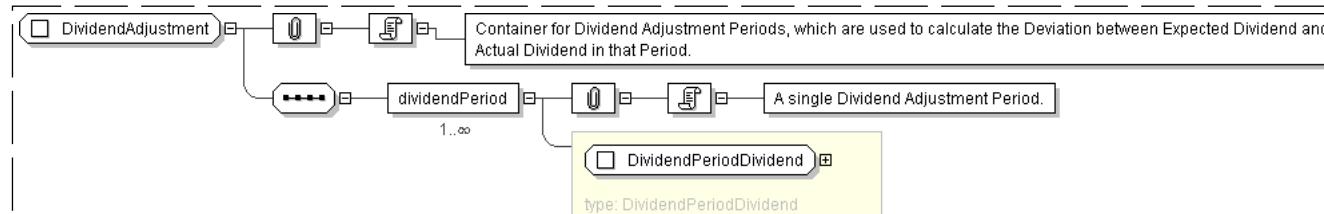
**Super-types:** None  
**Sub-types:** None

<b>Name</b>	DividendAdjustment
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">OptionFeatures</a>
<b>Abstract</b>	no
<b>Documentation</b>	Container for Dividend Adjustment Periods, which are used to calculate the Deviation between Expected Dividend and Actual Dividend in that Period.

**XML Instance Representation**

```
<...>
  <dividendPeriod> DividendPeriodDividend </dividendPeriod> [1..*]
  'A single Dividend Adjustment Period.'
```

&lt;/...&gt;

**Diagram****Schema Component Representation**

```
<xsd:complexType name="DividendAdjustment">
  <xsd:sequence>
    <xsd;element name="dividendPeriod" type="DividendPeriodDividend" maxOccurs="unbounded" />
  </xsd:sequence>
</xsd:complexType>
```

[top](#)**Complex Type: DividendConditions**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None
<b>Name</b>	DividendConditions
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">Return</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type describing the conditions governing the payment of dividends to the receiver of the equity return. With the exception of the dividend payout ratio, which is defined for each of the underlying components.
<b>XML Instance Representation</b>	
<...>	
<dividendReinvestment> <a href="#">xsd:boolean</a> </dividendReinvestment> [0..1]	
'Boolean element that defines whether the dividend will be reinvested or not.'	
<dividendEntitlement> <a href="#">DividendEntitlementEnum</a> </dividendEntitlement> [0..1]	
'Defines the date on which the receiver on the equity return is entitled to the dividend.'	
<dividendAmount> <a href="#">DividendAmountTypeEnum</a> </dividendAmount> [0..1]	
<dividendPaymentDate> <a href="#">DividendPaymentDate</a> </dividendPaymentDate> [0..1]	
'Specifies when the dividend will be paid to the receiver of the equity return. Has the meaning as defined in the ISDA 2002 Equity Derivatives Definitions. Is not applicable in the case of a dividend reinvestment election.'	
Start <a href="#">Choice</a> [1]	
<dividendPeriodEffectiveDate> <a href="#">DateReference</a> </dividendPeriodEffectiveDate> [0..1]	
'Dividend period has the meaning as defined in the ISDA 2002 Equity Derivatives Definitions. This element specifies the date on which the dividend period will commence.'	
<dividendPeriodEndDate> <a href="#">DateReference</a> </dividendPeriodEndDate> [0..1]	
'Dividend period has the meaning as defined in the ISDA 2002 Equity Derivatives Definitions. This element specifies the date on which the dividend period will end. It includes a boolean attribute for defining whether this end date is included or excluded from the dividend period.'	
<dividendPeriod> <a href="#">DividendPeriodEnum</a> </dividendPeriod> [1]	
'Defines the First Period or the Second Period, as defined in the 2002 ISDA Equity Derivatives Definitions.'	
End Choice	
<extraOrdinaryDividends> <a href="#">PartyReference</a> </extraOrdinaryDividends> [0..1]	
'Reference to the party which determines if dividends are extraordinary in relation to normal levels.'	
<excessDividendAmount> <a href="#">DividendAmountTypeEnum</a> </excessDividendAmount> [0..1]	
'Determination of Gross Cash Dividend per Share.'	
Start Group: <a href="#">CurrencyAndDeterminationMethod.model</a> [0..1]	
Start <a href="#">Choice</a> [1]	
<currency> <a href="#">IdentifiedCurrency</a> </currency> [1]	
'The currency in which an amount is denominated.'	
<determinationMethod> <a href="#">DeterminationMethod</a> </determinationMethod> [1]	
'Specifies the method according to which an amount or a date is determined.'	
<currencyReference> <a href="#">IdentifiedCurrencyReference</a> </currencyReference> [1]	
'Reference to a currency defined elsewhere in the document'	
End Choice	
End Group: <a href="#">CurrencyAndDeterminationMethod.model</a>	
<dividendFxTriggerDate> <a href="#">DividendPaymentDate</a> </dividendFxTriggerDate> [0..1]	
'Specifies the date on which the FX rate will be considered in the case of a Composite FX swap.'	
<interestAccrualsMethod> <a href="#">InterestAccrualsCompoundingMethod</a> </interestAccrualsMethod> [0..1]	
'Defines the way in which interests are accrued: the applicable rate (fixed or	

*floating reference) and the compounding method.'*, 'FpML entity'

<numberOfIndexUnits> [NonNegativeDecimal](#) </numberOfIndexUnits> [0..1]

'Defines the Number Of Index Units applicable to a Dividend.'

<declaredCashDividendPercentage> [NonNegativeDecimal](#) </declaredCashDividendPercentage> [0..1]

'Declared Cash Dividend Percentage.'

<declaredCashEquivalentDividendPercentage> [NonNegativeDecimal](#)

</declaredCashEquivalentDividendPercentage> [0..1]

'Declared Cash Equivalent Dividend Percentage.'

<nonCashDividendTreatment> [NonCashDividendTreatmentEnum](#) </nonCashDividendTreatment> [0..1]

'Defines treatment of Non-Cash Dividends.'

<dividendComposition> [DividendCompositionEnum](#) </dividendComposition> [0..1]

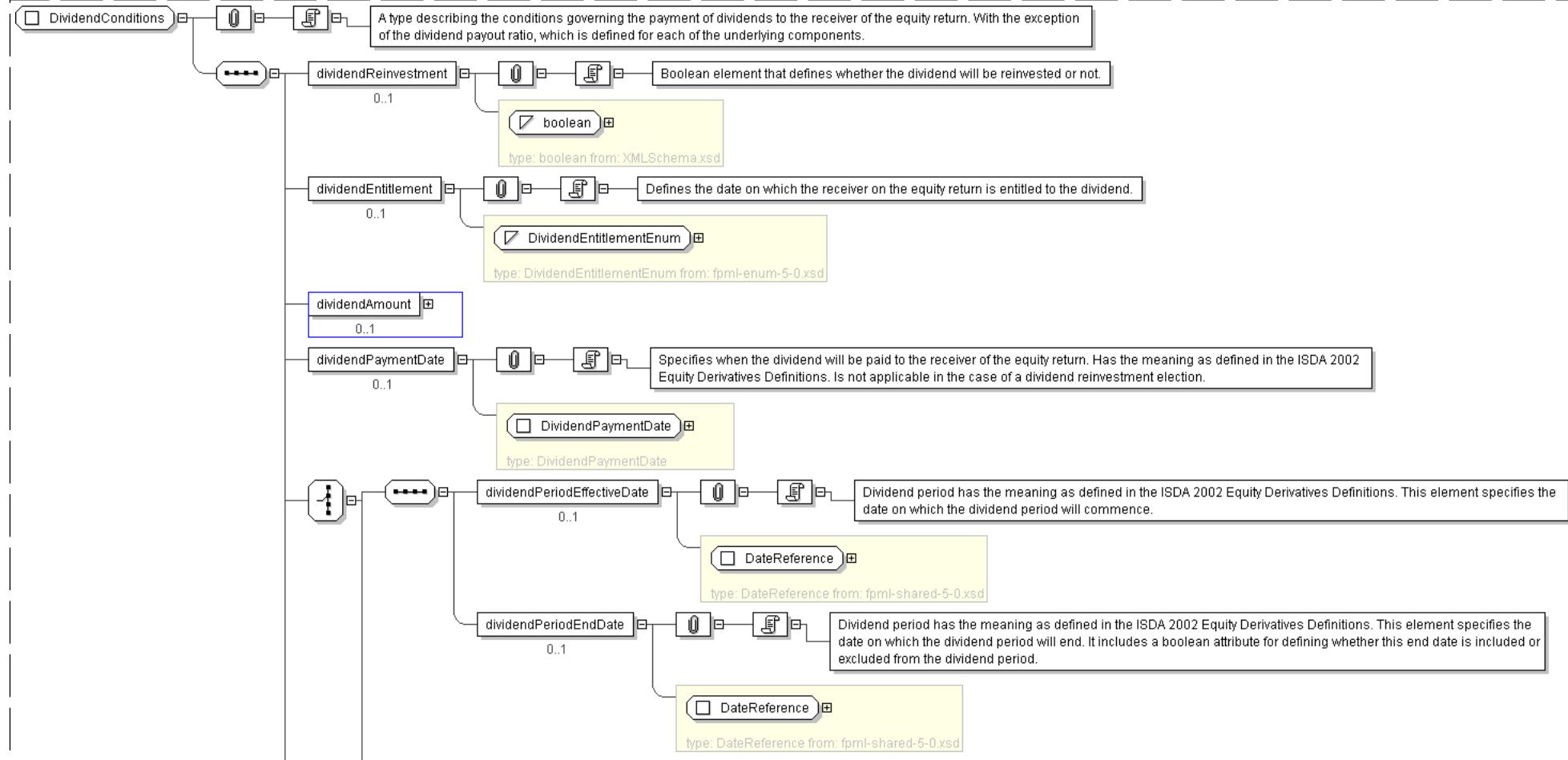
'Defines how the composition of Dividends is to be determined.'

<specialDividends> [xsd:boolean](#) </specialDividends> [0..1]

'Specifies the method according to which special dividends are determined.'

</...>

#### Diagram



**Schema Component Representation**

```

<xsd:complexType name="DividendConditions">
  <xsd:sequence>
    <xsd:element name="dividendReinvestment" type="xsd:boolean" minOccurs="0"/>
    <xsd:element name="dividendEntitlement" type="DividendEntitlementEnum" minOccurs="0"/>
    <xsd:element name="dividendAmount" type="DividendAmountTypeEnum" minOccurs="0"/>
    <xsd:element name="dividendPaymentDate" type="DividendPaymentDate" minOccurs="0"/>
    <xsd:choice>
      <xsd:sequence>
        <xsd:element name="dividendPeriodEffectiveDate" type="DateReference" minOccurs="0"/>
        <xsd:element name="dividendPeriodEndDate" type="DateReference" minOccurs="0"/>
      </xsd:sequence>
      <xsd:element name="dividendPeriod" type="DividendPeriodEnum" />
    </xsd:choice>
    <xsd:element name="extraOrdinaryDividends" type="PartyReference" minOccurs="0"/>
    <xsd:element name="excessDividendAmount" type="DividendAmountTypeEnum" minOccurs="0"/>
    <xsd:group ref="CurrencyAndDeterminationMethod.model" minOccurs="0"/>
    <xsd:element name="dividendFxTriggerDate" type="DividendPaymentDate" minOccurs="0"/>
    <xsd:element name="interestAccrualsMethod" type="InterestAccrualsCompoundingMethod"
      " minOccur="0"/>
    <xsd:element name="numberOfWorkUnits" type="NonNegativeDecimal" minOccurs="0"/>
    <xsd:group ref="DeclaredCashAndCashEquivalentDividendPercentage.model" />
    <xsd:element name="nonCashDividendTreatment" type="NonCashDividendTreatmentEnum"
      " minOccurs="0"/>
    <xsd:element name="dividendComposition" type="DividendCompositionEnum" minOccurs="0"/>
    <xsd:element name="specialDividends" type="xsd:boolean" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>

```

[top](#)

## Complex Type: DividendPaymentDate

Super-types:	None
Sub-types:	None

Name	DividendPaymentDate
Used by (from the same schema document)	Complex Type <a href="#">DividendConditions</a> , Complex Type <a href="#">DividendConditions</a>
Abstract	no
Documentation	A type describing the date on which the dividend will be paid/received. This type is also used to specify the date on which the FX rate will be determined, when applicable.

### XML Instance Representation

```

<...>
Start Choice [1]
  <dividendDateReference> DividendDateReferenceEnum </dividendDateReference> [1]
  'Specification of the dividend date using an enumeration, with values such as the pay date,
  the ex date or the record date.'

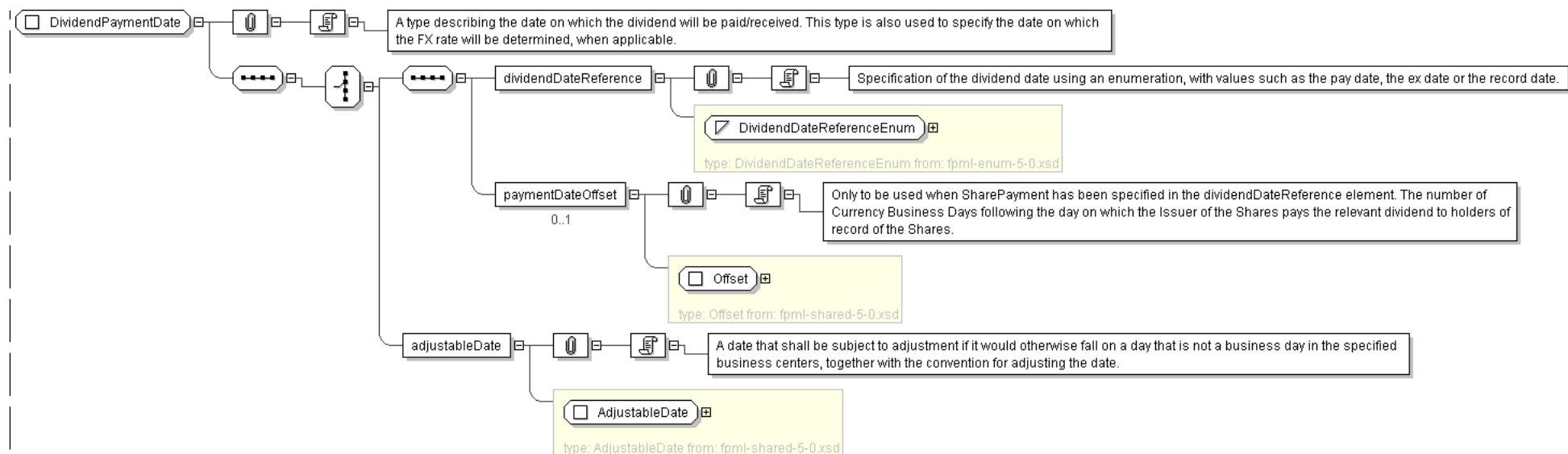
  <paymentDateOffset> Offset </paymentDateOffset> [0..1]
  'Only to be used when SharePayment has been specified in the dividendDateReference element.
  The number of Currency Business Days following the day on which the Issuer of the Shares
  pays the relevant dividend to holders of record of the Shares.'

  <adjustableDate> AdjustableDate </adjustableDate> [1]
  '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.'

End Choice
</...>

```

### Diagram



#### Schema Component Representation

```

<xsd:complexType name="DividendPaymentDate">
  <xsd:sequence>
    <xsd:choice>
      <xsd:sequence>
        <xsd:element name="dividendDateReference" type="DividendDateReferenceEnum" />
        <xsd:element name="paymentDateOffset" type="Offset" minOccurs="0" />
      </xsd:sequence>
      <xsd:element name="adjustableDate" type="AdjustableDate" />
    </xsd:choice>
  </xsd:sequence>
</xsd:complexType>

```

top

#### Complex Type: `DividendPeriod`

Super-types:	None
Sub-types:	<ul style="list-style-type: none"> <li><code>DividendPeriodDividend</code> (by extension)</li> </ul>

Name	DividendPeriod
Abstract	yes
Documentation	Abstract base class of all time bounded dividend period types.

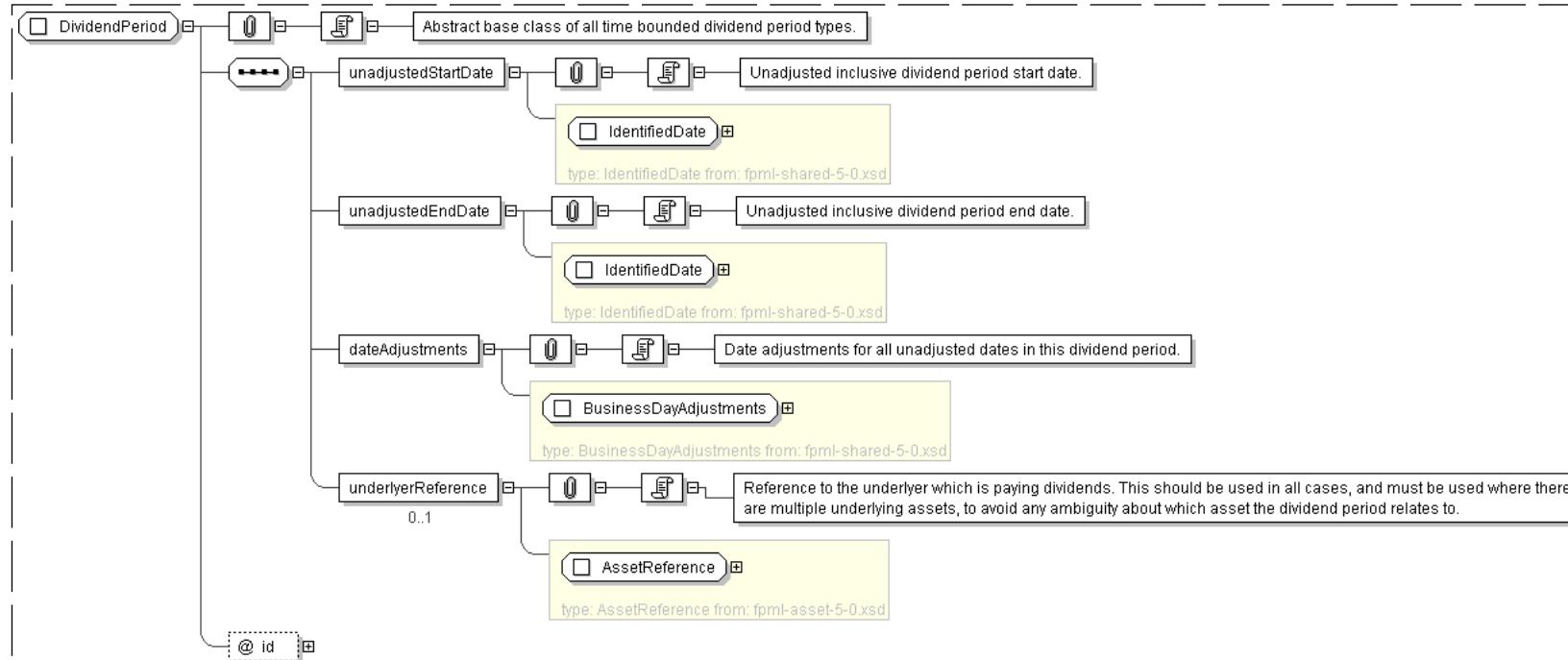
#### XML Instance Representation

```

<...>
<id id=" [0..1]">
  <unadjustedStartDate> IdentifiedDate </unadjustedStartDate> [1]
  'Unadjusted inclusive dividend period start date.'
  <unadjustedEndDate> IdentifiedDate </unadjustedEndDate> [1]
  'Unadjusted inclusive dividend period end date.'
  <dateAdjustments> BusinessDayAdjustments </dateAdjustments> [1]
  'Date adjustments for all unadjusted dates in this dividend period.'
  <underlyerReference> AssetReference </underlyerReference> [0..1]
  'Reference to the underlyer which is paying dividends. This should be used in all cases,'
```

and must be used where there are multiple underlying assets, to avoid any ambiguity about which asset the dividend period relates to.'

&lt;/...&gt;

**Diagram****Schema Component Representation**

```

<xsd:complexType name="DividendPeriod" abstract="true">
    <xsd:sequence>
        <xsd:element name="unadjustedStartDate" type="IdentifiedDate" />
        <xsd:element name="unadjustedEndDate" type="IdentifiedDate" />
        <xsd:element name="dateAdjustments" type="BusinessDayAdjustments" />
        <xsd:element name="underlyerReference" type="AssetReference" minOccurs="0" />
    </xsd:sequence>
    <xsd:attribute name="id" type="xsd:ID" />
</xsd:complexType>

```

top

**Complex Type: DividendPeriodDividend**

<b>Super-types:</b>	<a href="#">DividendPeriod</a> < <b>DividendPeriodDividend</b> (by extension)
<b>Sub-types:</b>	None
<b>Name</b>	DividendPeriodDividend
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">DividendAdjustment</a>
<b>Abstract</b>	no
<b>Documentation</b>	A time bounded dividend period, with an expected dividend for each period.

**XML Instance Representation**

```

<...
id="xsd:ID [0..1]">

```

```

<unadjustedStartDate> IdentifiedDate </unadjustedStartDate> [1]
'Unadjusted inclusive dividend period start date.'

<unadjustedEndDate> IdentifiedDate </unadjustedEndDate> [1]
'Unadjusted inclusive dividend period end date.'

<dateAdjustments> BusinessDayAdjustments </dateAdjustments> [1]
'Date adjustments for all unadjusted dates in this dividend period.'

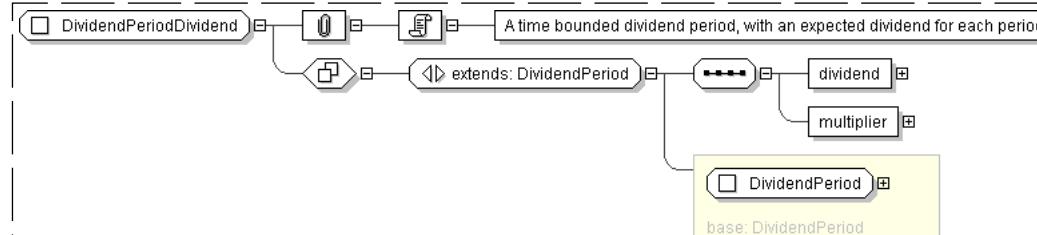
<underlyerReference> AssetReference </underlyerReference> [0..1]
'Reference to the underlyer which is paying dividends. This should be used in all cases,
and must be used where there are multiple underlying assets, to avoid any ambiguity about
which asset the dividend period relates to.'

<dividend> NonNegativeMoney </dividend> [1]
'Expected dividend in this period.'

<multiplier> PositiveDecimal </multiplier> [1]
'Multiplier is a percentage value which is used to produce Deviation by multiplying
the difference between Expected Dividend and Actual Dividend Deviation = Multiplier *
(Expected Dividend - Actual Dividend).'

</...>

```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="DividendPeriodDividend">
  <xsd:complexContent>
    <xsd:extension base="DividendPeriod">
      <xsd:sequence>
        <xsd:element name="dividend" type="NonNegativeMoney" />
        <xsd:element name="multiplier" type="PositiveDecimal" />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

top

**Complex Type: EquityCorporateEvents**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None
<b>Name</b>	EquityCorporateEvents
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">ExtraordinaryEvents</a> , Complex Type <a href="#">ExtraordinaryEvents</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type for defining the merger events and their treatment.

**XML Instance Representation**

```

<...>
| <shareForShare> ShareExtraordinaryEventEnum </shareForShare> [1]

```

'The consideration paid for the original shares following the Merger Event consists wholly of new shares.'

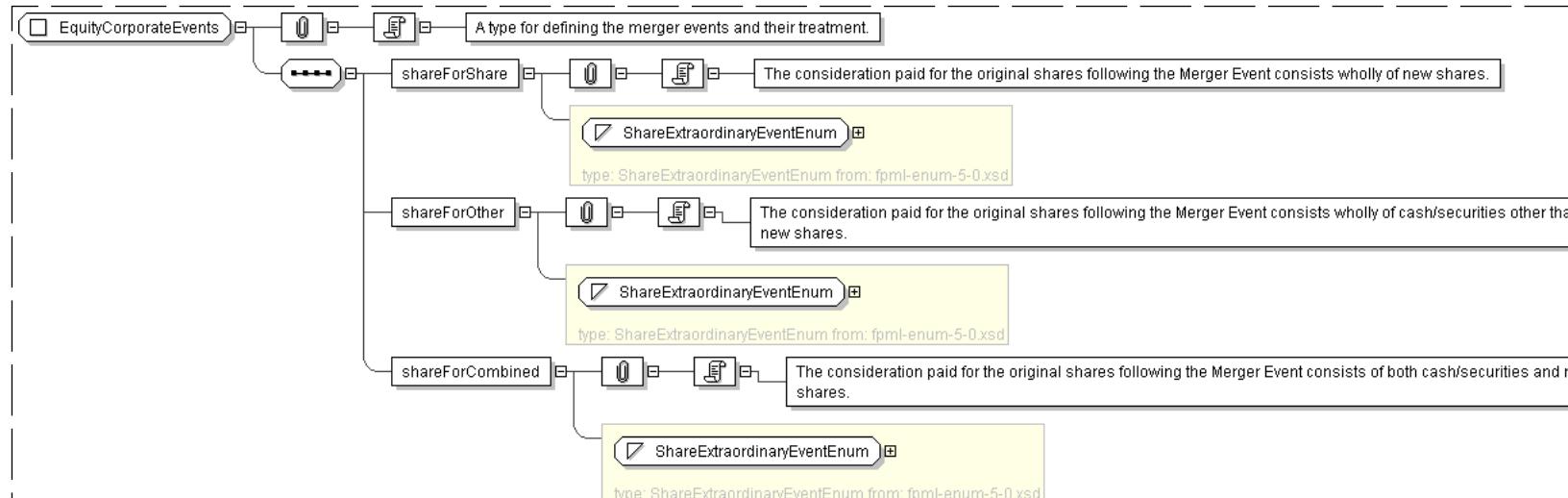
```
<shareForOther> ShareExtraordinaryEventEnum </shareForOther> [1]
```

'The consideration paid for the original shares following the Merger Event consists wholly of cash/securities other than new shares.'

```
<shareForCombined> ShareExtraordinaryEventEnum </shareForCombined> [1]
```

'The consideration paid for the original shares following the Merger Event consists of both cash/securities and new shares.'

&lt;/...&gt;

**Diagram****Schema Component Representation**

```

<xsd:complexType name="EquityCorporateEvents">
  <xsd:sequence>
    <xsd:element name="shareForShare" type=" ShareExtraordinaryEventEnum " />
    <xsd:element name="shareForOther" type=" ShareExtraordinaryEventEnum " />
    <xsd:element name="shareForCombined" type=" ShareExtraordinaryEventEnum " />
  </xsd:sequence>
</xsd:complexType>
  
```

top

**Complex Type: [EquityPremium](#)**

<b>Super-types:</b>	<a href="#">PaymentBase</a> < <b>EquityPremium</b> (by extension)
---------------------	---

<b>Sub-types:</b>	None
-------------------	------

<b>Name</b>	EquityPremium
-------------	---------------

<b>Abstract</b>	no
-----------------	----

<b>Documentation</b>	A type used to describe the amount paid for an equity option.
----------------------	---

**XML Instance Representation**

```

<...
id=" xsd:ID [0..1]">
  <payerPartyReference> PartyReference </payerPartyReference> [1]
  'A reference to the party responsible for making the payments defined by this structure.'
  
```

```

<payerAccountReference> AccountReference </payerAccountReference> [0..1]
'A reference to the account responsible for making the payments defined by this structure.'

<receiverPartyReference> PartyReference </receiverPartyReference> [1]
'A reference to the party that receives the payments corresponding to this structure.'

<receiverAccountReference> AccountReference </receiverAccountReference> [0..1]
'A reference to the account that receives the payments corresponding to this structure.'

<premiumType> PremiumTypeEnum </premiumType> [0..1]
'Forward start Premium type'

<paymentAmount> NonNegativeMoney </paymentAmount> [0..1]
'The currency amount of the payment.'

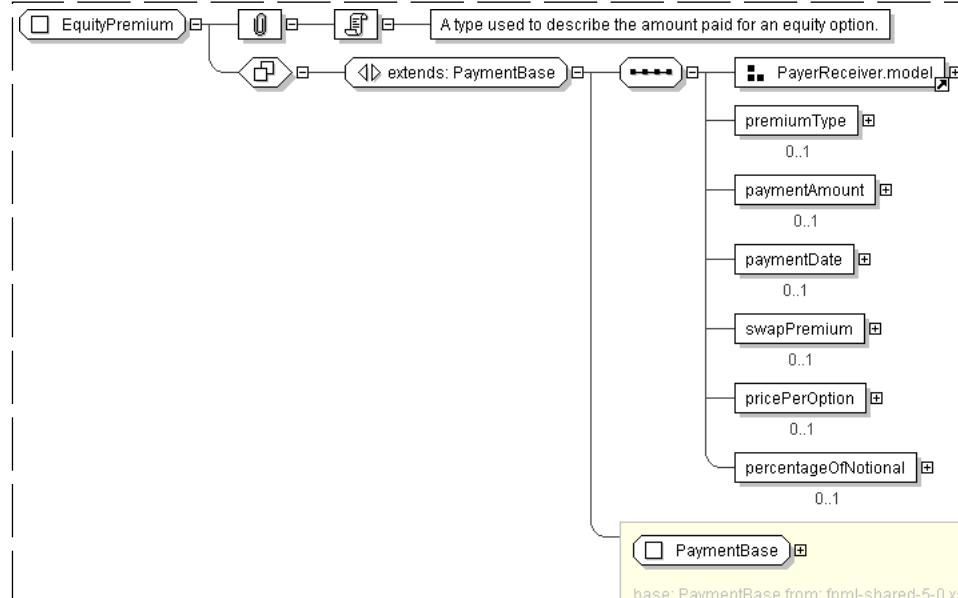
<paymentDate> AdjustableDate </paymentDate> [0..1]
'The payment date. This date is subject to adjustment in accordance with any applicable business day convention.'

<swapPremium> xsd:boolean </swapPremium> [0..1]
'Specifies whether or not the premium is to be paid in the style of payments under an interest rate swap contract.'

<pricePerOption> NonNegativeMoney </pricePerOption> [0..1]
'The amount of premium to be paid expressed as a function of the number of options.'

<percentageOfNotional> NonNegativeDecimal </percentageOfNotional> [0..1]
'The amount of premium to be paid expressed as a percentage of the notional value of the transaction. A percentage of 5% would be expressed as 0.05.'
```

&lt;/...&gt;

**Diagram****Schema Component Representation**

```

<xsd:complexType name="EquityPremium">
  <xsd:complexContent>
    <xsd:extension base=" PaymentBase ">
```

```

<xsd:sequence>
  <xsd:group ref=" PayerReceiver.model " />
  <xsd:element name="premiumType" type=" PremiumTypeEnum " minOccurs="0" />
  <xsd:element name="paymentAmount" type=" NonNegativeMoney " minOccurs="0" />
  <xsd:element name="paymentDate" type=" AdjustableDate " minOccurs="0" />
  <xsd:element name="swapPremium" type=" xsd:boolean " minOccurs="0" />
  <xsd:element name="pricePerOption" type=" NonNegativeMoney " minOccurs="0" />
  <xsd:element name="percentageOfNotional" type=" NonNegativeDecimal " minOccurs="0" />
</xsd:sequence>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>

```

## Complex Type: **EquityStrike**

**Super-types:** None  
**Sub-types:** None

<b>Name</b>	EquityStrike
<b>Abstract</b>	no
<b>Documentation</b>	A type for defining the strike price for an equity option. The strike price is either: (i) in respect of an index option transaction, the level of the relevant index specified or otherwise determined in the transaction; or (ii) in respect of a share option transaction, the price per share specified or otherwise determined in the transaction. This can be expressed either as a percentage of notional amount or as an absolute value.

### XML Instance Representation

```

<...>
Start Choice [1]
  <strikePrice> xsd:decimal </strikePrice> [1]
    'The price or level at which the option has been struck.'

  <strikePercentage> xsd:decimal </strikePercentage> [1]
    'The price or level expressed as a percentage of the forward starting spot price.'

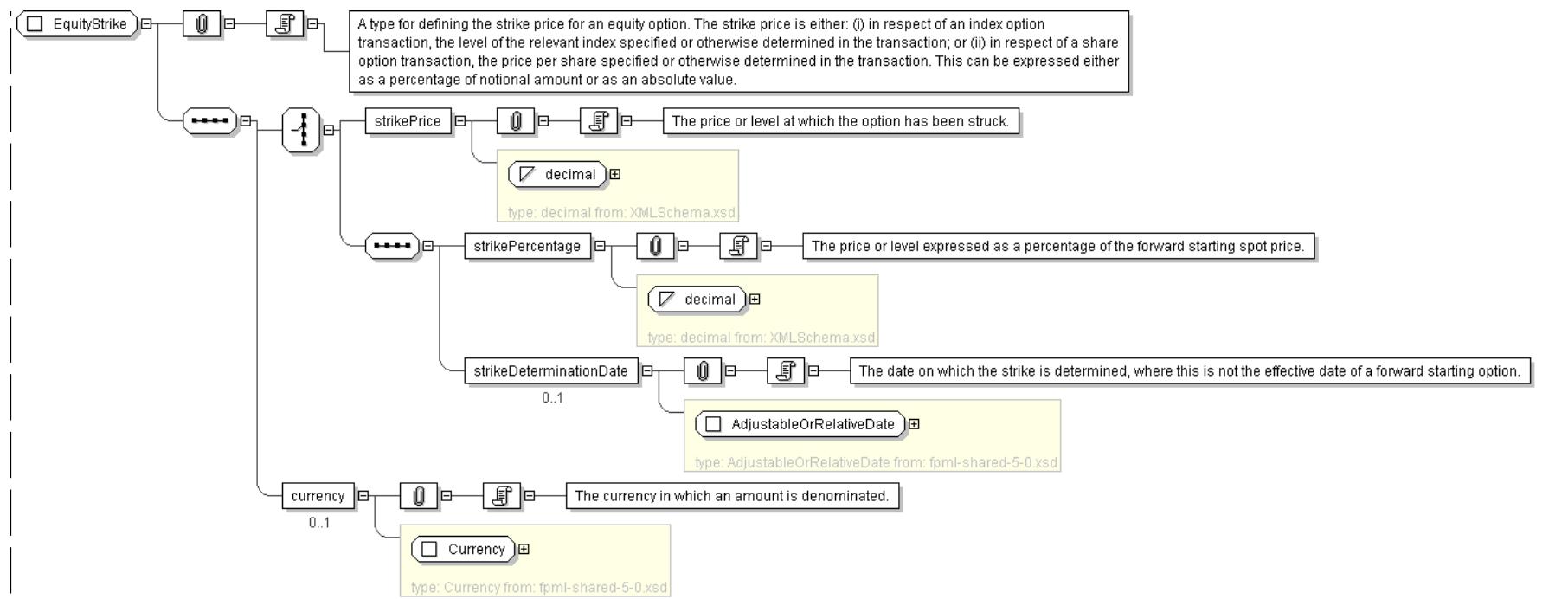
  <strikeDeterminationDate> AdjustableOrRelativeDate </strikeDeterminationDate> [0..1]
    'The date on which the strike is determined, where this is not the effective date of a
forward starting option.'

End Choice
  <currency> Currency </currency> [0..1]
    'The currency in which an amount is denominated.'

</...>

```

### Diagram

**Schema Component Representation**

```

<xsd:complexType name="EquityStrike">
  <xsd:sequence>
    <xsd:choice>
      <xsd:element name="strikePrice" type="xsd:decimal" />
      <xsd:sequence>
        <xsd:element name="strikePercentage" type="xsd:decimal" />
        <xsd:element name="strikeDeterminationDate" type="AdjustableOrRelativeDate" minOccurs="0" />
      </xsd:sequence>
    </xsd:choice>
    <xsd:element name="currency" type="Currency" minOccurs="0" />
  </xsd:sequence>
</xsd:complexType>

```

top

**Complex Type: EquityValuation**

Super-types:	None
Sub-types:	None
Name	EquityValuation
Used by (from the same schema document)	Complex Type <a href="#">DirectionalLegUnderlyerValuation</a> , Complex Type <a href="#">ReturnLegValuationPrice</a>
Abstract	no
Documentation	A type for defining how and when an equity option is to be valued.

**XML Instance Representation**

```

<...>
<id=" xsd:ID [0..1]">
Start Choice [0..1]
  <valuationDate> AdjustableDateOrRelativeDateSequence </valuationDate> [1]
  'The term \'Valuation Date\' is assumed to have the meaning as defined in the ISDA 2002
  Equity Derivatives Definitions.'

```

```

<valuationDates> AdjustableRelativeOrPeriodicDates </valuationDates> [1]
'Specifies the interim equity valuation dates of a swap.'

End Choice
<valuationTimeType> TimeTypeEnum </valuationTimeType> [0..1]
'The time of day at which the calculation agent values the underlying, for example the
official closing time of the exchange.'

<valuationTime> BusinessCenterTime </valuationTime> [0..1]
'The specific time of day at which the calculation agent values the underlying..'

<futuresPriceValuation> xsd:boolean </futuresPriceValuation> [0..1]
'The official settlement price as announced by the related exchange is applicable,
in accordance with the ISDA 2002 definitions.'

<optionsPriceValuation> xsd:boolean </optionsPriceValuation> [0..1]
'The official settlement price as announced by the related exchange is applicable,
in accordance with the ISDA 2002 definitions..'

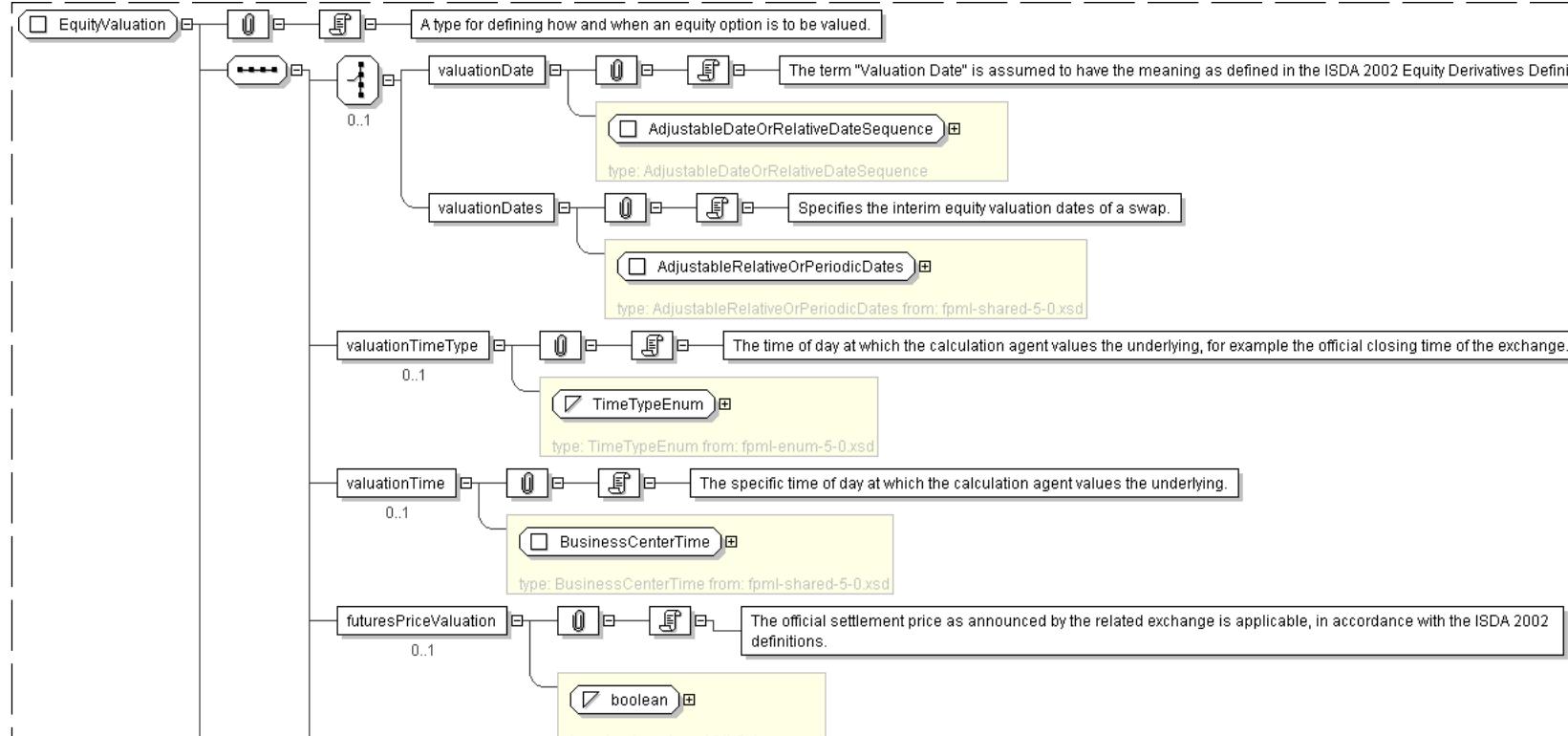
<numberOfValuationDates> xsd:nonNegativeInteger </numberOfValuationDates> [0..1]
'The number of valuation dates between valuation start date and valuation end date..'

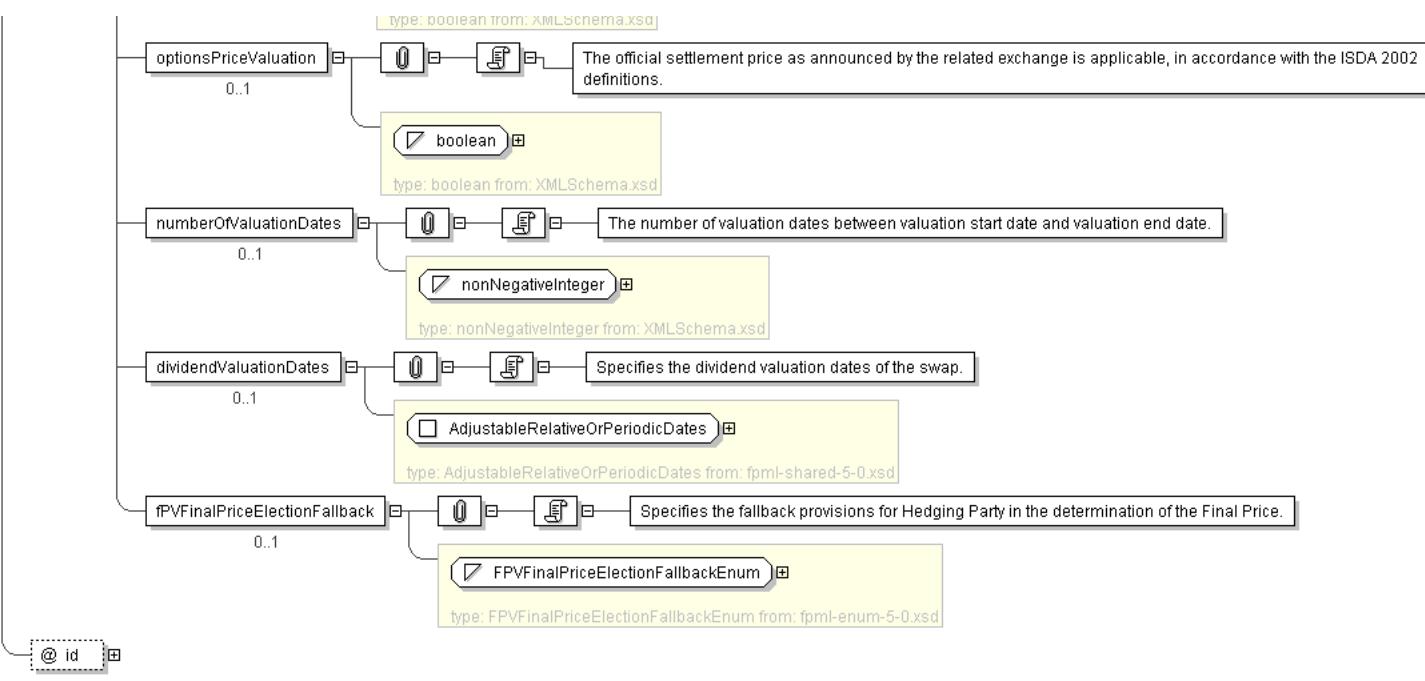
<dividendValuationDates> AdjustableRelativeOrPeriodicDates </dividendValuationDates> [0..1]
'Specifies the dividend valuation dates of the swap.'

<fPVPFinalPriceElectionFallback> FpVFinalPriceElectionFallbackEnum
</fPVPFinalPriceElectionFallback> [0..1]
'Specifies the fallback provisions for Hedging Party in the determination of the Final Price.'

```

&lt;/...&gt;

**Diagram**

**Schema Component Representation**

```

<xsd:complexType name="EquityValuation">
  <xsd:sequence>
    <xsd:choice minOccurs="0">
      <xsd:element name="valuationDate" type=" AdjustableDateOrRelativeDateSequence " />
      <xsd:element name="valuationDates" type=" AdjustableRelativeOrPeriodicDates " />
    </xsd:choice>
    <xsd:element name="valuationTimeType" type=" TimeTypeEnum " minOccurs="0" />
    <xsd:element name="valuationTime" type=" BusinessCenterTime " minOccurs="0" />
    <xsd:element name="futuresPriceValuation" type=" xsd:boolean " minOccurs="0" />
    <xsd:element name="optionsPriceValuation" type=" xsd:boolean " minOccurs="0" />
    <xsd:element name="numberOfValuationDates" type=" xsd:nonNegativeInteger " minOccurs="0" />
    <xsd:element name="dividendValuationDates" type=" AdjustableRelativeOrPeriodicDates "
      " minOccurs="0" />
    <xsd:element name="fPVFinalPriceElectionFallback" type=" FPVFinalPriceElectionFallbackEnum "
      " minOccurs="0" />
  </xsd:sequence>
  <xsd:attribute name="id" type=" xsd:ID " />
</xsd:complexType>

```

top

**Complex Type: ExtraordinaryEvents**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None

<b>Name</b>	ExtraordinaryEvents
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">NettedSwapBase</a> , Complex Type <a href="#">ReturnSwap</a>
<b>Abstract</b>	no
<b>Documentation</b>	Where the underlying is shares, defines market events affecting the issuer of those shares that may require the terms of the transaction to be adjusted.

**XML Instance Representation**

```

< . . . >
|
```

```
<mergerEvents> EquityCorporateEvents </mergerEvents> [0..1]
'Occurs when the underlying ceases to exist following a merger between the Issuer and
another company.'
```

```
<tenderOffer> xsd:boolean </tenderOffer> [0..1]
'If present and true, then tender offer is applicable.'
```

```
<tenderOfferEvents> EquityCorporateEvents </tenderOfferEvents> [0..1]
'ISDA 2002 Equity Tender Offer Events.'
```

```
<compositionOfCombinedConsideration> xsd:boolean </compositionOfCombinedConsideration> [0..1]
'If present and true, then composition of combined consideration is applicable.'
```

```
<indexAdjustmentEvents> IndexAdjustmentEvents </indexAdjustmentEvents> [0..1]
'ISDA 2002 Equity Index Adjustment Events.'
```

```
Start Choice [1]
<additionalDisruptionEvents> AdditionalDisruptionEvents </additionalDisruptionEvents> [1]
'ISDA 2002 Equity Additional Disruption Events.'
```

```
<failureToDeliver> xsd:boolean </failureToDeliver> [1]
'If true, failure to deliver is applicable.'
```

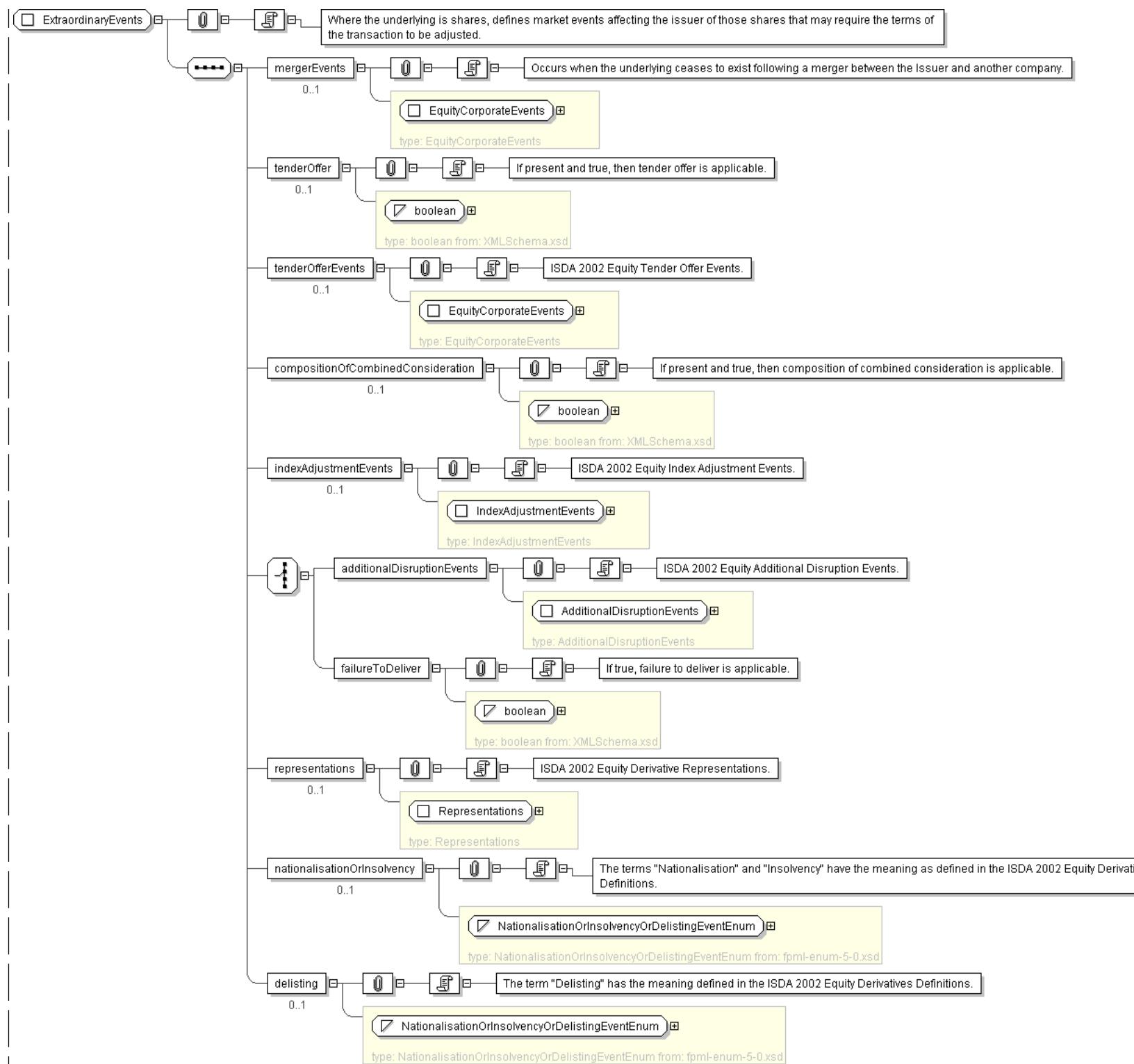
```
End Choice
<representations> Representations </representations> [0..1]
'ISDA 2002 Equity Derivative Representations.'
```

```
<nationalisationOrInsolvency> NationalisationOrInsolvencyOrDelistingEventEnum
</nationalisationOrInsolvency> [0..1]
'The terms \"Nationalisation\" and \"Insolvency\" have the meaning as defined in the ISDA
2002 Equity Derivatives Definitions.'
```

```
<delisting> NationalisationOrInsolvencyOrDelistingEventEnum </delisting> [0..1]
'The term \"Delisting\" has the meaning defined in the ISDA 2002 Equity
Derivatives Definitions.'
```

```
</...>
```

**Diagram**



**Schema Component Representation**

```
<xsd:complexType name="ExtraordinaryEvents">
  <xsd:sequence>
    <xsd:element name="mergerEvents" type=" EquityCorporateEvents " minOccurs="0" />
    <xsd:element name="tenderOffer" type=" xsd:boolean " minOccurs="0" />
    <xsd:element name="tenderOfferEvents" type=" EquityCorporateEvents " minOccurs="0" />
    <xsd:element name="compositionOfCombinedConsideration" type=" xsd:boolean " minOccurs="0" />
    <xsd:element name="indexAdjustmentEvents" type=" IndexAdjustmentEvents " minOccurs="0" />
    <xsd:choice>
      <xsd:element name="additionalDisruptionEvents" type=" AdditionalDisruptionEvents " />
      <xsd:element name="failureToDeliver" type=" xsd:boolean " />
    </xsd:choice>
    <xsd:element name="representations" type=" Representations " minOccurs="0" />
    <xsd:element name="nationalisationOrInsolvency"
      type=" NationalisationOrInsolvencyOrDelistingEventEnum " minOccurs="0" />
    <xsd:element name="delisting" type=" NationalisationOrInsolvencyOrDelistingEventEnum "
      minOccurs="0" />
  </xsd:sequence>
</xsd:complexType>
```

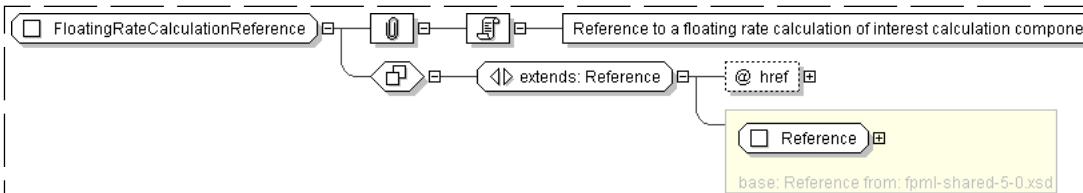
[top](#)**Complex Type: FloatingRateCalculationReference**

<b>Super-types:</b>	<a href="#">Reference</a> < <b>FloatingRateCalculationReference</b> (by extension)
<b>Sub-types:</b>	None

<b>Name</b>	FloatingRateCalculationReference
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">CompoundingRate</a>
<b>Abstract</b>	no
<b>Documentation</b>	Reference to a floating rate calculation of interest calculation component.

**XML Instance Representation**

```
<...
  href=" xsd:IDREF [1]" />
```

**Diagram****Schema Component Representation**

```
<xsd:complexType name="FloatingRateCalculationReference">
  <xsd:complexContent>
    <xsd:extension base=" Reference ">
      <xsd:attribute name="href" type=" xsd:IDREF "
        use="required" reference="floatingRateCalculation" />
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

[top](#)**Complex Type: IndexAdjustmentEvents**

<b>Super-types:</b>	None
---------------------	------

Sub-types: None

Name	IndexAdjustmentEvents
Used by (from the same schema document)	Complex Type <a href="#">ExtraordinaryEvents</a>
Abstract	no
Documentation	Defines the specification of the consequences of Index Events as defined by the 2002 ISDA Equity Derivatives Definitions.

#### XML Instance Representation

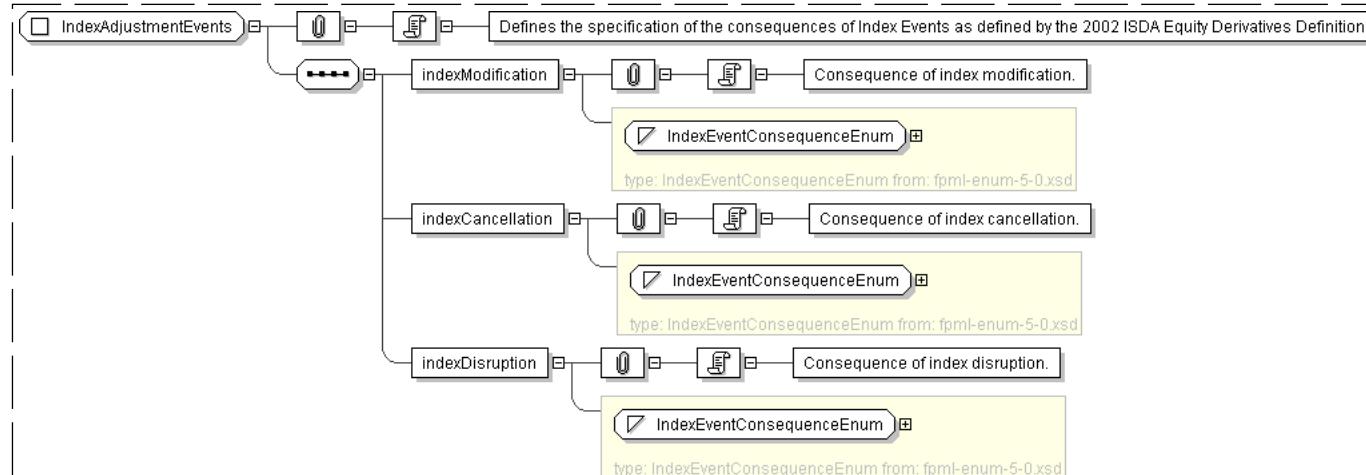
```
<...>
<indexModification> IndexEventConsequenceEnum </indexModification> [1]
'Consequence of index modification.'

<indexCancellation> IndexEventConsequenceEnum </indexCancellation> [1]
'Consequence of index cancellation.'

<indexDisruption> IndexEventConsequenceEnum </indexDisruption> [1]
'Consequence of index disruption.'

</...>
```

#### Diagram



#### Schema Component Representation

```

<xsd:complexType name="IndexAdjustmentEvents">
  <xsd:sequence>
    <xsd:element name="indexModification" type=" IndexEventConsequenceEnum " />
    <xsd:element name="indexCancellation" type=" IndexEventConsequenceEnum " />
    <xsd:element name="indexDisruption" type=" IndexEventConsequenceEnum " />
  </xsd:sequence>
</xsd:complexType>

```

top

#### Complex Type: [InterestCalculation](#)

Super-types:

[InterestAccrualsMethod](#) < [InterestCalculation](#) (by extension)

Sub-types:

None

Name	InterestCalculation
Used by (from the same schema document)	Complex Type <a href="#">InterestLeg</a>
Abstract	no

**Documentation**

Specifies the calculation method of the interest rate leg of the return swap. Includes the floating or fixed rate calculation definitions, along with the determination of the day count fraction.

**XML Instance Representation**

```
<...>
  id=" xsd:ID [0..1]">
  Start Choice [1]
    <floatingRateCalculation> FloatingRateCalculation </floatingRateCalculation> [1]
    'The floating rate calculation definitions'

    <fixedRate> xsd:decimal </fixedRate> [1]
    'The calculation period fixed rate. A per annum rate, expressed as a decimal. A fixed rate
    of 5% would be represented as 0.05.'

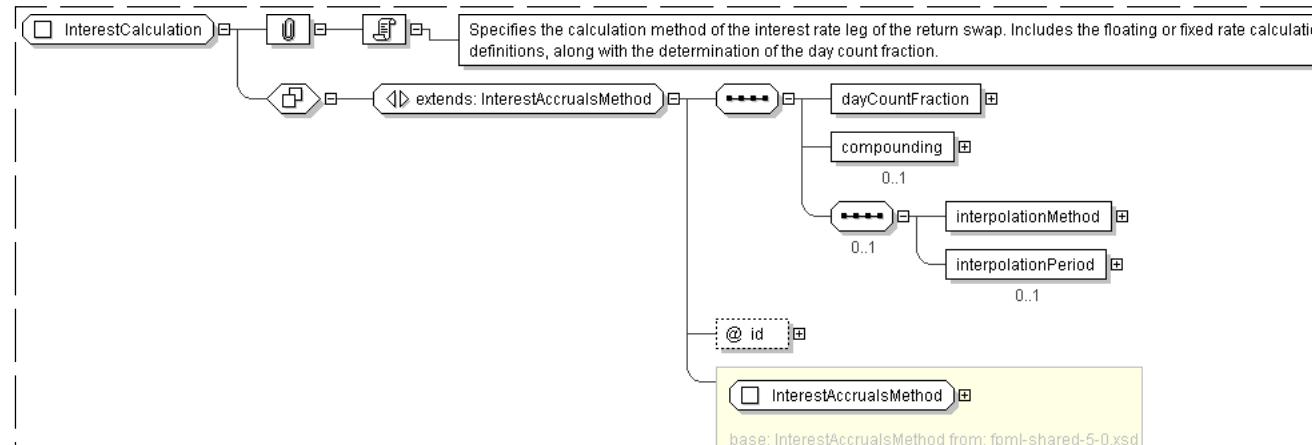
  End Choice
  <dayCountFraction> DayCountFraction </dayCountFraction> [1]
  'The day count fraction.'

  <compounding> Compounding </compounding> [0..1]
  'Defines compounding rates on the Interest Leg.'

Start Sequence [0..1]
  <interpolationMethod> InterpolationMethod </interpolationMethod> [1]
  'Specifies the type of interpolation used.'

  <interpolationPeriod> InterpolationPeriodEnum </interpolationPeriod> [0..1]
  'Defines applicable periods for interpolation.'

End Sequence
</...>
```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="InterestCalculation">
  <xsd:complexContent>
    <xsd:extension base=" InterestAccrualsMethod ">
      <xsd:sequence>
        <xsd:element name="dayCountFraction" type=" DayCountFraction "/>
        <xsd:element name="compounding" type=" Compounding " minOccurs="0"/>
        <xsd:sequence minOccurs="0">
          <xsd:element name="interpolationMethod" type=" InterpolationMethod "/>
          <xsd:element name="interpolationPeriod" type=" InterpolationPeriodEnum " minOccurs="0"/>
        </xsd:sequence>
      </xsd:sequence>
      <xsd:attribute name="id" type=" xsd:ID "/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
  
```

```
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>
```

**Complex Type: InterestLeg**

**Super-types:** Leg < DirectionalLeg (by extension) < InterestLeg (by extension)

**Sub-types:** None

<b>Name</b>	InterestLeg
<b>Used by (from the same schema document)</b>	Element <a href="#">interestLeg</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type describing the fixed income leg of the equity swap.

**XML Instance Representation**

```
<...>
<id="xsd:ID [0..1]">
  <legIdentifier> LegIdentifier </legIdentifier> [0..*]
  'Version aware identification of this leg.'

  <payerPartyReference> PartyReference </payerPartyReference> [1]
  'A reference to the party responsible for making the payments defined by this structure.'

  <payerAccountReference> AccountReference </payerAccountReference> [0..1]
  'A reference to the account responsible for making the payments defined by this structure.'

  <receiverPartyReference> PartyReference </receiverPartyReference> [1]
  'A reference to the party that receives the payments corresponding to this structure.'

  <receiverAccountReference> AccountReference </receiverAccountReference> [0..1]
  'A reference to the account that receives the payments corresponding to this structure.'

  <effectiveDate> AdjustableOrRelativeDate </effectiveDate> [0..1]
  'Specifies the effective date of this leg of the swap. When defined in relation to a date specified somewhere else in the document (through the relativeDate component), this element will typically point to the effective date of the other leg of the swap.'

  <terminationDate> AdjustableOrRelativeDate </terminationDate> [0..1]
  'Specifies the termination date of this leg of the swap. When defined in relation to a date specified somewhere else in the document (through the relativeDate component), this element will typically point to the termination date of the other leg of the swap.'

  <interestLegCalculationPeriodDates> InterestLegCalculationPeriodDates
  </interestLegCalculationPeriodDates> [1]
  'Component that holds the various dates used to specify the interest leg of the equity swap. It is used to define the InterestPeriodDates identifier.'

  <notional> ReturnSwapNotional </notional> [1]
  'Specifies the notional of a return type swap. When used in the equity leg, the definition will typically combine the actual amount (using the notional component defined by the FpML industry group) and the determination method. When used in the interest leg, the definition will typically point to the definition of the equity leg.'

  <interestAmount> LegAmount </interestAmount> [1]
  'Specifies, in relation to each Interest Payment Date, the amount to which the Interest Payment Date relates. Unless otherwise specified, this term has the meaning defined in the ISDA 2000 ISDA Definitions.'

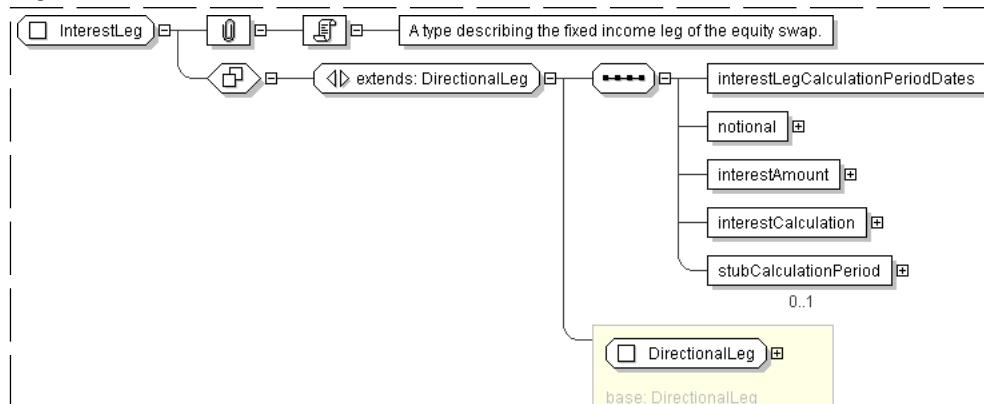
  <interestCalculation> InterestCalculation </interestCalculation> [1]
  'Specifies the calculation method of the interest rate leg of the equity swap. Includes the floating or fixed rate calculation definitions, along with the determination of the day count fraction.'
```

```
<stubCalculationPeriod> StubCalculationPeriod </stubCalculationPeriod> [0..1]
```

'Specifies the stub calculation period.'

</...>

#### Diagram



#### Schema Component Representation

```

<xsd:complexType name="InterestLeg">
  <xsd:complexContent>
    <xsd:extension base="DirectionalLeg ">
      <xsd:sequence>
        <xsd:element name="interestLegCalculationPeriodDates" type="InterestLegCalculationPeriodDates " />
        <xsd:element name="notional" type="ReturnSwapNotional " />
        <xsd:element name="interestAmount" type="LegAmount " />
        <xsd:element name="interestCalculation" type="InterestCalculation " />
        <xsd:element name="stubCalculationPeriod" type="StubCalculationPeriod " minOccurs="0"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
  
```

[top](#)

#### Complex Type: InterestLegCalculationPeriodDates

Super-types:	None
Sub-types:	None

Name	InterestLegCalculationPeriodDates
Used by (from the same schema document)	Complex Type <a href="#">InterestLeg</a>
Abstract	no
Documentation	Component that holds the various dates used to specify the interest leg of the return swap. It is used to define the InterestPeriodDates identifier.

#### XML Instance Representation

```

<...
  id="xsd:ID [1]">
  <effectiveDate> AdjustableOrRelativeDate </effectiveDate> [1]
  'Specifies the effective date of the return swap. This global element is valid within
  the return swaps namespace. Within the FpML namespace, another effectiveDate global element
  has been defined, that is different in the sense that it does not propose the choice
  of referring to another date in the document.'
  <terminationDate> AdjustableOrRelativeDate </terminationDate> [1]
  
```

'Specifies the termination date of the return swap. This global element is valid within the return swaps namespace. Within the FpML namespace, another terminationDate global element has been defined, that is different in the sense that it does not propose the choice of referring to another date in the document.'

<interestLegResetDates> [InterestLegResetDates](#) </interestLegResetDates> [1]

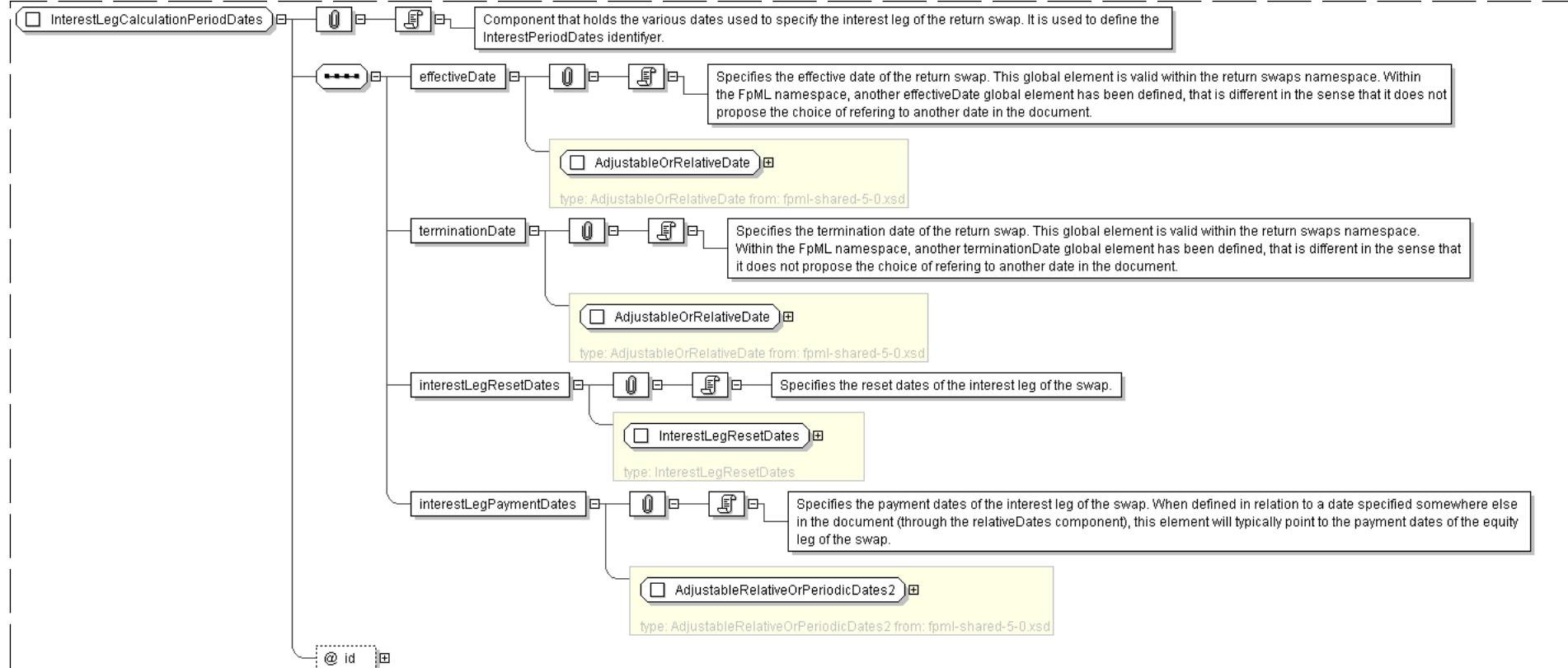
'Specifies the reset dates of the interest leg of the swap.'

<interestLegPaymentDates> [AdjustableRelativeOrPeriodicDates2](#) </interestLegPaymentDates> [1]

'Specifies the payment dates of the interest leg of the swap. When defined in relation to a date specified somewhere else in the document (through the relativeDates component), this element will typically point to the payment dates of the equity leg of the swap.'

<...>

#### Diagram



#### Schema Component Representation

```

<xsd:complexType name="InterestLegCalculationPeriodDates">
  <xsd:sequence>
    <xsd:element name="effectiveDate" type="AdjustableOrRelativeDate" />
    <xsd:element name="terminationDate" type="AdjustableOrRelativeDate" />
    <xsd:element name="interestLegResetDates" type="InterestLegResetDates" />
    <xsd:element name="interestLegPaymentDates" type="AdjustableRelativeOrPeriodicDates2" />
  </xsd:sequence>
  <xsd:attribute name="id" type="xsd:ID" use="required"/>
</xsd:complexType>
  
```

**Complex Type: InterestLegCalculationPeriodDatesReference**

Super-types:

[Reference](#) < InterestLegCalculationPeriodDatesReference (by extension)

Sub-types:

None

**Name**

InterestLegCalculationPeriodDatesReference

**Used by (from the same schema document)**Complex Type [InterestLegResetDates](#)**Abstract**

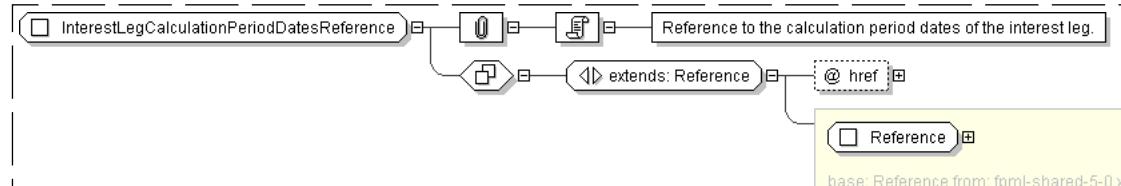
no

**Documentation**

Reference to the calculation period dates of the interest leg.

**XML Instance Representation**

```
<...>
  href=" xsd:IDREF [1]" />
```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="InterestLegCalculationPeriodDatesReference">
  <xsd:complexContent>
    <xsd:extension base=" Reference ">
      <xsd:attribute name="href" type=" xsd:IDREF "
        use="required" reference="InterestLegCalculationPeriodDates" />
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
  
```

[top](#)**Complex Type: InterestLegResetDates**

Super-types:

None

Sub-types:

None

**Name**

InterestLegResetDates

**Used by (from the same schema document)**Complex Type [InterestLegCalculationPeriodDates](#)**Abstract**

no

**XML Instance Representation**

```

<...>
  <calculationPeriodDatesReference> InterestLegCalculationPeriodDatesReference
  </calculationPeriodDatesReference> [1]
  'A pointer style reference to the associated calculation period dates component
  defined elsewhere in the document..'
  
```

Start **Choice** [1]

```

  <resetRelativeTo> ResetRelativeTypeEnum </resetRelativeTo> [1]
  '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.'
  
```

```

  <resetFrequency> ResetFrequency </resetFrequency> [1]
  '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.'

End Choice

<initialFixingDate> RelativeDateOffset </initialFixingDate> [0..1]

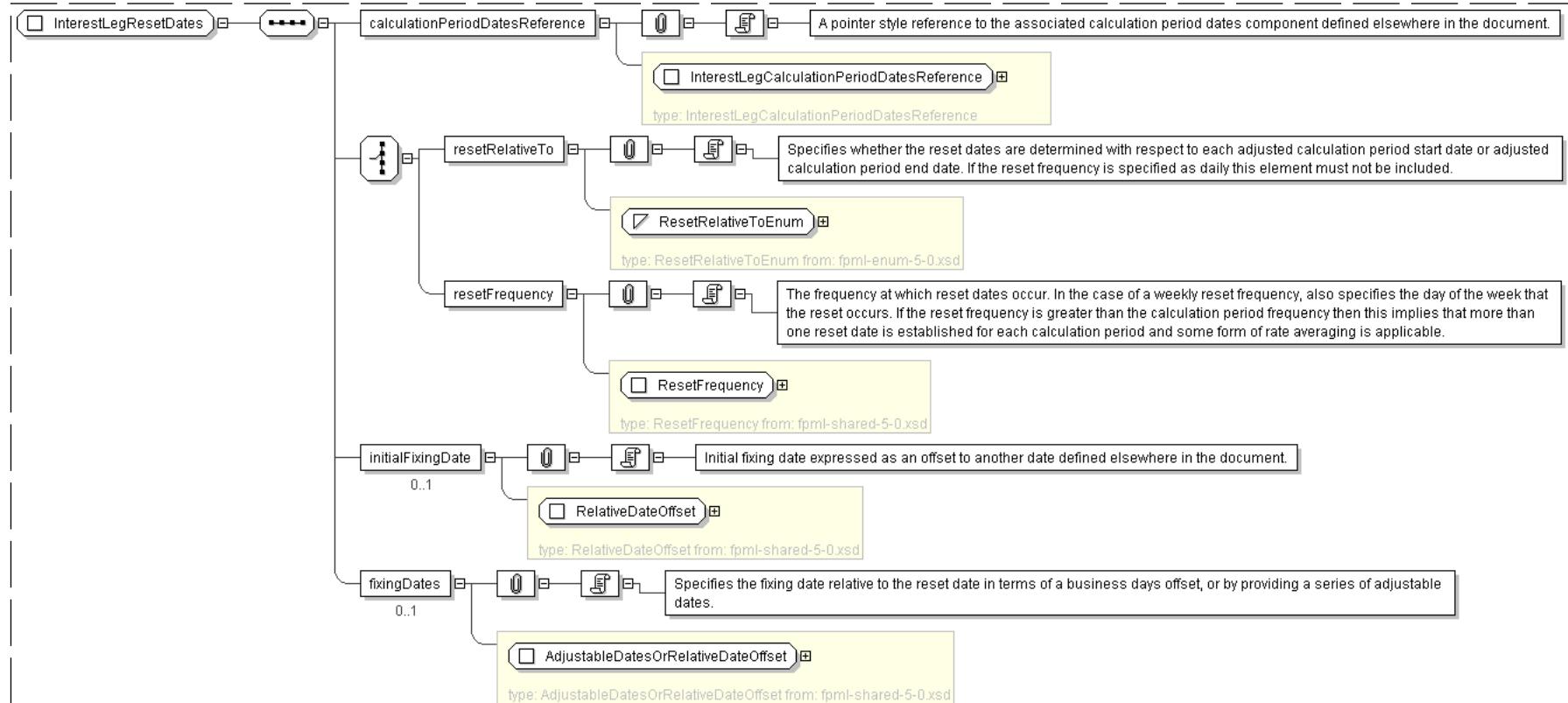
'Initial fixing date expressed as an offset to another date defined elsewhere in the document.'

<fixingDates> AdjustableDatesOrRelativeDateOffset </fixingDates> [0..1]

'Specifies the fixing date relative to the reset date in terms of a business days offset, or by providing a series of adjustable dates.'

</...>

#### Diagram



#### Schema Component Representation

```

<xsd:complexType name="InterestLegResetDates">
  <xsd:sequence>
    <xsd;element name="calculationPeriodDatesReference"
      type="#_InterestLegCalculationPeriodDatesReference" />
    <xsd:choice>
      <xsd;element name="resetRelativeTo" type="#_ResetRelativeToEnum" />
      <xsd;element name="resetFrequency" type="#_ResetFrequency" />
    </xsd:choice>
    <xsd;element name="initialFixingDate" type="#_RelativeDateOffset" minOccurs="0"/>
    <xsd;element name="fixingDates" type="#_AdjustableDatesOrRelativeDateOffset" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>
  
```

**Complex Type: LegAmount**

<b>Super-types:</b>	None
<b>Sub-types:</b>	<ul style="list-style-type: none"> <li>• <a href="#">ReturnSwapAmount</a> (by extension)</li> </ul>
<b>Name</b>	LegAmount
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">InterestLeg</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type describing the amount that will paid or received on each of the payment dates. This type is used to define both the Equity Amount and the Interest Amount.

**XML Instance Representation**

```

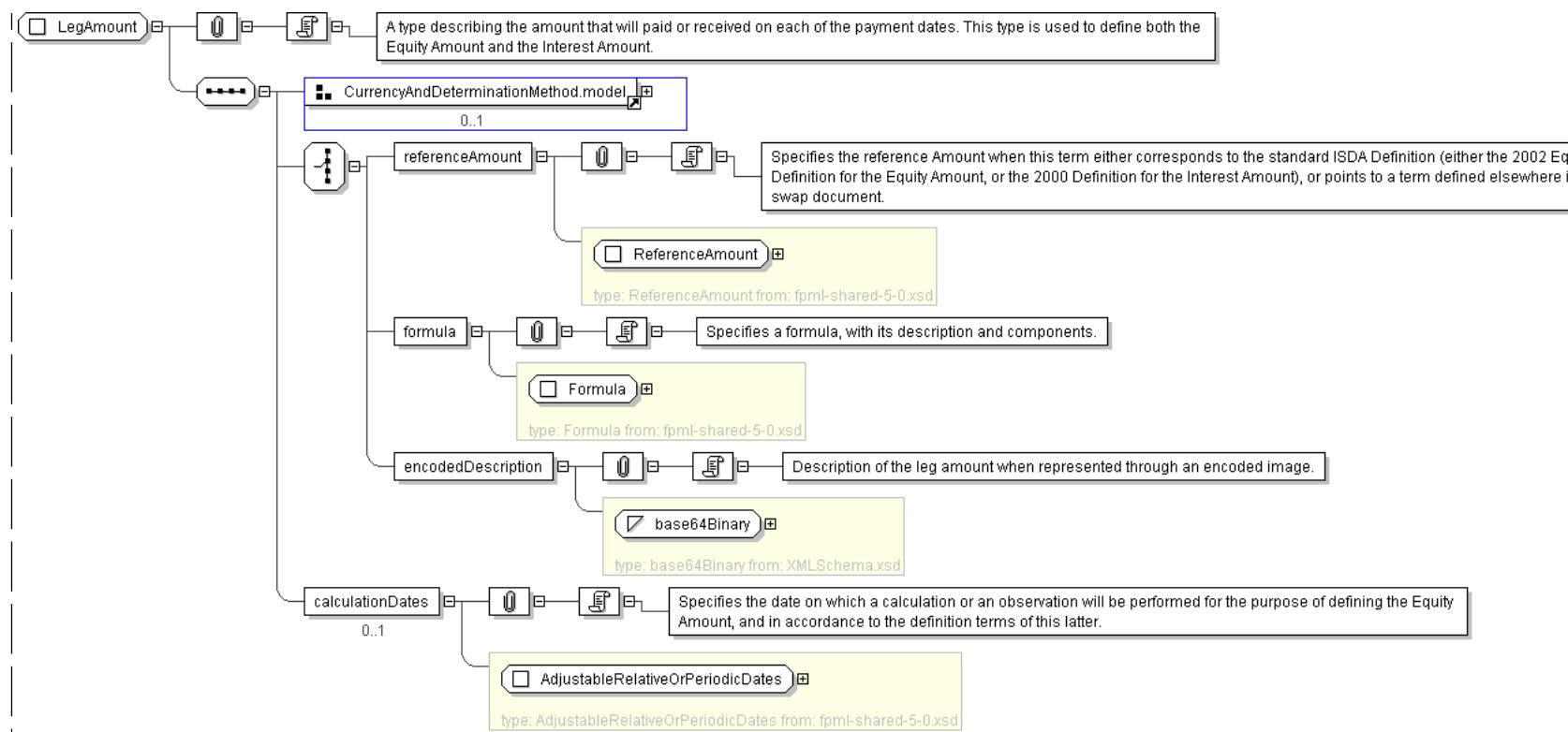
<...>
Start Group: CurrencyAndDeterminationMethod.model [0..1]
Start Choice [1]
  <currency> IdentifiedCurrency </currency> [1]
  'The currency in which an amount is denominated.'
  <determinationMethod> DeterminationMethod </determinationMethod> [1]
  'Specifies the method according to which an amount or a date is determined.'
  <currencyReference> IdentifiedCurrencyReference </currencyReference> [1]
  'Reference to a currency defined elsewhere in the document'

End Choice
End Group: CurrencyAndDeterminationMethod.model
Start Choice [1]
  <referenceAmount> ReferenceAmount </referenceAmount> [1]
  'Specifies the reference Amount when this term either corresponds to the standard
  ISDA Definition (either the 2002 Equity Definition for the Equity Amount, or the
  2000 Definition for the Interest Amount), or points to a term defined elsewhere in the
  swap document.'
  <formula> Formula </formula> [1]
  'Specifies a formula, with its description and components.'
  <encodedDescription> xsd:base64Binary </encodedDescription> [1]
  'Description of the leg amount when represented through an encoded image.'

End Choice
<calculationDates> AdjustableRelativeOrPeriodicDates </calculationDates> [0..1]
  'Specifies the date on which a calculation or an observation will be performed for the
  purpose of defining the Equity Amount, and in accordance to the definition terms of
  this latter.'
</...>

```

**Diagram**



#### Schema Component Representation

```

<xsd:complexType name="LegAmount">
  <xsd:sequence>
    <xsd:group ref=" CurrencyAndDeterminationMethod.model " minOccurs="0 "/>
    <xsd:choice>
      <xsd:element name="referenceAmount" type=" ReferenceAmount " />
      <xsd:element name="formula" type=" Formula " />
      <xsd:element name="encodedDescription" type=" xsd:base64Binary " />
    </xsd:choice>
    <xsd:element name="calculationDates" type=" AdjustableRelativeOrPeriodicDates " minOccurs="0 "/>
  </xsd:sequence>
</xsd:complexType>

```

top

#### Complex Type: LegId

Super-types:  
Sub-types:

[Token60](#) < **LegId** (by extension)  
None

Name	LegId
Used by (from the same schema document)	Complex Type <a href="#">LegIdentifier</a>
Abstract	no
Documentation	Leg identity.

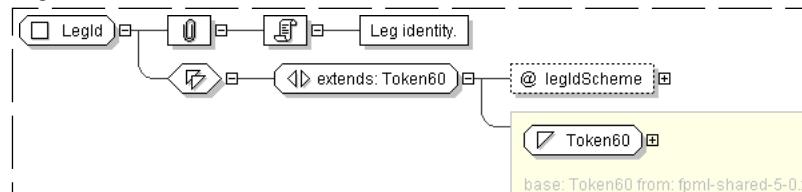
#### XML Instance Representation

```

<...>
<legIdScheme=" xsd:anyURI [1]">
  Token60

```

&lt; / &gt;

**Diagram****Schema Component Representation**

```

<xsd:complexType name="LegId">
  <xsd:simpleContent>
    <xsd:extension base=" Token60 ">
      <xsd:attribute name="legIdScheme" type=" xsd:anyURI " use="required"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
  
```

[top](#)**Complex Type: LegIdentifier**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None

<b>Name</b>	LegIdentifier
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">DirectionalLeg</a>
<b>Abstract</b>	no
<b>Documentation</b>	Version aware identification of a leg.

**XML Instance Representation**

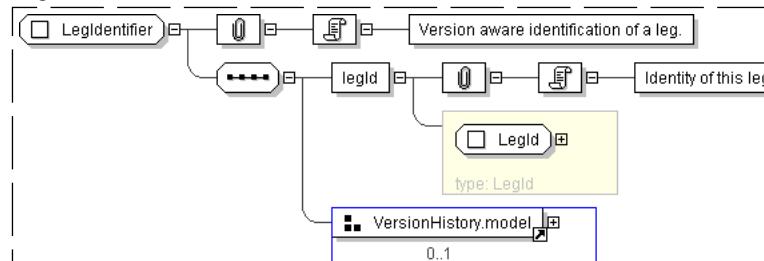
```

<...>
<legId> LegId </legId> [1]
  'Identity of this leg.'

Start Group: VersionHistory.model [0..1]
  <version> xsd:nonNegativeInteger </version> [1]
  'The version number'

  <effectiveDate> IdentifiedDate </effectiveDate> [0..1]
  'Optionally it is possible to specify a version effective date when a versionId is supplied.'

End Group: VersionHistory.model
</...>
  
```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="LegIdentifier">
  <xsd:sequence>
    <xsd:element name="legId" type=" LegId " />
    <xsd:group ref=" VersionHistory.model " minOccurs="0" />
  </xsd:sequence>
</xsd:complexType>

```

[top](#)

## Complex Type: MakeWholeProvisions

Super-types: None  
 Sub-types: None

Name	MakeWholeProvisions
Abstract	no
Documentation	A type to hold early exercise provisions.

### XML Instance Representation

```

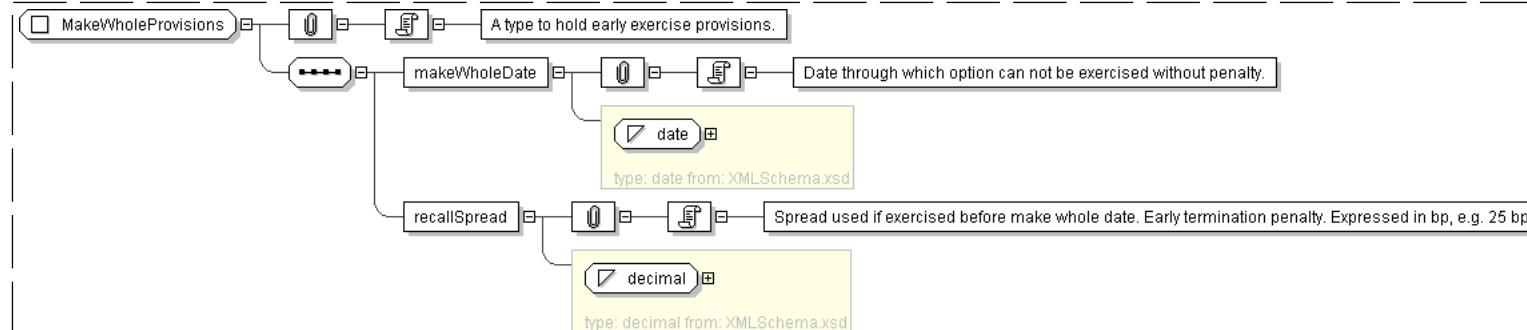
<...>
<makeWholeDate> xsd:date </makeWholeDate> [1]
  'Date through which option can not be exercised without penalty.'

<recallSpread> xsd:decimal </recallSpread> [1]
  'Spread used if exercised before make whole date. Early termination penalty. Expressed in bp,
  e.g. 25 bp.'

</...>

```

### Diagram



### Schema Component Representation

```

<xsd:complexType name="MakeWholeProvisions">
  <xsd:sequence>
    <xsd:element name="makeWholeDate" type=" xsd:date " />
    <xsd:element name="recallSpread" type=" xsd:decimal " />
  </xsd:sequence>
</xsd:complexType>

```

[top](#)

## Complex Type: NettedSwapBase

Super-types: [Product](#) < NettedSwapBase (by extension)  
 Sub-types: None

Name	NettedSwapBase
------	----------------

<b>Abstract</b>	yes
<b>Documentation</b>	An abstract base class for all swap types which have a single netted leg, such as Variance Swaps, and Correlation Swaps.

**XML Instance Representation**

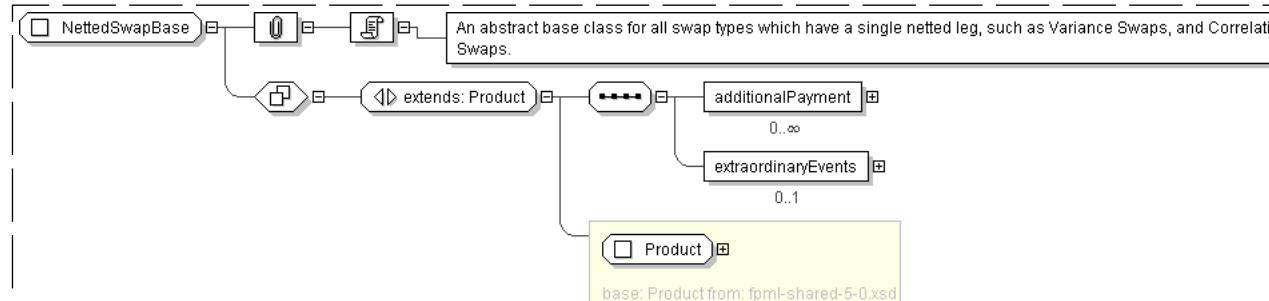
```
<...>
  id=" xsd:ID [0..1]">
    <productType> ProductType </productType> [0..*]
    'A classification of the type of product. FpML defines a simple product categorization using
    a coding scheme.'

    <productId> ProductId </productId> [0..*]
    'A product 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.'

    <additionalPayment> ClassifiedPayment </additionalPayment> [0..*]
    'Specifies additional payment(s) between the principal parties to the netted swap.'

    <extraordinaryEvents> ExtraordinaryEvents </extraordinaryEvents> [0..1]
    'Where the underlying is shares, specifies events affecting the issuer of those shares that
    may require the terms of the transaction to be adjusted.'

</...>
```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="NettedSwapBase" abstract="true">
  <xsd:complexContent>
    <xsd:extension base=" Product ">
      <xsd:sequence>
        <xsd:element name="additionalPayment" type=" ClassifiedPayment "
          minOccurs="0" maxOccurs="unbounded"/>
        <xsd:element name="extraordinaryEvents" type=" ExtraordinaryEvents " minOccurs="0"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
  
```

top

**Complex Type: OptionFeatures**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None

<b>Name</b>	OptionFeatures
<b>Used by (from the same schema document)</b>	Model Group <a href="#">Feature.model</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type for defining option features.

**XML Instance Representation**

```
<...>
<asian> Asian </asian> [0..1]
'An option where and average price is taken on valuation.'

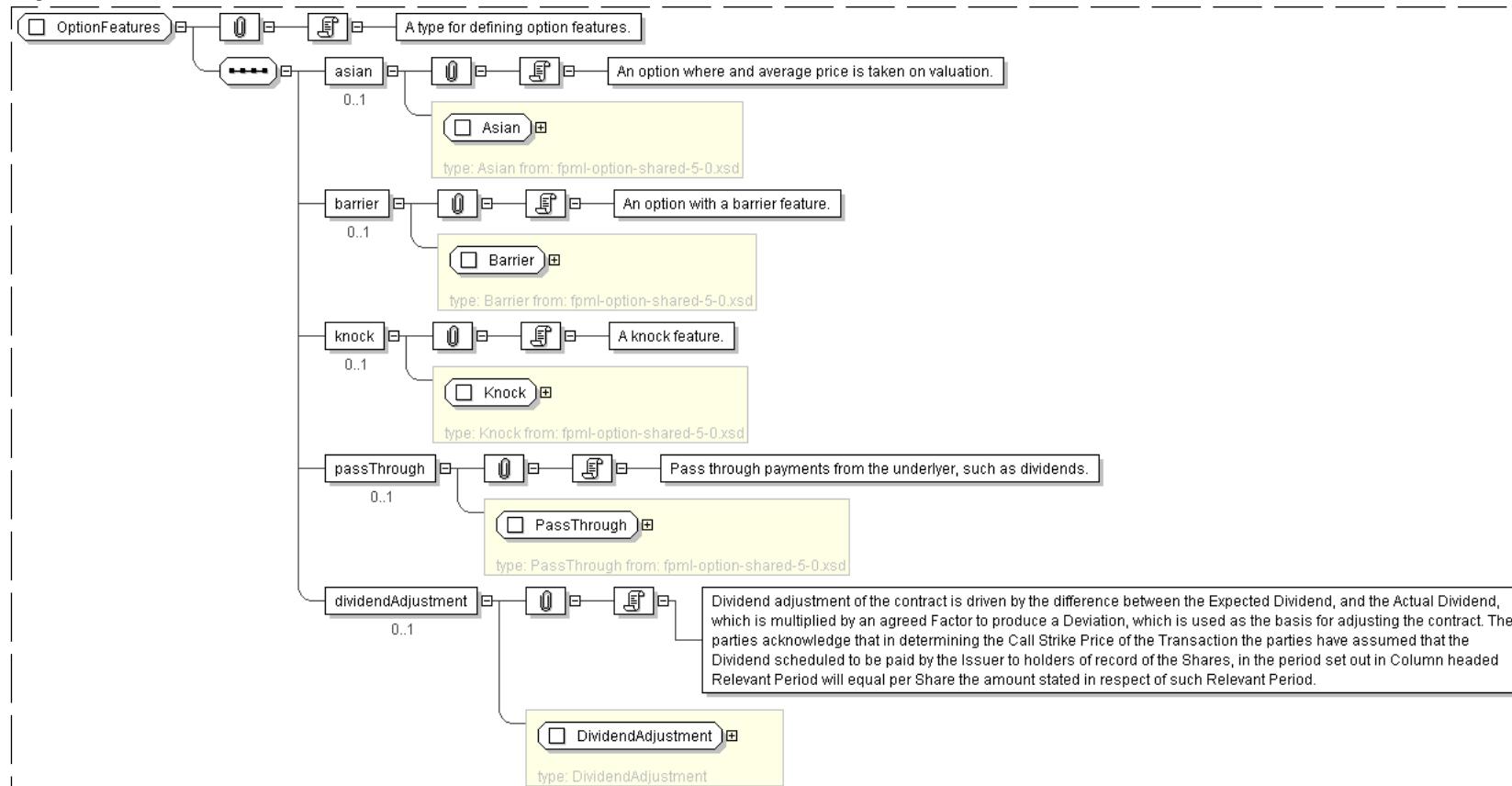
<barrier> Barrier </barrier> [0..1]
'An option with a barrier feature.'

<knock> Knock </knock> [0..1]
'A knock feature.'

<passThrough> PassThrough </passThrough> [0..1]
'Pass through payments from the underlyer, such as dividends.'

<dividendAdjustment> DividendAdjustment </dividendAdjustment> [0..1]
'Dividend adjustment of the contract is driven by the difference between the Expected Dividend, and the Actual Dividend, which is multiplied by an agreed Factor to produce a Deviation, which is used as the basis for adjusting the contract. The parties acknowledge that in determining the Call Strike Price of the Transaction the parties have assumed that the Dividend scheduled to be paid by the Issuer to holders of record of the Shares, in the period set out in Column headed Relevant Period will equal per Share the amount stated in respect of such Relevant Period.'
```

&lt;/...&gt;

**Diagram****Schema Component Representation**

```

<xsd:complexType name="OptionFeatures">
  <xsd:sequence>
    <xsd:element name="asian" type="Asian" minOccurs="0"/>
    <xsd:element name="barrier" type="Barrier" minOccurs="0"/>
    <xsd:element name="knock" type="Knock" minOccurs="0"/>
    <xsd:element name="passThrough" type="PassThrough" minOccurs="0"/>
    <xsd:element name="dividendAdjustment" type="DividendAdjustment" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>

```

top

## Complex Type: PrincipalExchangeAmount

**Super-types:** None  
**Sub-types:** None

<b>Name</b>	PrincipalExchangeAmount
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">PrincipalExchangeDescriptions</a>
<b>Abstract</b>	no
<b>Documentation</b>	Specifies the principal exchange amount, either by explicitly defining it, or by point to an amount defined somewhere else in the swap document.

### XML Instance Representation

```

<...>
Start Choice [1]
<amountRelativeTo> AmountReference </amountRelativeTo> [1]
'Reference to an amount defined elsewhere in the document.'

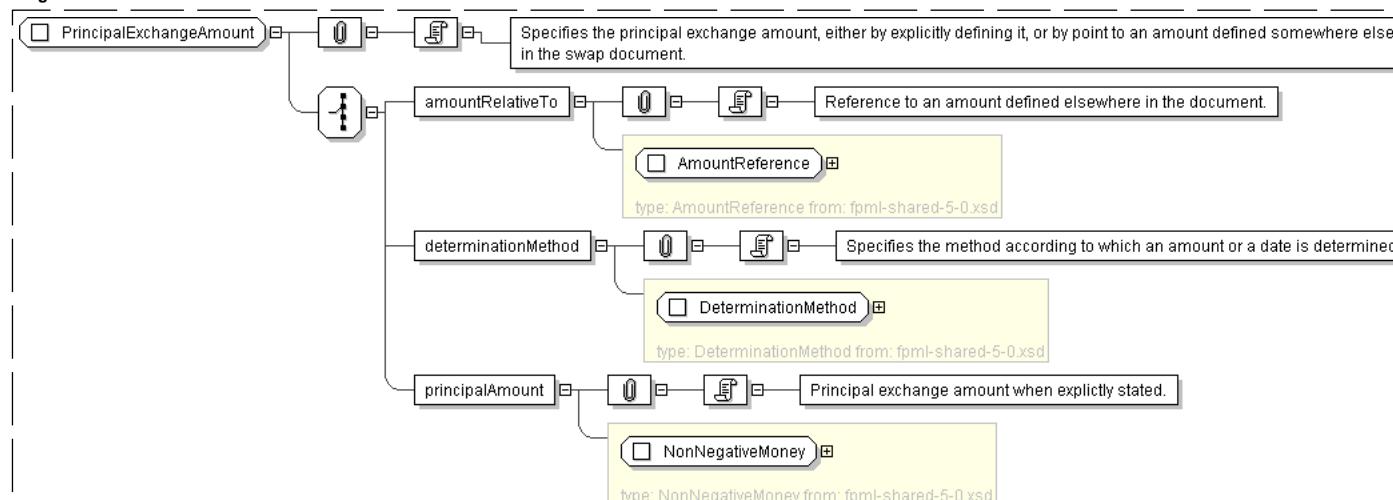
<determinationMethod> DeterminationMethod </determinationMethod> [1]
'Specifies the method according to which an amount or a date is determined.'

<principalAmount> NonNegativeMoney </principalAmount> [1]
'Principal exchange amount when explicitly stated.'

End Choice
</...>

```

### Diagram



### Schema Component Representation

```

<xsd:complexType name="PrincipalExchangeAmount">

```

```

<xsd:choice>
  <xsd:element name="amountRelativeTo" type="#AmountReference"/>
  <xsd:element name="determinationMethod" type="#DeterminationMethod"/>
  <xsd:element name="principalAmount" type="#NonNegativeMoney"/>
</xsd:choice>
</xsd:complexType>

```

top

## Complex Type: PrincipalExchangeDescriptions

Super-types: None  
Sub-types: None

Name	PrincipalExchangeDescriptions
Used by (from the same schema document)	Complex Type <a href="#">PrincipalExchangeFeatures</a>
Abstract	no
Documentation	Specifies each of the characteristics of the principal exchange cashflows, in terms of paying/receiving counterparties, amounts and dates.

### XML Instance Representation

```

<...>
<payerPartyReference> PartyReference </payerPartyReference> [1]
'A reference to the party responsible for making the payments defined by this structure.'

<payerAccountReference> AccountReference </payerAccountReference> [0..1]
'A reference to the account responsible for making the payments defined by this structure.'

<receiverPartyReference> PartyReference </receiverPartyReference> [1]
'A reference to the party that receives the payments corresponding to this structure.'

<receiverAccountReference> AccountReference </receiverAccountReference> [0..1]
'A reference to the account that receives the payments corresponding to this structure.'

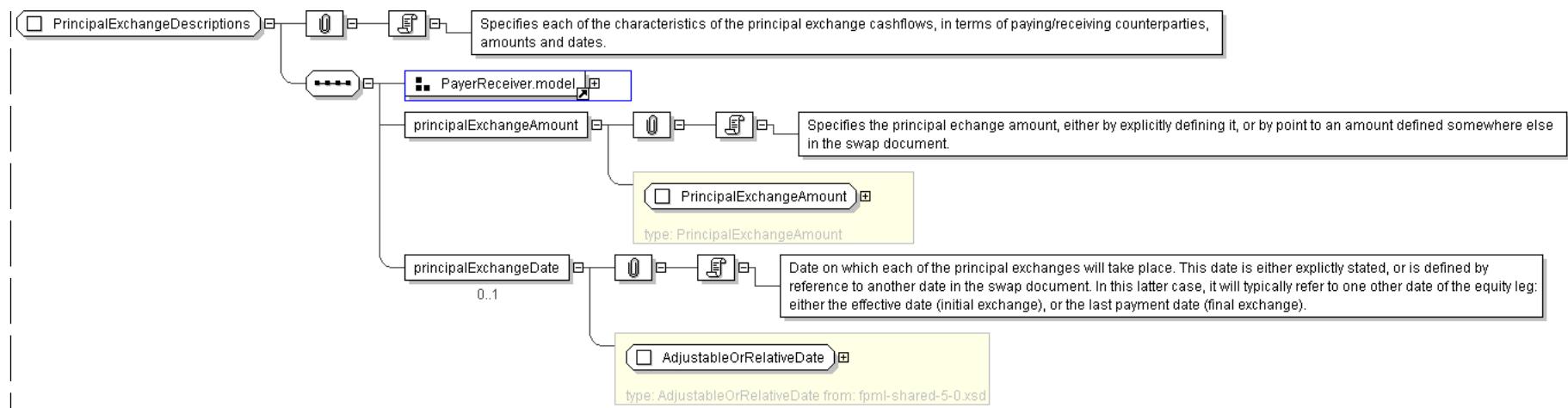
<principalExchangeAmount> PrincipalExchangeAmount </principalExchangeAmount> [1]
'Specifies the principal exchange amount, either by explicitly defining it, or by point to
an amount defined somewhere else in the swap document.'

<principalExchangeDate> AdjustableOrRelativeDate </principalExchangeDate> [0..1]
'Date on which each of the principal exchanges will take place. This date is either
explicitly stated, or is defined by reference to another date in the swap document. In
this latter case, it will typically refer to one other date of the equity leg: either
the effective date (initial exchange), or the last payment date (final exchange).'

</...>

```

### Diagram

**Schema Component Representation**

```

<xsd:complexType name="PrincipalExchangeDescriptions">
  <xsd:sequence>
    <xsd:group ref=" PayerReceiver.model ">/>
    <xsd:element name="principalExchangeAmount" type=" PrincipalExchangeAmount ">/>
    <xsd:element name="principalExchangeDate" type=" AdjustableOrRelativeDate " minOccurs="0">/>
  </xsd:sequence>
</xsd:complexType>
  
```

top

**Complex Type: PrincipalExchangeFeatures**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None

<b>Name</b>	PrincipalExchangeFeatures
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">ReturnSwapBase</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type describing the principal exchange features of the return swap.

**XML Instance Representation**

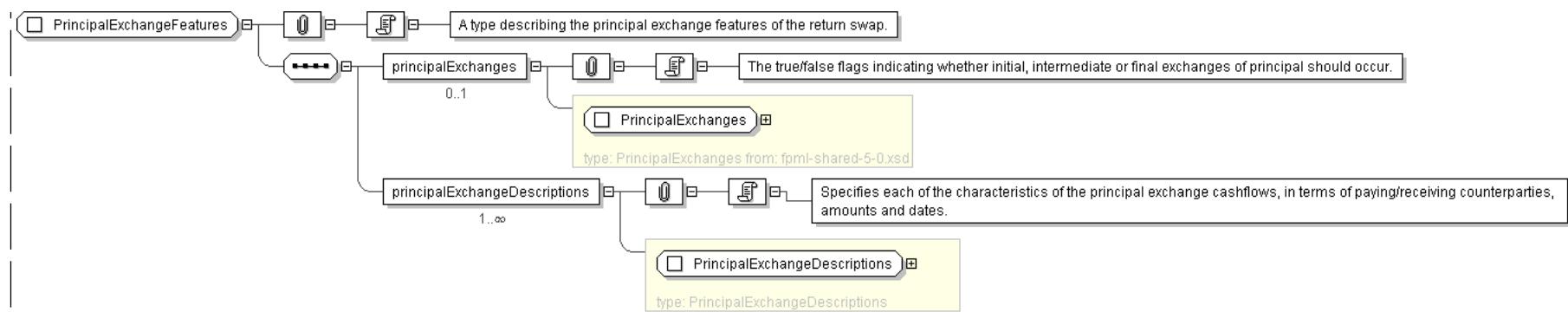
```

<...>
<principalExchanges> PrincipalExchanges </principalExchanges> [0..1]
  'The true/false flags indicating whether initial, intermediate or final exchanges of
  principal should occur.'

<principalExchangeDescriptions> PrincipalExchangeDescriptions </
  principalExchangeDescriptions> [1..*]
  'Specifies each of the characteristics of the principal exchange cashflows, in terms of
  paying/receiving counterparties, amounts and dates.'

</...>
  
```

**Diagram**

**Schema Component Representation**

```

<xsd:complexType name="PrincipalExchangeFeatures">
  <xsd:sequence>
    <xsd;element name="principalExchanges" type="PrincipalExchanges" minOccurs="0"/>
    <xsd;element name="principalExchangeDescriptions" type="PrincipalExchangeDescriptions"
      " maxOccurs="unbounded" />
  </xsd:sequence>
</xsd:complexType>

```

[top](#)**Complex Type: Representations**

Super-types:	None
Sub-types:	None

Name	Representations
Used by (from the same schema document)	Complex Type <a href="#">ExtraordinaryEvents</a>
Abstract	no
Documentation	A type for defining ISDA 2002 Equity Derivative Representations.

**XML Instance Representation**

```

<...>
<nonReliance> xsd:boolean </nonReliance> [1]
  'If true, then non reliance is applicable.'

<agreementsRegardingHedging> xsd:boolean </agreementsRegardingHedging> [1]
  'If true, then agreements regarding hedging are applicable.'

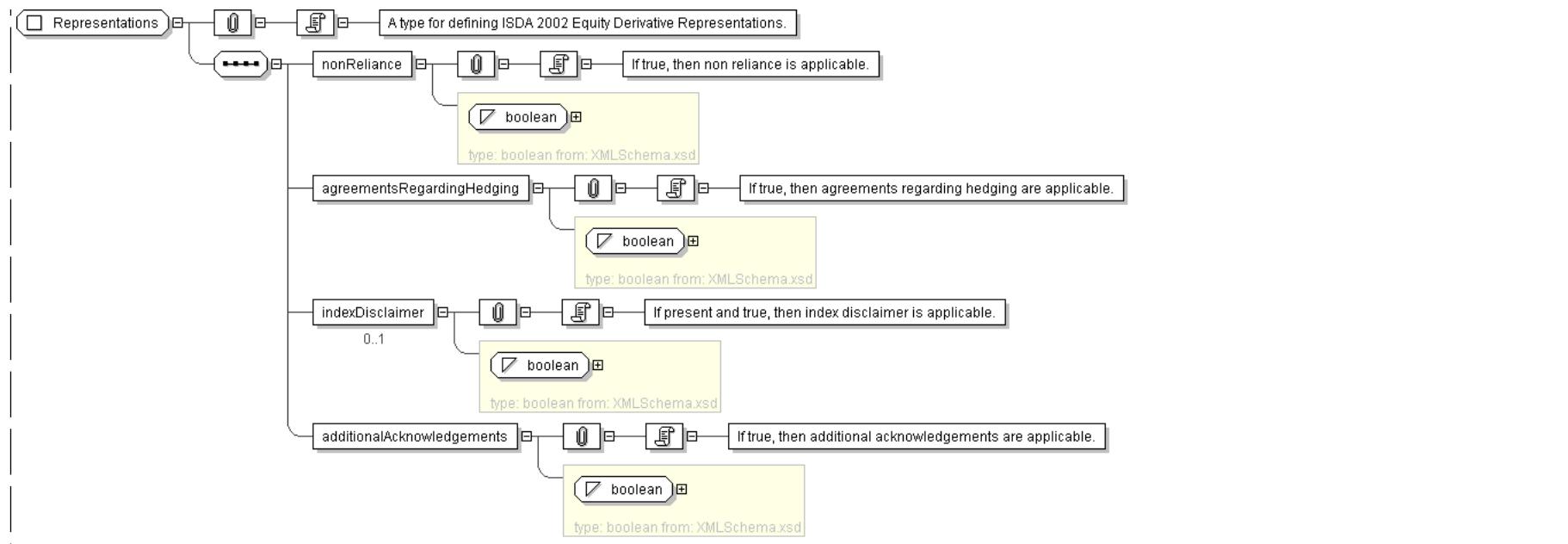
<indexDisclaimer> xsd:boolean </indexDisclaimer> [0..1]
  'If present and true, then index disclaimer is applicable.'

<additionalAcknowledgements> xsd:boolean </additionalAcknowledgements> [1]
  'If true, then additional acknowledgements are applicable.'

</...>

```

**Diagram**

**Schema Component Representation**

```

<xsd:complexType name="Representations">
  <xsd:sequence>
    <xsd;element name="nonReliance" type="xsd:boolean" />
    <xsd;element name="agreementsRegardingHedging" type="xsd:boolean" />
    <xsd;element name="indexDisclaimer" type="xsd:boolean" minOccurs="0"/>
    <xsd;element name="additionalAcknowledgements" type="xsd:boolean" />
  </xsd:sequence>
</xsd:complexType>

```

top

**Complex Type: Return**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None
<b>Name</b>	Return
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">ReturnLeg</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type describing the dividend return conditions applicable to the swap.

**XML Instance Representation**

```

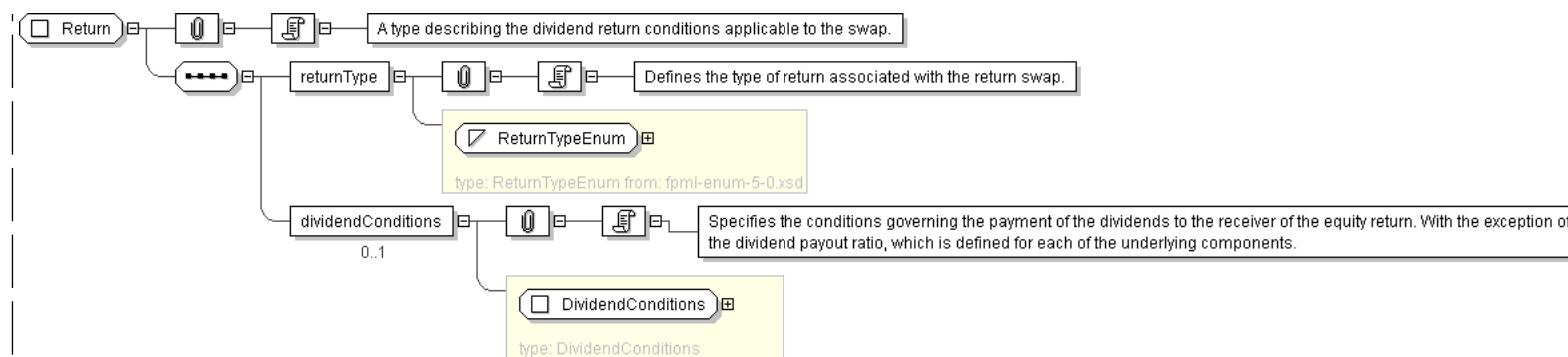
<...>
<returnType> ReturnTypeEnum </returnType> [1]
'Defines the type of return associated with the return swap.'

<dividendConditions> DividendConditions </dividendConditions> [0..1]
'Specifies the conditions governing the payment of the dividends to the receiver of the
equity return. With the exception of the dividend payout ratio, which is defined for each
of the underlying components.'

</...>

```

**Diagram**

**Schema Component Representation**

```

<xsd:complexType name="Return">
  <xsd:sequence>
    <xsd:element name="returnType" type=" ReturnTypeEnum " />
    <xsd:element name="dividendConditions" type=" DividendConditions " minOccurs="0" />
  </xsd:sequence>
</xsd:complexType>

```

[top](#)**Complex Type: `ReturnLeg`**

<b>Super-types:</b>	<a href="#">Leg</a> < <a href="#">DirectionalLeg</a> (by extension) < <a href="#">ReturnSwapLegUnderlier</a> (by extension) < <b>ReturnLeg</b> (by extension)
<b>Sub-types:</b>	None

<b>Name</b>	ReturnLeg
<b>Used by (from the same schema document)</b>	Element <a href="#">returnLeg</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type describing the return leg of a return type swap.

**XML Instance Representation**

```

<...
  id=" xsd:ID [0..1]">
  <legIdentifier> LegIdentifier </legIdentifier> [0..*]
  'Version aware identification of this leg.'

  <payerPartyReference> PartyReference </payerPartyReference> [1]
  'A reference to the party responsible for making the payments defined by this structure.'

  <payerAccountReference> AccountReference </payerAccountReference> [0..1]
  'A reference to the account responsible for making the payments defined by this structure.'

  <receiverPartyReference> PartyReference </receiverPartyReference> [1]
  'A reference to the party that receives the payments corresponding to this structure.'

  <receiverAccountReference> AccountReference </receiverAccountReference> [0..1]
  'A reference to the account that receives the payments corresponding to this structure.'

  <effectiveDate> AdjustableOrRelativeDate </effectiveDate> [0..1]
  'Specifies the effective date of this leg of the swap. When defined in relation to a date specified somewhere else in the document (through the relativeDate component), this element will typically point to the effective date of the other leg of the swap.'

  <terminationDate> AdjustableOrRelativeDate </terminationDate> [0..1]
  'Specifies the termination date of this leg of the swap. When defined in relation to a date specified somewhere else in the document (through the relativeDate component), this element will typically point to the termination date of the other leg of the swap.'

```

```

<strikeDate> AdjustableOrRelativeDate </strikeDate> [0..1]
'Specifies the strike date of this leg of the swap, used for forward starting swaps.
When defined in relation to a date specified somewhere else in the document (through
the relativeDate component), this element will typically be relative to the trade date of
the swap.'

<underlyer> Underlyer </underlyer> [1]
'Specifies the underlying component of the leg, which can be either one or many and consists
in either equity, index or convertible bond component, or a combination of these.'

<rateOfReturn> ReturnLegValuation </rateOfReturn> [1]
'Specifies the terms of the initial price of the return type swap and of the
subsequent valuations of the underlyer.'

<notional> ReturnSwapNotional </notional> [1]
'Specifies the notional of a return type swap. When used in the equity leg, the definition
will typically combine the actual amount (using the notional component defined by the
FxML industry group) and the determination method. When used in the interest leg,
the definition will typically point to the definition of the equity leg.'

<amount> ReturnSwapAmount </amount> [1]
'Specifies, in relation to each Payment Date, the amount to which the Payment Date relates.
For return swaps this element is equivalent to the Equity Amount term as defined in the
ISDA 2002 Equity Derivatives Definitions.'

<return> Return </return> [1]
'Specifies the conditions under which dividend affecting the underlyer will be paid to
the receiver of the amounts.'

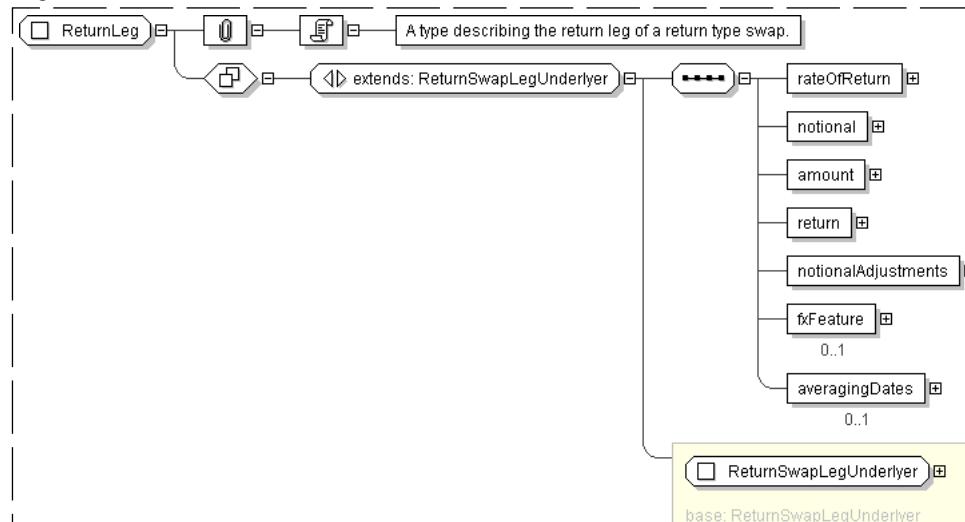
<notionalAdjustments> NotionalAdjustmentEnum </notionalAdjustments> [1]
'Specifies the conditions that govern the adjustment to the number of units of the return swap.'

<fxFeature> FxFeature </fxFeature> [0..1]
'A quanto or composite FX feature.'

<averagingDates> AveragingPeriod </averagingDates> [0..1]
'Averaging Dates used in the swap.'
```

</...>

## Diagram



**Schema Component Representation**

```

<xsd:complexType name="ReturnLeg">
  <xsd:complexContent>
    <xsd:extension base=" ReturnSwapLegUnderlyer ">
      <xsd:sequence>
        <xsd:element name="rateOfReturn" type=" ReturnLegValuation "/>
        <xsd:element name="notional" type=" ReturnSwapNotional "/>
        <xsd:element name="amount" type=" ReturnSwapAmount "/>
        <xsd:element name="return" type=" Return "/>
        <xsd:element name="notionalAdjustments" type=" NotionalAdjustmentEnum "/>
        <xsd:element name="fxFeature" type=" FxFeature " minOccurs="0"/>
        <xsd:element name="averagingDates" type=" AveragingPeriod " minOccurs="0"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

[top](#)**Complex Type: ReturnLegValuation**

Super-types:	None
Sub-types:	None

Name	ReturnLegValuation
Used by (from the same schema document)	Complex Type <a href="#">ReturnLeg</a>
Abstract	no
Documentation	A type describing the initial and final valuation of the underlyer.

**XML Instance Representation**

```

<...>
  <initialPrice> ReturnLegValuationPrice </initialPrice> [1]
  'Specifies the initial reference price of the underlyer. This price can be expressed either
  as an actual amount/currency, as a determination method, or by reference to another
  value specified in the swap document.'

  <notionalReset> xsd:boolean </notionalReset> [0..1]
  'For return swaps, this element is equivalent to the term \'Equity Notional Reset\' as
  defined in the ISDA 2002 Equity Derivatives Definitions. The reference to the ISDA
  definition is either \'Applicable\' or \'Inapplicable\'.'

  <valuationPriceInterim> ReturnLegValuationPrice </valuationPriceInterim> [0..1]
  'Specifies the final valuation price of the underlyer. This price can be expressed either as
  an actual amount/currency, as a determination method, or by reference to another
  value specified in the swap document.'

  <valuationPriceFinal> ReturnLegValuationPrice </valuationPriceFinal> [1]
  'Specifies the final valuation price of the underlyer. This price can be expressed either as
  an actual amount/currency, as a determination method, or by reference to another
  value specified in the swap document.'

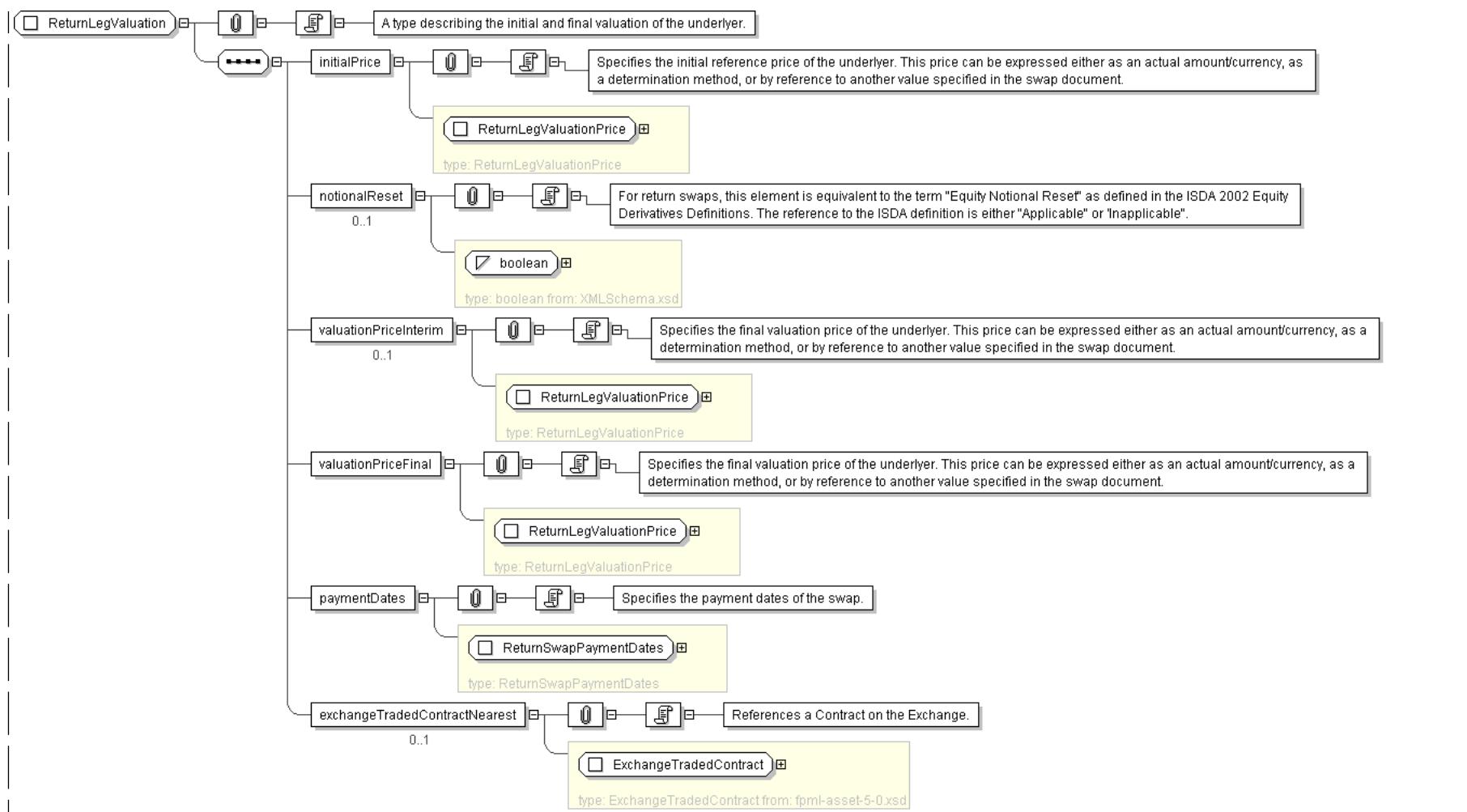
  <paymentDates> ReturnSwapPaymentDates </paymentDates> [1]
  'Specifies the payment dates of the swap.'

  <exchangeTradedContractNearest> ExchangeTradedContract </exchangeTradedContractNearest> [0..1]
  'References a Contract on the Exchange.'

</...>

```

**Diagram**



#### Schema Component Representation

```

<xsd:complexType name="ReturnLegValuation">
  <xsd:sequence>
    <xsd:element name="initialPrice" type="ReturnLegValuationPrice" />
    <xsd:element name="notionalReset" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="valuationPriceInterim" type="ReturnLegValuationPrice" minOccurs="0" />
    <xsd:element name="valuationPriceFinal" type="ReturnLegValuationPrice" />
    <xsd:element name="paymentDates" type="ReturnSwapPaymentDates" />
    <xsd:element name="exchangeTradedContractNearest" type="ExchangeTradedContract" minOccurs="0" />
  </xsd:sequence>
</xsd:complexType>

```

top

#### Complex Type: `ReturnLegValuationPrice`

Super-types:	<a href="#">Price</a> < <code>ReturnLegValuationPrice</code> (by extension)
Sub-types:	None

<b>Name</b>	ReturnLegValuationPrice
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">ReturnLegValuation</a> , Complex Type <a href="#">ReturnLegValuation</a> , Complex Type <a href="#">ReturnLegValuation</a>
<b>Abstract</b>	no

**XML Instance Representation**

```

<...>
<commission> Commission </commission> [0..1]
'This optional component specifies the commission to be charged for executing the hedge transactions.'

Start Choice [1]
  <determinationMethod> DeterminationMethod </determinationMethod> [1]
  'Specifies the method according to which an amount or a date is determined.'

Start Group: EquityPrice.model [0..1]
  <grossPrice> ActualPrice </grossPrice> [0..1]
  'Specifies the price of the underlyer, before commissions.'

  <netPrice> ActualPrice </netPrice> [1]
  'Specifies the price of the underlyer, net of commissions.'

  <accruedInterestPrice> xsd:decimal </accruedInterestPrice> [0..1]
  'Specifies the accrued interest that are part of the dirty price in the case of a fixed income security or a convertible bond. Expressed in percentage of the notional.'

  <fxConversion> FxConversion </fxConversion> [0..1]
  'Specifies the currency conversion rate that applies to an amount. This rate can either be defined elsewhere in the document (case of a quanto swap), or explicitly described through this component.'

End Group: EquityPrice.model
  <amountRelativeTo> AmountReference </amountRelativeTo> [1]
  'The href attribute value will be a pointer style reference to the element or component elsewhere in the document where the anchor amount is defined.'

  <grossPrice> ActualPrice </grossPrice> [0..1]
  'Specifies the price of the underlyer, before commissions.'

  <netPrice> ActualPrice </netPrice> [1]
  'Specifies the price of the underlyer, net of commissions.'

  <accruedInterestPrice> xsd:decimal </accruedInterestPrice> [0..1]
  'Specifies the accrued interest that are part of the dirty price in the case of a fixed income security or a convertible bond. Expressed in percentage of the notional.'

  <fxConversion> FxConversion </fxConversion> [0..1]
  'Specifies the currency conversion rate that applies to an amount. This rate can either be defined elsewhere in the document (case of a quanto swap), or explicitly described through this component.'

End Choice
  <cleanNetPrice> xsd:decimal </cleanNetPrice> [0..1]
  'The net price excluding accrued interest. The \"Dirty Price\" for bonds is put in the \"netPrice\" element, which includes accrued interest. Thus netPrice - cleanNetPrice = accruedInterest. The currency and price expression for this field are the same as those for the (dirty) netPrice.'

  <quotationCharacteristics> QuotationCharacteristics </quotationCharacteristics> [0..1]
  'Allows information about how the price was quoted to be provided.'

  <valuationRules> EquityValuation </valuationRules> [0..1]
  'Specifies valuation.'

</...>

```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="ReturnLegValuationPrice">
  <xsd:complexContent>
    <xsd:extension base=" Price ">
      <xsd:sequence>
        <xsd:element name="valuationRules" type=" EquityValuation " minOccurs="0" />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
  
```

[top](#)**Complex Type: ReturnSwap**

<b>Super-types:</b>	<a href="#">Product</a> < <a href="#">ReturnSwapBase</a> (by extension) < <a href="#">ReturnSwap</a> (by extension)
<b>Sub-types:</b>	None

<b>Name</b>	ReturnSwap
<b>Used by (from the same schema document)</b>	Element <a href="#">returnSwap</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type describing return swaps including return swaps (long form), total return swaps, and variance swaps.

**XML Instance Representation**

```

<...
id=" xsd:ID [0..1]">
<productType> ProductType </productType> [0..*]
' A classification of the type of product. FpML defines a simple product categorization using
a coding scheme.'
<productId> ProductId </productId> [0..*]
' A product 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.'
  
```

Start Group: [BuyerSeller.model](#) [0..1]

'BuyerSeller.model has been included as an optional child of `ReturnSwapBase` to support the situation where an implementor wishes to indicate who has manufactured the Swap through representing them as the Seller. It may be removed in future major revisions.'

```

<buyerPartyReference> PartyReference </buyerPartyReference> [1]
' A reference to the party that buys this instrument, i.e. pays for this instrument and
receives the rights defined by it. See 2000 ISDA definitions Article 11.1 (b). In the case
of FRAs this is the fixed rate payer.'
  
```

```

<buyerAccountReference> AccountReference </buyerAccountReference> [0..1]
' A reference to the account that buys this instrument.'
  
```

```

<sellerPartyReference> PartyReference </sellerPartyReference> [1]
' A reference to the party that sells ("writes") this instrument, i.e. that grants the
rights defined by this instrument and in return receives a payment for it. See 2000
ISDA definitions Article 11.1 (a). In the case of FRAs this is the floating rate payer.'
  
```

```

<sellerAccountReference> AccountReference </sellerAccountReference> [0..1]
  
```

'A reference to the account that sells this instrument.'

End Group: [BuyerSeller.model](#)

<returnSwapLeg> ... </returnSwapLeg> [1..\*]  
<principalExchangeFeatures> [PrincipalExchangeFeatures](#) </principalExchangeFeatures> [0..1]

'This is used to document a Fully Funded Return Swap.'

<additionalPayment> [ReturnSwapAdditionalPayment](#) </additionalPayment> [0..\*]

'Specifies additional payment(s) between the principal parties to the trade.'

<earlyTermination> [ReturnSwapEarlyTermination](#) </earlyTermination> [0..\*]

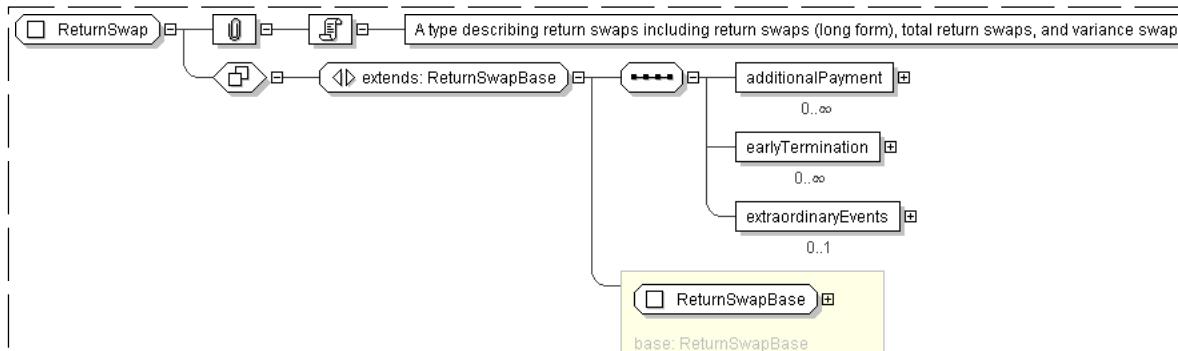
'Specifies, for one or for both the parties to the trade, the date from which it can early terminate it.'

<extraordinaryEvents> [ExtraordinaryEvents](#) </extraordinaryEvents> [0..1]

'Where the underlying is shares, specifies events affecting the issuer of those shares that may require the terms of the transaction to be adjusted.'

</...>

#### Diagram



#### Schema Component Representation

```

<xsd:complexType name="ReturnSwap">
  <xsd:complexContent>
    <xsd:extension base="" ReturnSwapBase ">
      <xsd:sequence>
        <xsd:element name="additionalPayment" type="" ReturnSwapAdditionalPayment ""
          minOccurs="0" maxOccurs="unbounded"/>
        <xsd:element name="earlyTermination" type="" ReturnSwapEarlyTermination ""
          minOccurs="0" maxOccurs="unbounded"/>
        <xsd:element name="extraordinaryEvents" type="" ExtraordinaryEvents "" minOccurs="0"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
  
```

[top](#)

#### Complex Type: [ReturnSwapAdditionalPayment](#)

Super-types:

[PaymentBase](#) < [ReturnSwapAdditionalPayment](#) (by extension)

Sub-types:

None

Name

[ReturnSwapAdditionalPayment](#)

Used by (from the same schema document)

Complex Type [ReturnSwap](#)

Abstract

no

**Documentation**

A type describing the additional payment(s) between the principal parties to the trade. This component extends some of the features of the additionalPayment component previously developed in FpML. Appropriate discussions will determine whether it would be appropriate to extend the shared component in order to meet the further requirements of equity swaps.

**XML Instance Representation**

```
<...>
  id=" xsd:ID [0..1]">
    <payerPartyReference> PartyReference </payerPartyReference> [1]
    'A reference to the party responsible for making the payments defined by this structure.'

    <payerAccountReference> AccountReference </payerAccountReference> [0..1]
    'A reference to the account responsible for making the payments defined by this structure.'

    <receiverPartyReference> PartyReference </receiverPartyReference> [1]
    'A reference to the party that receives the payments corresponding to this structure.'

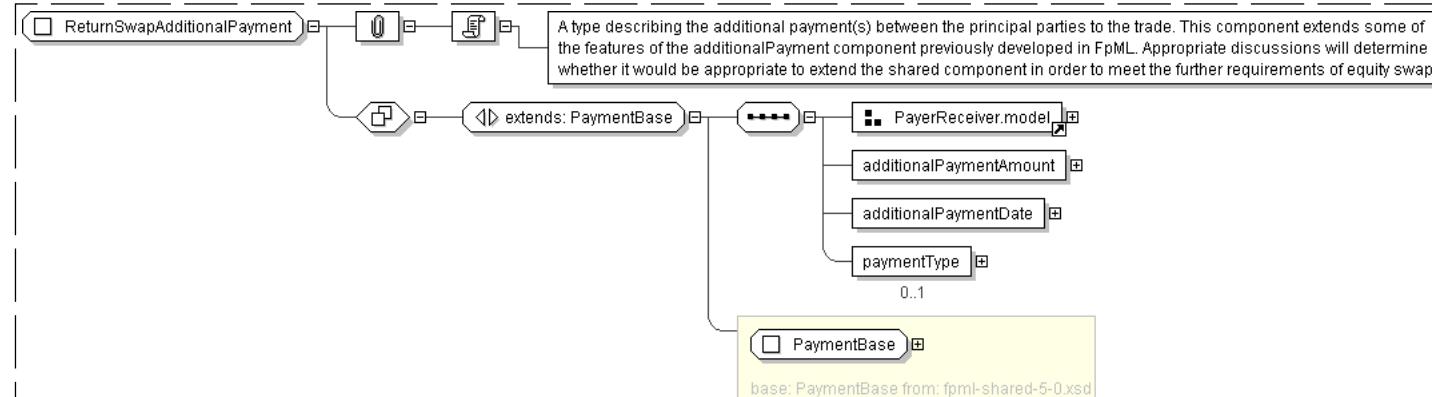
    <receiverAccountReference> AccountReference </receiverAccountReference> [0..1]
    'A reference to the account that receives the payments corresponding to this structure.'

    <additionalPaymentAmount> AdditionalPaymentAmount </additionalPaymentAmount> [1]
    'Specifies the amount of the fee along with, when applicable, the formula that supports its determination.'

    <additionalPaymentDate> AdjustableOrRelativeDate </additionalPaymentDate> [1]
    'Specifies the value date of the fee payment/receipt.'

    <paymentType> PaymentType </paymentType> [0..1]
    'Classification of the payment.'

</...>
```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="ReturnSwapAdditionalPayment">
  <xsd:complexContent>
    <xsd:extension base="PaymentBase">
      <xsd:sequence>
        <xsd:group ref="PayerReceiver.model" />
        <xsd:element name="additionalPaymentAmount" type="AdditionalPaymentAmount" />
        <xsd:element name="additionalPaymentDate" type="AdjustableOrRelativeDate" />
        <xsd:element name="paymentType" type="PaymentType" minOccurs="0"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
  
```

## Complex Type: ReturnSwapAmount

Super-types:	<a href="#">LegAmount</a> < <b>ReturnSwapAmount</b> (by extension)
Sub-types:	None
Name	ReturnSwapAmount
Used by (from the same schema document)	Complex Type <a href="#">ReturnLeg</a>
Abstract	no
Documentation	Specifies, in relation to each Payment Date, the amount to which the Payment Date relates. For Equity Swaps this element is equivalent to the Equity Amount term as defined in the ISDA 2002 Equity Derivatives Definitions.

### XML Instance Representation

```

<...>
Start Group: CurrencyAndDeterminationMethod.model [0..1]
Start Choice [1]
  <currency> IdentifiedCurrency </currency> [1]
    'The currency in which an amount is denominated.'

  <determinationMethod> DeterminationMethod </determinationMethod> [1]
    'Specifies the method according to which an amount or a date is determined.'

  <currencyReference> IdentifiedCurrencyReference </currencyReference> [1]
    'Reference to a currency defined elsewhere in the document'

End Choice
End Group: CurrencyAndDeterminationMethod.model
Start Choice [1]
  <referenceAmount> ReferenceAmount </referenceAmount> [1]
    'Specifies the reference Amount when this term either corresponds to the standard
    ISDA Definition (either the 2002 Equity Definition for the Equity Amount, or the
    2000 Definition for the Interest Amount), or points to a term defined elsewhere in the
    swap document.'

  <formula> Formula </formula> [1]
    'Specifies a formula, with its description and components.'

  <encodedDescription> xsd:base64Binary </encodedDescription> [1]
    'Description of the leg amount when represented through an encoded image.'

End Choice
<calculationDates> AdjustableRelativeOrPeriodicDates </calculationDates> [0..1]
  'Specifies the date on which a calculation or an observation will be performed for the
  purpose of defining the Equity Amount, and in accordance to the definition terms of
  this latter.'

<cashSettlement> xsd:boolean </cashSettlement> [1]
  'If true, then cash settlement is applicable.'

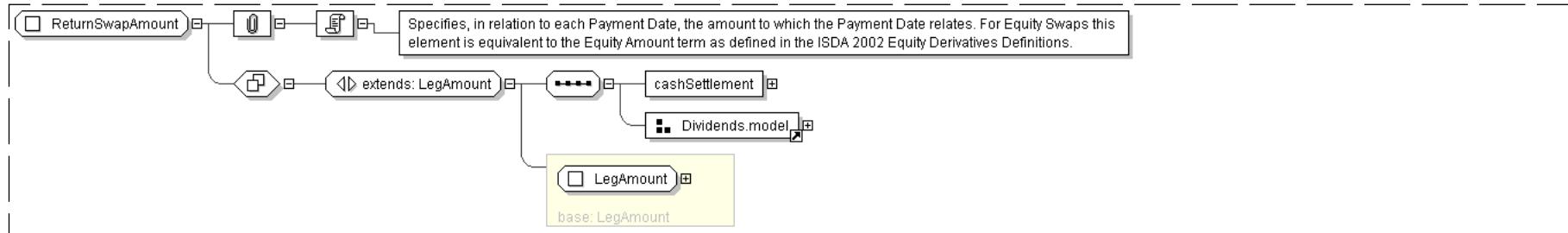
<optionsExchangeDividends> xsd:boolean </optionsExchangeDividends> [0..1]
  'If present and true, then options exchange dividends are applicable.'

<additionalDividends> xsd:boolean </additionalDividends> [0..1]
  'If present and true, then additional dividends are applicable.'

<allDividends> xsd:boolean </allDividends> [0..1]
  'Represents the European Master Confirmation value of \'All Dividends\' which, when
  applicable, signifies that, for a given Ex-Date, the daily observed Share Price for that day
  is adjusted (reduced) by the cash dividend and/or the cash value of any non cash dividend
  per Share (including Extraordinary Dividends) declared by the Issuer.'

</...>

```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="ReturnSwapAmount">
  <xsd:complexContent>
    <xsd:extension base=" LegAmount ">
      <xsd:sequence>
        <xsd:element name="cashSettlement" type=" xsd:boolean ">
        <xsd:group ref=" Dividends.model ">
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
  
```

top

**Complex Type: ReturnSwapBase**

**Super-types:** Product <ReturnSwapBase (by extension)  
**Sub-types:** • ReturnSwap (by extension)

<b>Name</b>	ReturnSwapBase
<b>Abstract</b>	yes
<b>Documentation</b>	A type describing the components that are common for return type swaps, including short and long form return swaps representations.

**XML Instance Representation**

```

<...
  id=" xsd:ID [0..1]">
  <productType> ProductType </productType> [0..*]
  'A classification of the type of product. FpML defines a simple product categorization using
  a coding scheme.'
  <productId> ProductId </productId> [0..*]
  'A product 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.'
  
```

Start Group: BuyerSeller.model [0..1]

'BuyerSeller.model has been included as an optional child of ReturnSwapBase to support the situation where an implementor wishes to indicate who has manufactured the Swap through representing them as the Seller. It may be removed in future major revisions.'

<buyerPartyReference> PartyReference </buyerPartyReference> [1]  
 'A reference to the party that buys this instrument, ie. pays for this instrument and receives the rights defined by it. See 2000 ISDA definitions Article 11.1 (b). In the case of FRAs this the fixed rate payer.'

<buyerAccountReference> AccountReference </buyerAccountReference> [0..1]

'A reference to the account that buys this instrument.'

<sellerPartyReference> PartyReference </sellerPartyReference> [1]

'A reference to the party that sells ("writes") this instrument, i.e. that grants the rights defined by this instrument and in return receives a payment for it. See 2000 ISDA definitions Article 11.1 (a). In the case of FRAs this is the floating rate payer.'

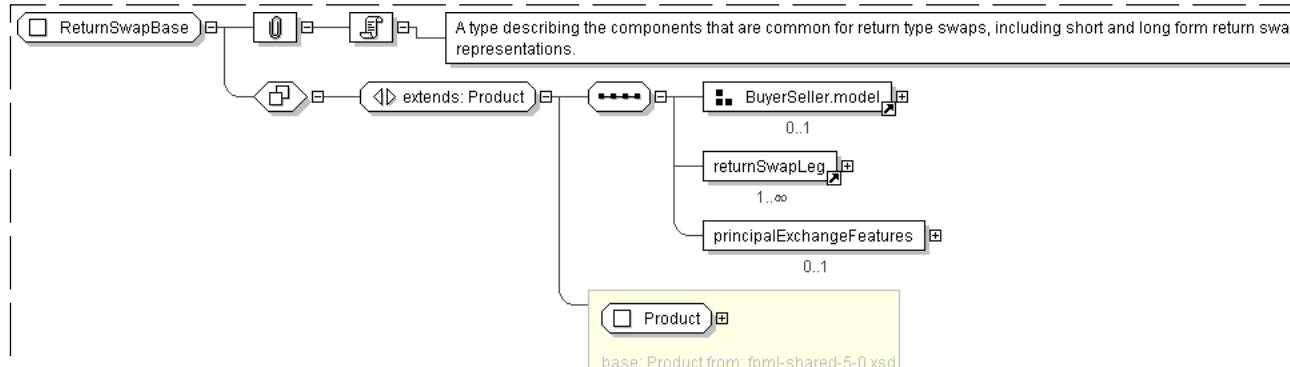
<sellerAccountReference> [AccountReference](#) </sellerAccountReference> [0..1]  
'A reference to the account that sells this instrument.'

End Group: [BuyerSeller.model](#)

<returnSwapLeg> ... </returnSwapLeg> [1..\*]  
<principalExchangeFeatures> [PrincipalExchangeFeatures](#) </principalExchangeFeatures> [0..1]  
'This is used to document a Fully Funded Return Swap.'

<...>

#### Diagram



#### Schema Component Representation

```

<xsd:complexType name="ReturnSwapBase" abstract="true">
  <xsd:complexContent>
    <xsd:extension base="#Product">
      <xsd:sequence>
        <xsd:group ref="BuyerSeller.model" minOccurs="0"/>
        <xsd:element ref="returnSwapLeg" maxOccurs="unbounded"/>
        <xsd:element name="principalExchangeFeatures" type="PrincipalExchangeFeatures" minOccurs="0"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
  
```

[top](#)

## Complex Type: [ReturnSwapEarlyTermination](#)

Super-types:	None
Sub-types:	None

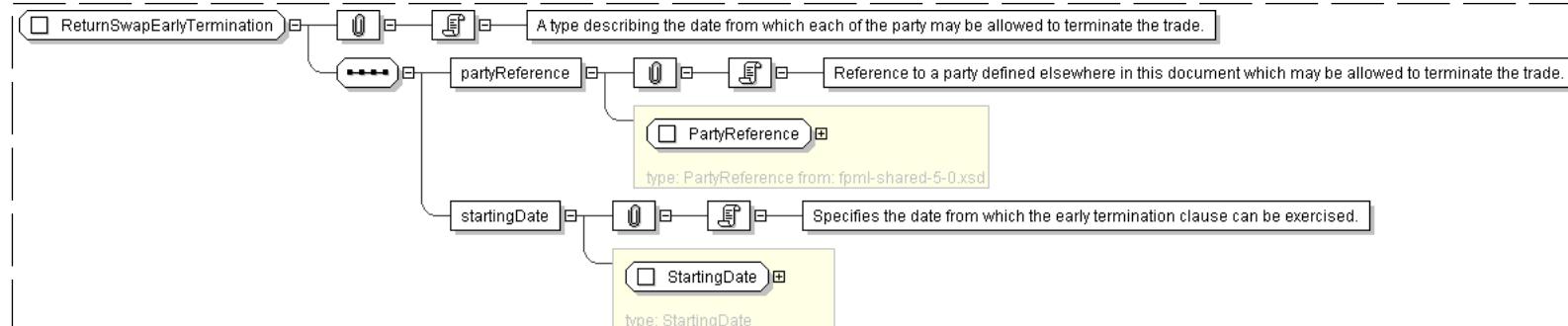
Name	ReturnSwapEarlyTermination
Used by (from the same schema document)	Complex Type <a href="#">ReturnSwap</a>
Abstract	no
Documentation	A type describing the date from which each of the party may be allowed to terminate the trade.

#### XML Instance Representation

<...>  
<partyReference> [PartyReference](#) </partyReference> [1]  
'Reference to a party defined elsewhere in this document which may be allowed to terminate the trade.'

```
<startingDate> StartingDate </startingDate> [1]
'Specifies the date from which the early termination clause can be exercised.'
```

&lt;/...&gt;

**Diagram****Schema Component Representation**

```
<xsd:complexType name="ReturnSwapEarlyTermination">
  <xsd:sequence>
    <xsd:element name="partyReference" type="PartyReference" />
    <xsd:element name="startDate" type="StartingDate" />
  </xsd:sequence>
</xsd:complexType>
```

[top](#)**Complex Type: ReturnSwapLegUnderlyer**

<b>Super-types:</b>	Leg < DirectionalLeg (by extension) < <b>ReturnSwapLegUnderlyer</b> (by extension)
<b>Sub-types:</b>	• <a href="#">ReturnLeg</a> (by extension)

<b>Name</b>	ReturnSwapLegUnderlyer
-------------	------------------------

<b>Abstract</b>	yes
-----------------	-----

<b>Documentation</b>	A base class for all return leg types with an underlyer.
----------------------	--

**XML Instance Representation**

```
<...
  id=" xsd:ID [0..1]">
  <legIdentifier> LegIdentifier </legIdentifier> [0..*]
  'Version aware identification of this leg.'

  <payerPartyReference> PartyReference </payerPartyReference> [1]
  'A reference to the party responsible for making the payments defined by this structure.'

  <payerAccountReference> AccountReference </payerAccountReference> [0..1]
  'A reference to the account responsible for making the payments defined by this structure.'

  <receiverPartyReference> PartyReference </receiverPartyReference> [1]
  'A reference to the party that receives the payments corresponding to this structure.'

  <receiverAccountReference> AccountReference </receiverAccountReference> [0..1]
  'A reference to the account that receives the payments corresponding to this structure.'

  <effectiveDate> AdjustableOrRelativeDate </effectiveDate> [0..1]
  'Specifies the effective date of this leg of the swap. When defined in relation to a date specified somewhere else in the document (through the relativeDate component), this element will typically point to the effective date of the other leg of the swap.'
```

```

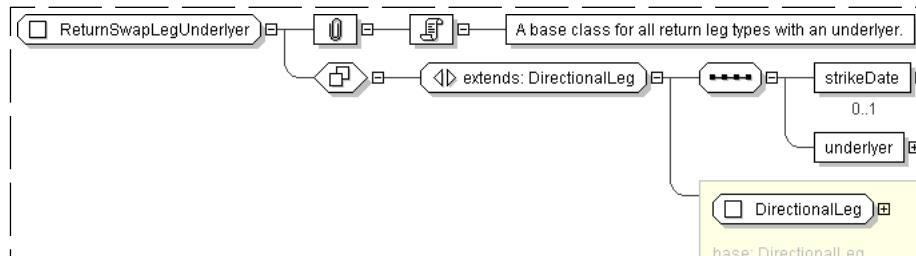
<terminationDate> AdjustableOrRelativeDate </terminationDate> [0..1]
'Specifies the termination date of this leg of the swap. When defined in relation to a
date specified somewhere else in the document (through the relativeDate component),
this element will typically point to the termination date of the other leg of the swap.'

<strikeDate> AdjustableOrRelativeDate </strikeDate> [0..1]
'Specifies the strike date of this leg of the swap, used for forward starting swaps.
When defined in relation to a date specified somewhere else in the document (through
the relativeDate component), this element will typically be relative to the trade date of
the swap.'

<underlyer> Underlyer </underlyer> [1]
'Specifies the underlying component of the leg, which can be either one or many and consists
in either equity, index or convertible bond component, or a combination of these.'

</...>

```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="ReturnSwapLegUnderlyer" abstract="true">
  <xsd:complexContent>
    <xsd:extension base="#DirectionalLeg">
      <xsd:sequence>
        <xsd:element name="strikeDate" type="AdjustableOrRelativeDate" minOccurs="0"/>
        <xsd:element name="underlyer" type="Underlyer" />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

top

**Complex Type: ReturnSwapNotional**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None
<b>Name</b>	ReturnSwapNotional
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">InterestLeg</a> , Complex Type <a href="#">ReturnLeg</a>
<b>Abstract</b>	no
<b>Documentation</b>	Specifies the notional of return type swap. When used in the equity leg, the definition will typically combine the actual amount (using the notional component defined by the FpML industry group) and the determination method. When used in the interest leg, the definition will typically point to the definition of the equity leg.

**XML Instance Representation**

```

<...
  id="xsd:ID [0..1]">
  Start Choice [1]
    <relativeNotionalAmount> ReturnSwapNotionalAmountReference </relativeNotionalAmount> [1]
    'A reference to the return swap notional amount defined elsewhere in this document.'

```

```

<relativeDeterminationMethod> DeterminationMethodReference </relativeDeterminationMethod> [1]
'A reference to the return swap notional determination method defined elsewhere in
this document.'

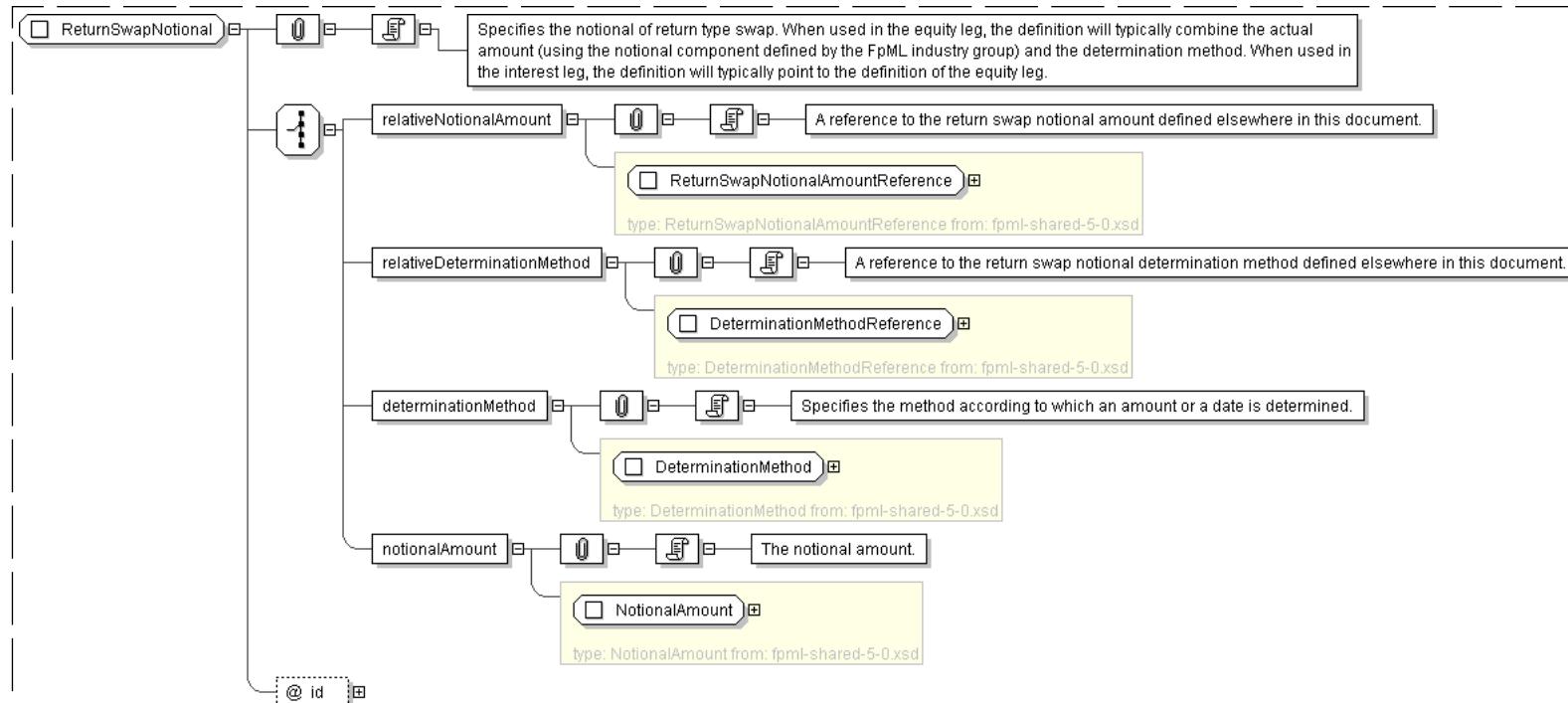
<determinationMethod> DeterminationMethod </determinationMethod> [1]
'Specifies the method according to which an amount or a date is determined.'

<notionalAmount> NotionalAmount </notionalAmount> [1]
'The notional amount.'

```

End Choice

&lt;/...&gt;

**Diagram****Schema Component Representation**

```

<xsd:complexType name="ReturnSwapNotional">
  <xsd:choice>
    <xsd:element name="relativeNotionalAmount" type="ReturnSwapNotionalAmountReference" />
    <xsd:element name="relativeDeterminationMethod" type="DeterminationMethodReference" />
    <xsd:element name="determinationMethod" type="DeterminationMethod" />
    <xsd:element name="notionalAmount" type="NotionalAmount" />
  </xsd:choice>
  <xsd:attribute name="id" type="xsd:ID" />
</xsd:complexType>

```

top

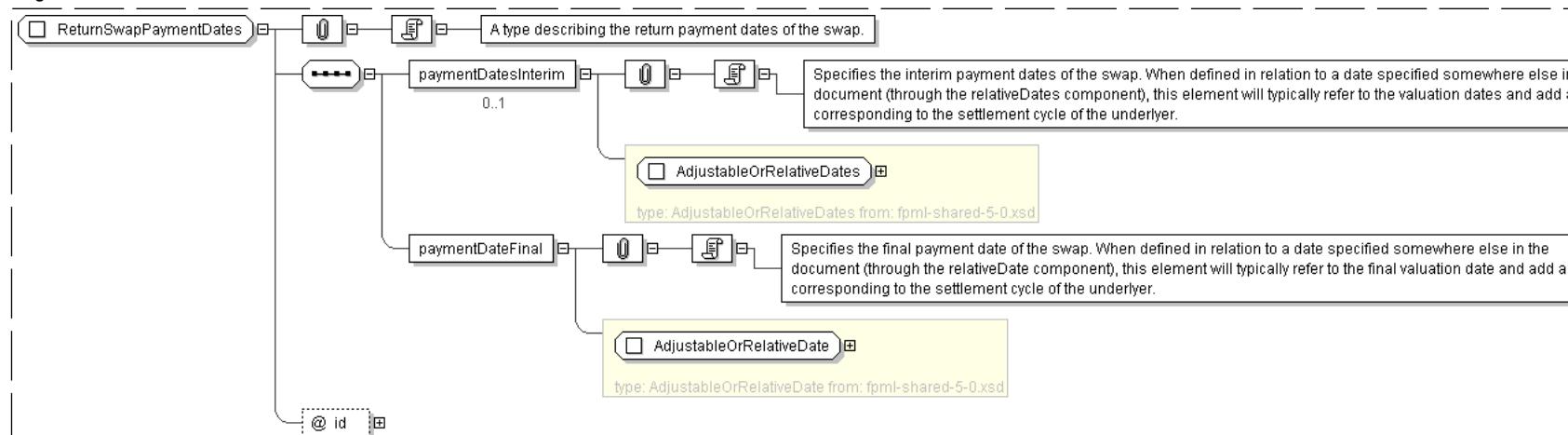
**Complex Type: ReturnSwapPaymentDates**

Super-types:	None
Sub-types:	None

<b>Name</b>	ReturnSwapPaymentDates
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">ReturnLegValuation</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type describing the return payment dates of the swap.

**XML Instance Representation**

```
<...>
  id=" xsd:ID [0..1]">
    <paymentDatesInterim> AdjustableOrRelativeDates </paymentDatesInterim> [0..1]
      'Specifies the interim payment dates of the swap. When defined in relation to a date specified somewhere else in the document (through the relativeDates component), this element will typically refer to the valuation dates and add a lag corresponding to the settlement cycle of the underlyer.'
    <paymentDateFinal> AdjustableOrRelativeDate </paymentDateFinal> [1]
      'Specifies the final payment date of the swap. When defined in relation to a date specified somewhere else in the document (through the relativeDate component), this element will typically refer to the final valuation date and add a lag corresponding to the settlement cycle of the underlyer.'
  </...>
```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="ReturnSwapPaymentDates">
  <xsd:sequence>
    <xsd:element name="paymentDatesInterim" type=" AdjustableOrRelativeDates " minOccurs="0" />
    <xsd:element name="paymentDateFinal" type=" AdjustableOrRelativeDate " />
  </xsd:sequence>
  <xsd:attribute name="id" type=" xsd:ID " />
</xsd:complexType>
  
```

top

**Complex Type: StartingDate**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None

<b>Name</b>	StartingDate
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">ReturnSwapEarlyTermination</a>
<b>Abstract</b>	no

**Documentation**

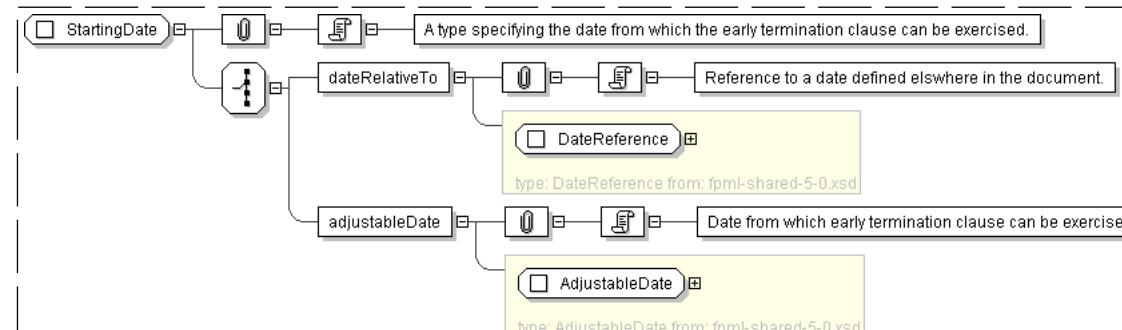
A type specifying the date from which the early termination clause can be exercised.

**XML Instance Representation**

```
<...>
Start Choice [1]
  <dateRelativeTo> DateReference </dateRelativeTo> [1]
    'Reference to a date defined elsewhere in the document.'

  <adjustableDate> AdjustableDate </adjustableDate> [1]
    'Date from which early termination clause can be exercised.'

End Choice
</...>
```

**Diagram****Schema Component Representation**

```
<xsd:complexType name="StartingDate">
  <xsd:choice>
    <xsd:element name="dateRelativeTo" type=" DateReference " />
    <xsd:element name="adjustableDate" type=" AdjustableDate " />
  </xsd:choice>
</xsd:complexType>
```

top

**Complex Type: StubCalculationPeriod**

Super-types:

None

Sub-types:

None

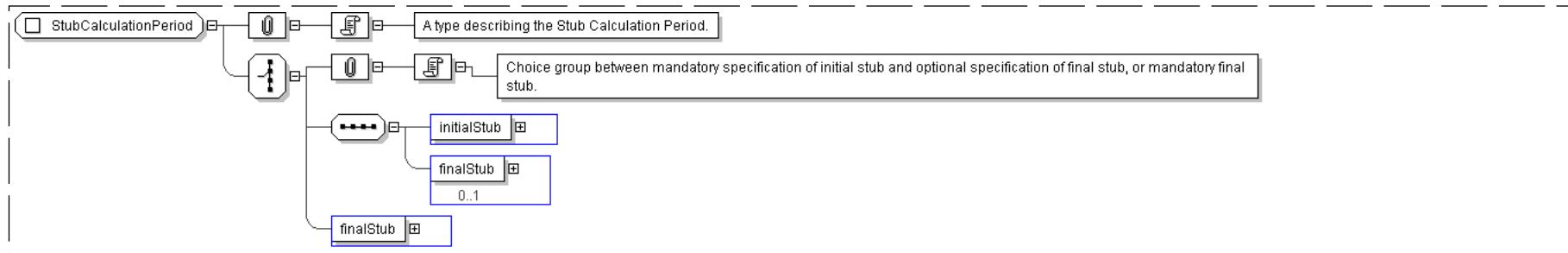
Name	StubCalculationPeriod
Used by (from the same schema document)	Complex Type <a href="#">InterestLeg</a>
Abstract	no
Documentation	A type describing the Stub Calculation Period.

**XML Instance Representation**

```
<...>
Start Choice [1]
  'Choice group between mandatory specification of initial stub and optional specification
  of final stub, or mandatory final stub.'

  <initialStub> Stub </initialStub> [1]
  <finalStub> Stub </finalStub> [0..1]
  <finalStub> Stub </finalStub> [1]

End Choice
</...>
```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="StubCalculationPeriod">
  <xsd:choice>
    <xsd:sequence>
      <xsd:element name="initialStub" type=" Stub "/>
      <xsd:element name="finalStub" type=" Stub " minOccurs="0"/>
    </xsd:sequence>
    <xsd:element name="finalStub" type=" Stub "/>
  </xsd:choice>
</xsd:complexType>
  
```

top

**Complex Type: Variance**

Super-types:	<a href="#">CalculationFromObservation</a> < <b>Variance</b> (by extension)
Sub-types:	None

Name	Variance
Abstract	no
Documentation	A type describing the variance amount of a variance swap.

**XML Instance Representation**

```

<...>
Start Choice [1]
<initialLevel> xsd:decimal </initialLevel> [1]
'Contract will strike off this initial level.'

<closingLevel> xsd:boolean </closingLevel> [1]
'If true this contract will strike off the closing level of the default exchange
traded contract.'

<expiringLevel> xsd:boolean </expiringLevel> [1]
'If true this contract will strike off the expiring level of the default exchange
traded contract.'

End Choice
<expectedN> xsd:positiveInteger </expectedN> [0..1]
'Expected number of trading days.'

<varianceAmount> NonNegativeMoney </varianceAmount> [1]
'Variance amount, which is a cash multiplier.'

Start Choice [1]
'Choice between expressing the strike as volatility or variance.'

<volatilityStrikePrice> NonNegativeDecimal </volatilityStrikePrice> [1]
<varianceStrikePrice> NonNegativeDecimal </varianceStrikePrice> [1]
  
```

```

End Choice
<varianceCap> xsd:boolean </varianceCap> [0..1]
  'If present and true, then variance cap is applicable.'

<unadjustedVarianceCap> PositiveDecimal </unadjustedVarianceCap> [0..1]
  'For use when varianceCap is applicable. Contains the scaling factor of the Variance Cap
  that can differ on a trade-by-trade basis in the European market. For example, a Variance
  Cap of 2.5^2 x Variance Strike Price has an unadjustedVarianceCap of 2.5.'

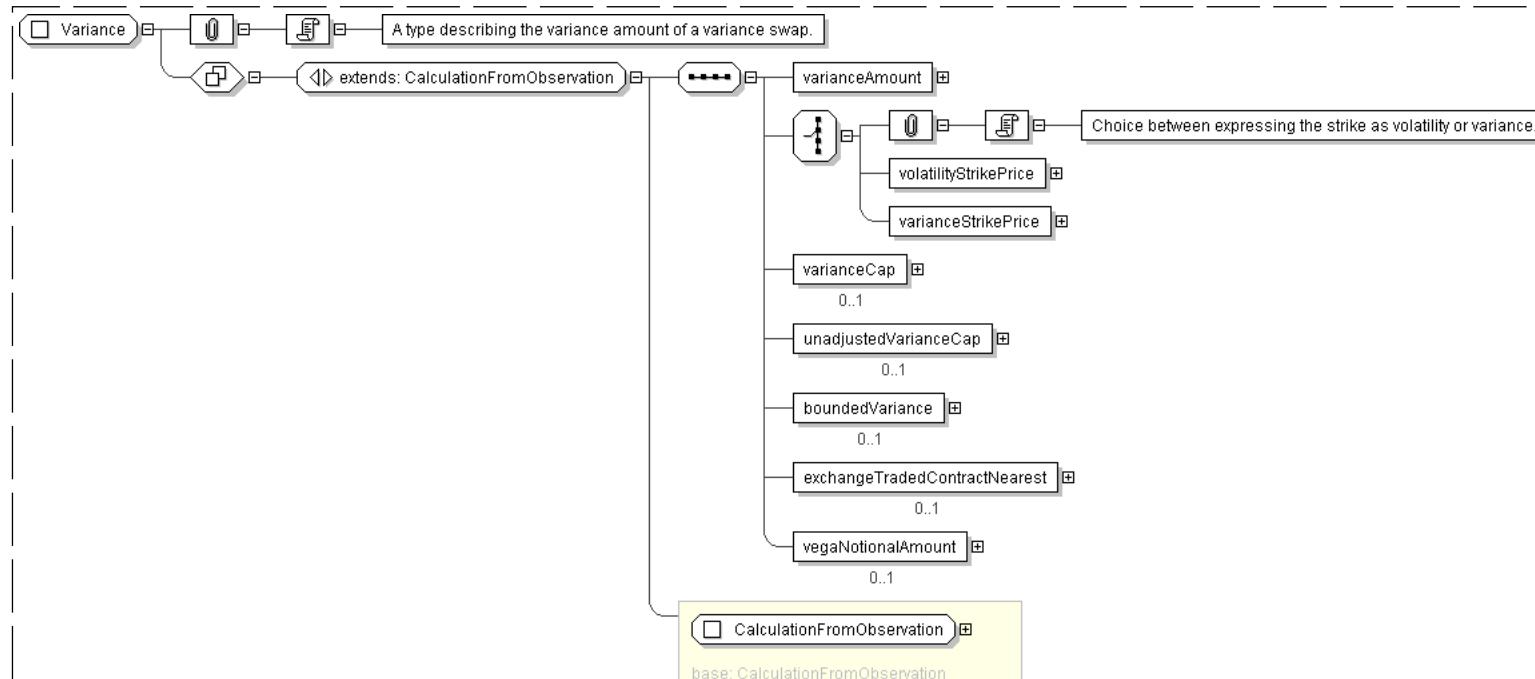
<boundedVariance> BoundedVariance </boundedVariance> [0..1]
  'Conditions which bound variance. The contract specifies one or more boundary levels.
  These levels are expressed as prices for confirmation purposes Underlyer price must be equal
  to or higher than Lower Barrier is known as Up Conditional Swap Underlyer price must be
  equal to or lower than Upper Barrier is known as Down Conditional Swap Underlyer price must
  be equal to or higher than Lower Barrier and must be equal to or lower than Upper Barrier
  is known as Barrier Conditional Swap.'

<exchangeTradedContractNearest> ExchangeTradedContract </exchangeTradedContractNearest> [0..1]
  'Specification of the exchange traded contract nearest.'

<vegaNotionalAmount> xsd:decimal </vegaNotionalAmount> [0..1]
  'Vega Notional represents the approximate gain/loss at maturity for a 1% difference
  between RVol (realised vol) and KVol (strike vol). It does not necessarily represent the
  Vega Risk of the trade.'

</...>

```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="Variance">
  <xsd:complexContent>
    <xsd:extension base=" CalculationFromObservation ">
      <xsd:sequence>
        <xsd:element name="varianceAmount" type=" NonNegativeMoney "/>
        <xsd:choice>
          <xsd:element name="volatilityStrikePrice" type=" NonNegativeDecimal "/>

```

```

<xsd:element name="varianceStrikePrice" type=" NonNegativeDecimal " />
</xsd:choice>
<xsd:element name="varianceCap" type=" xsd:boolean " minOccurs="0" />
<xsd:element name="unadjustedVarianceCap" type=" PositiveDecimal " minOccurs="0" />
<xsd:element name="boundedVariance" type=" BoundedVariance " minOccurs="0" />
<xsd:element name="exchangeTradedContractNearest" type=" ExchangeTradedContract "
minOccurs="0" />
<xsd:element name="vegaNotionalAmount" type=" xsd:decimal " minOccurs="0" />
</xsd:sequence>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>

```

[top](#)

## Model Group: CurrencyAndDeterminationMethod.model

Name	CurrencyAndDeterminationMethod.model
Used by (from the same schema document)	Complex Type <a href="#">DividendConditions</a> , Complex Type <a href="#">LegAmount</a>
Documentation	A group containing return swap amount currency definition methods

### XML Instance Representation

```

Start Choice [1]
<currency> IdentifiedCurrency </currency> [1]
'The currency in which an amount is denominated.'

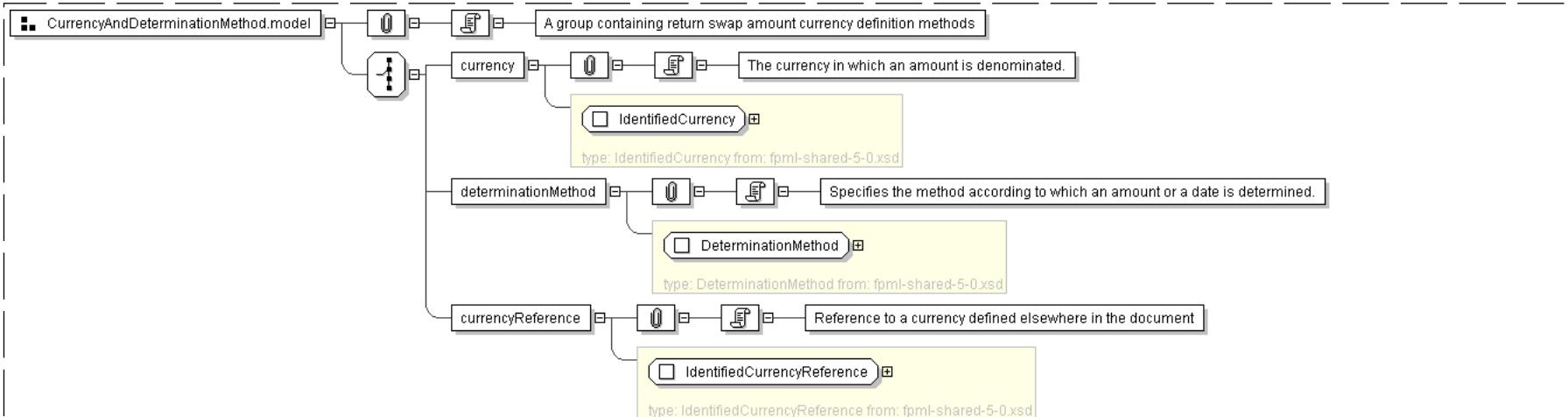
<determinationMethod> DeterminationMethod </determinationMethod> [1]
'Specifies the method according to which an amount or a date is determined.'

<currencyReference> IdentifiedCurrencyReference </currencyReference> [1]
'Reference to a currency defined elsewhere in the document'

End Choice

```

### Diagram



### Schema Component Representation

```

<xsd:group name="CurrencyAndDeterminationMethod.model">
  <xsd:choice>
    <xsd:element name="currency" type=" IdentifiedCurrency " />
    <xsd:element name="determinationMethod" type=" DeterminationMethod " />
    <xsd:element name="currencyReference" type=" IdentifiedCurrencyReference " />
  </xsd:choice>

```

&lt;/xsd:group&gt;

top

**Model Group: DeclaredCashAndCashEquivalentDividendPercentage.model****Name** DeclaredCashAndCashEquivalentDividendPercentage.model**Used by (from the same schema document)** Complex Type [DividendConditions](#)**XML Instance Representation**

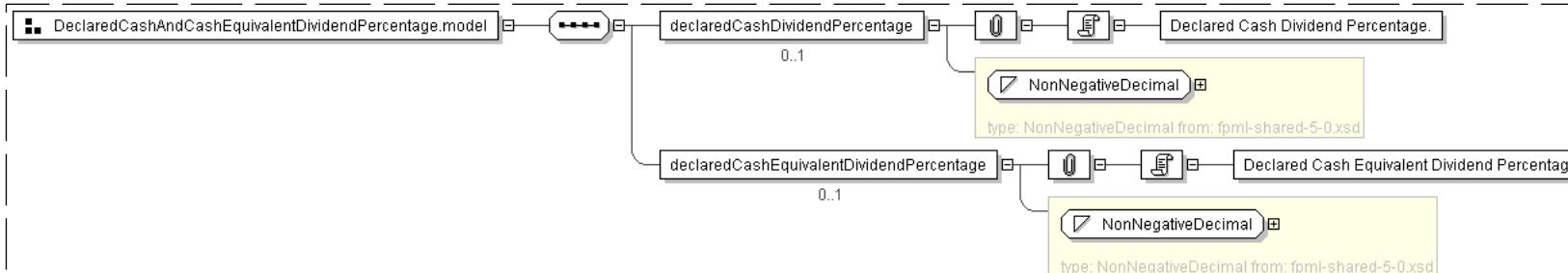
&lt;declaredCashDividendPercentage&gt; NonNegativeDecimal &lt;/declaredCashDividendPercentage&gt; [0..1]

'Declared Cash Dividend Percentage.'

&lt;declaredCashEquivalentDividendPercentage&gt; NonNegativeDecimal

&lt;/declaredCashEquivalentDividendPercentage&gt; [0..1]

'Declared Cash Equivalent Dividend Percentage.'

**Diagram****Schema Component Representation**

```

<xsd:group name="DeclaredCashAndCashEquivalentDividendPercentage.model">
  <xsd:sequence>
    <xsd:element name="declaredCashDividendPercentage" type="NonNegativeDecimal" minOccurs="0"/>
    <xsd:element name="declaredCashEquivalentDividendPercentage" type="NonNegativeDecimal"
      "minOccurs="0"/>
  </xsd:sequence>
</xsd:group>
  
```

top

**Model Group: Dividends.model****Name** Dividends.model**Used by (from the same schema document)** Complex Type [CalculatedAmount](#), Complex Type [ReturnSwapAmount](#)**Documentation** A group containing Dividend content**XML Instance Representation**

&lt;optionsExchangeDividends&gt; xsd:boolean &lt;/optionsExchangeDividends&gt; [0..1]

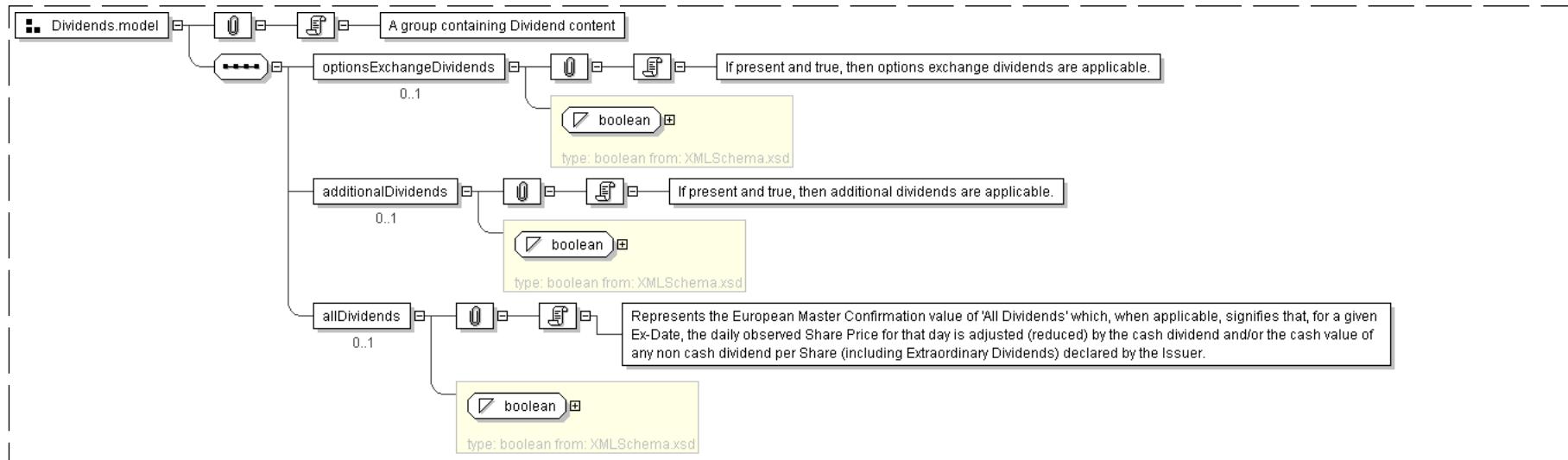
'If present and true, then options exchange dividends are applicable.'

&lt;additionalDividends&gt; xsd:boolean &lt;/additionalDividends&gt; [0..1]

'If present and true, then additional dividends are applicable.'

&lt;allDividends&gt; xsd:boolean &lt;/allDividends&gt; [0..1]

'Represents the European Master Confirmation value of \'All Dividends\' which, when applicable, signifies that, for a given Ex-Date, the daily observed Share Price for that day is adjusted (reduced) by the cash dividend and/or the cash value of any non cash dividend per Share (including Extraordinary Dividends) declared by the Issuer.'

**Diagram****Schema Component Representation**

```

<xsd:group name="Dividends.model">
  <xsd:sequence>
    <xsd:element name="optionsExchangeDividends" type="xsd:boolean" minOccurs="0"/>
    <xsd:element name="additionalDividends" type="xsd:boolean" minOccurs="0"/>
    <xsd:element name="allDividends" type="xsd:boolean" minOccurs="0"/>
  </xsd:sequence>
</xsd:group>
  
```

[top](#)**Model Group: EquityUnderlyerProvisions.model**

<b>Name</b>	EquityUnderlyerProvisions.model
<b>Documentation</b>	A group containing Equity Underlyer provisions.

**XML Instance Representation**

```

Start Group: IndexAnnexFallback.model [0..1]
Start Choice [1]
  <multipleExchangeIndexAnnexFallback> xsd:boolean </multipleExchangeIndexAnnexFallback> [1]
  'For an index option transaction, a flag to indicate whether a relevant Multiple Exchange Index Annex is applicable to the transaction. This annex defines additional provisions which are applicable where an index is comprised of component securities that are traded on multiple exchanges.'

```

```

  <componentSecurityIndexAnnexFallback> xsd:boolean </componentSecurityIndexAnnexFallback> [1]
  'For an index option transaction, a flag to indicate whether a relevant Component Security Index Annex is applicable to the transaction.'

```

```

End Choice
End Group: IndexAnnexFallback.model
<localJurisdiction> CountryCode </localJurisdiction> [0..1]

```

```

'Local Jurisdiction is a term used in the AEJ Master Confirmation, which is used to determine local taxes, which shall mean taxes, duties, and similar charges imposed by the taxing authority of the Local Jurisdiction. If this element is not present Local Jurisdiction is Not Applicable.'

```

```

<relevantJurisdiction> CountryCode </relevantJurisdiction> [0..1]

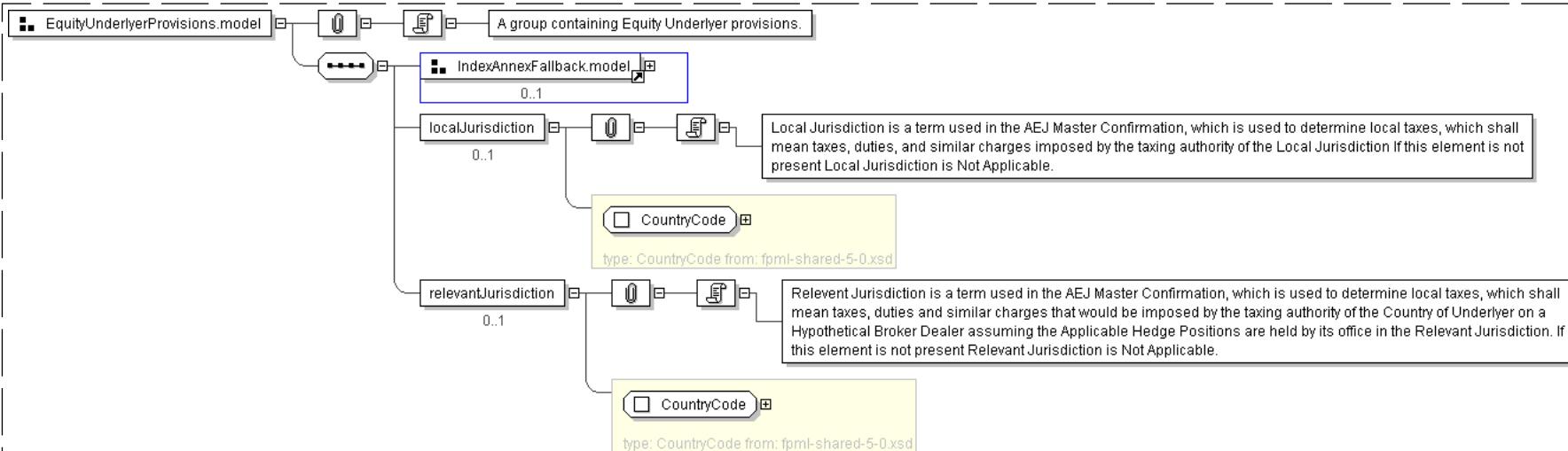
```

```

'Relevant Jurisdiction is a term used in the AEJ Master Confirmation, which is used

```

to determine local taxes, which shall mean taxes, duties and similar charges that would be imposed by the taxing authority of the Country of Underlyer on a Hypothetical Broker Dealer assuming the Applicable Hedge Positions are held by its office in the Relevant Jurisdiction. If this element is not present Relevant Jurisdiction is Not Applicable.'

**Diagram****Schema Component Representation**

```

<xsd:group name="EquityUnderlyerProvisions.model">
  <xsd:sequence>
    <xsd:group ref=" IndexAnnexFallback.model " minOccurs="0"/>
    <xsd:element name="localJurisdiction" type=" CountryCode " minOccurs="0"/>
    <xsd:element name="relevantJurisdiction" type=" CountryCode " minOccurs="0"/>
  </xsd:sequence>
</xsd:group>
  
```

top

**Model Group: Feature.model**

Name	Feature.model
Documentation	A group containing Swap and Derivative features.

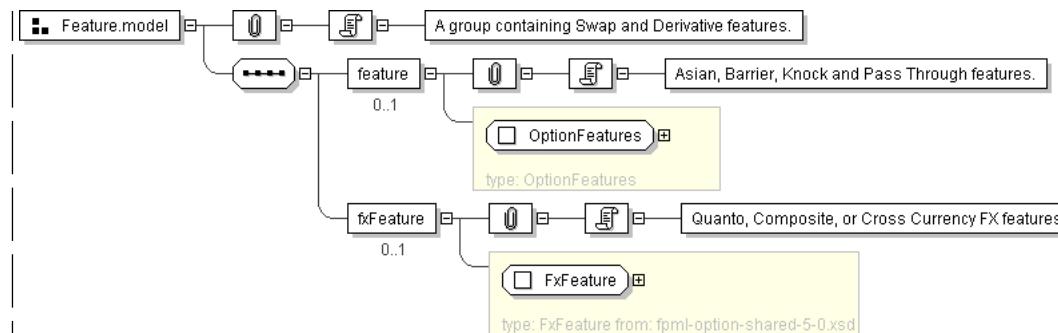
**XML Instance Representation**

```

<feature> OptionFeatures </feature> [0..1]
'Asian, Barrier, Knock and Pass Through features.'

<fxFeature> FxFeature </fxFeature> [0..1]
'Quanto, Composite, or Cross Currency FX features.'
  
```

**Diagram**

**Schema Component Representation**

```

<xsd:group name="Feature.model">
  <xsd:sequence>
    <xsd:element name="feature" type=" OptionFeatures " minOccurs="0" />
    <xsd:element name="fxFeature" type=" FxFeature " minOccurs="0" />
  </xsd:sequence>
</xsd:group>
  
```

[top](#)**Model Group: IndexAnnexFallback.model**

Name	IndexAnnexFallback.model
Used by (from the same schema document)	Model Group <a href="#">EquityUnderlyerProvisions.model</a>

**XML Instance Representation**

Start Choice [1]

```

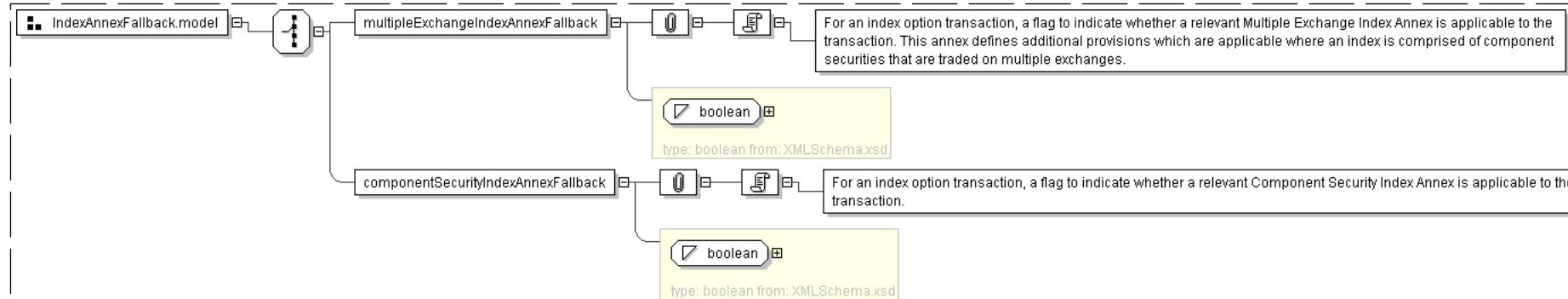
<multipleExchangeIndexAnnexFallback> xsd:boolean </multipleExchangeIndexAnnexFallback> [1]
  
```

'For an index option transaction, a flag to indicate whether a relevant Multiple Exchange Index Annex is applicable to the transaction. This annex defines additional provisions which are applicable where an index is comprised of component securities that are traded on multiple exchanges.'

<componentSecurityIndexAnnexFallback> xsd:boolean </componentSecurityIndexAnnexFallback> [1]

'For an index option transaction, a flag to indicate whether a relevant Component Security Index Annex is applicable to the transaction.'

End Choice

**Diagram****Schema Component Representation**

```

<xsd:group name="IndexAnnexFallback.model">
  <xsd:choice>
    <xsd:element name="multipleExchangeIndexAnnexFallback" type="xsd:boolean" />
    <xsd:element name="componentSecurityIndexAnnexFallback" type="xsd:boolean" />
  </xsd:choice>
</xsd:group>

```

top

## Model Group: MutualOrOptionalEarlyTermination.model

Name MutualOrOptionalEarlyTermination.model

## XML Instance Representation

```

Start Choice [1]
<mutualEarlyTermination> xsd:boolean </mutualEarlyTermination> [0..1]
  'Used for specifying whether the Mutual Early Termination Right that is detailed in the
  Master Confirmation will apply.'

<optionalEarlyTermination> xsd:boolean </optionalEarlyTermination> [1]
  'A Boolean element used for specifying whether the Optional Early Termination clause
  detailed in the agreement will apply.'

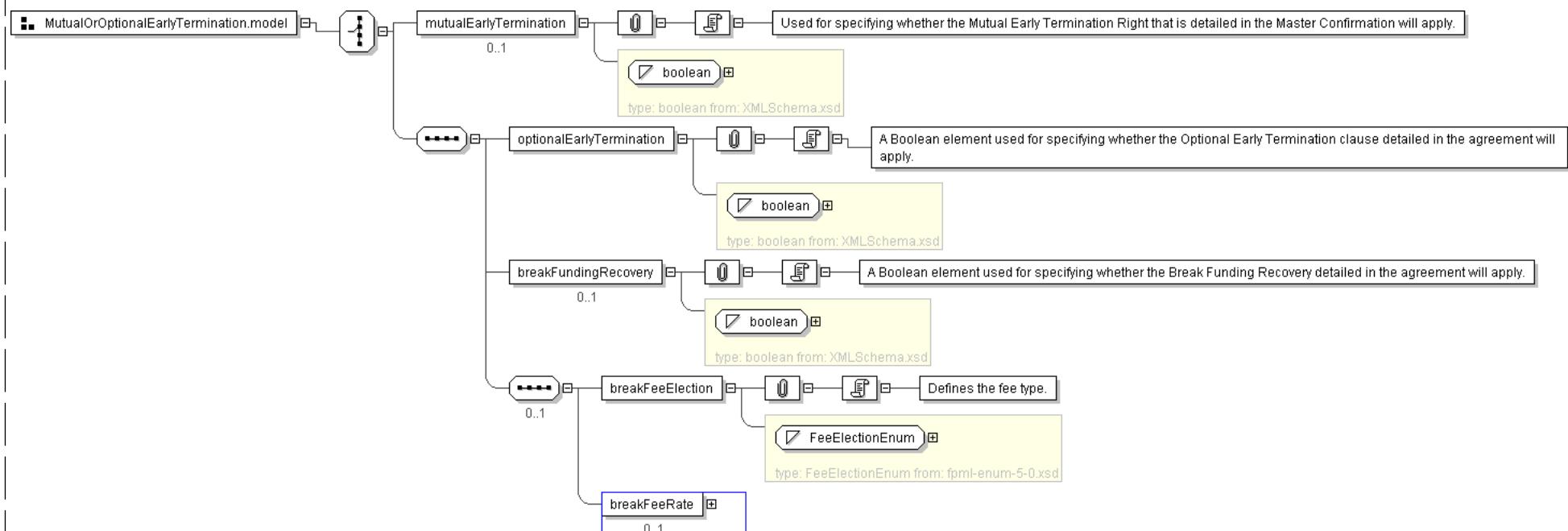
<breakFundingRecovery> xsd:boolean </breakFundingRecovery> [0..1]
  'A Boolean element used for specifying whether the Break Funding Recovery detailed in
  the agreement will apply.'

Start Sequence [0..1]
  <breakFeeElection> FeeElectionEnum </breakFeeElection> [1]
    'Defines the fee type.'

    <breakFeeRate> NonNegativeDecimal </breakFeeRate> [0..1]
End Sequence
End Choice

```

## Diagram



...  
[...]**Schema Component Representation**

```
<xsd:group name="MutualOrOptionalEarlyTermination.model">
  <xsd:choice>
    <xsd:element name="mutualEarlyTermination" type="xsd:boolean" minOccurs="0"/>
    <xsd:sequence>
      <xsd:element name="optionalEarlyTermination" type="xsd:boolean" />
      <xsd:element name="breakFundingRecovery" type="xsd:boolean" minOccurs="0"/>
      <xsd:sequence minOccurs="0">
        <xsd:element name="breakFeeElection" type="FeeElectionEnum" />
        <xsd:element name="breakFeeRate" type="NonNegativeDecimal" minOccurs="0"/>
      </xsd:sequence>
    </xsd:sequence>
  </xsd:choice>
</xsd:group>
```

top

**Legend****Complex Type:**  
Schema Component Type**AusAddress**  
Schema Component Name

**Super-types:** [Address](#) < AusAddress (by extension)  
**Sub-types:** • [QLDAddress](#) (by restriction)

If this schema component is a type definition, its type hierarchy is shown in a gray-bordered box.

Name	AusAddress
Abstract	no

The table above displays the properties of this schema component.

**XML Instance Representation**

```
<... country="Australia" >
<unitNo> string </unitNo> [0..1]
<houseNo> string </houseNo> [1]
<street> string </street> [1]
Start Choice [1]
<city> string </city> [1]
<town> string </town> [1]
End Choice
<state> AusStates </state> [1]
<postcode> string <>pattern = [1-9][0-9]{3}</> </postcode> [1]
</...>
```

The XML Instance Representation table above shows the schema component's content as an XML instance.

- The minimum and maximum occurrence of elements and attributes are provided in square brackets, e.g. [0..1].
- Model group information are shown in gray, e.g. Start Choice ... End Choice.
- For type derivations, the elements and attributes that have been added to or changed from the base type's content are shown in **bold**.
- If an element/attribute has a fixed value, the fixed value is shown in green, e.g. country="Australia".
- Otherwise, the type of the element/attribute is displayed.
  - If the element/attribute's type is in the schema, a link is provided to it.
  - For local simple type definitions, the constraints are displayed in angle brackets, e.g. <>pattern = [1-9][0-9]{3}</>.

**Schema Component Representation**

```
<complexType name="AusAddress">
  <complexContent>
    <extension base=" Address " >
      <sequence>
        <element name="state" type=" AusStates " />
        <element name="postcode">
```

```

<simpleType>
<restriction base=" string ">
<pattern value="[1-9][0-9]{3}" />
</restriction>
</simpleType>
<element>
</sequence>
<attribute name="country" type=" string " fixed="Australia"/>
</extension>
</complexContent>
</complexType>

```

The Schema Component Representation table above displays the underlying XML representation of the schema component. (Annotations are not shown.)

[top](#)

## Glossary

**Abstract** (Applies to complex type definitions and element declarations). An abstract element or complex type cannot be used to validate an element instance. If there is a reference to an abstract element, only element declarations that can substitute the abstract element can be used to validate the instance. For references to abstract type definitions, only derived types can be used.

**All Model Group** Child elements can be provided *in any order* in instances. See: <http://www.w3.org/TR/xmlschema-1/#element-all>.

**Choice Model Group** Only one from the list of child elements and model groups can be provided in instances. See: <http://www.w3.org/TR/xmlschema-1/#element-choice>.

**Collapse Whitespace Policy** Replace tab, line feed, and carriage return characters with space character (Unicode character 32). Then, collapse contiguous sequences of space characters into single space character, and remove leading and trailing space characters.

**Disallowed Substitutions** (Applies to element declarations). If *substitution* is specified, then [substitution group](#) members cannot be used in place of the given element declaration to validate element instances. If *derivation methods*, e.g. extension, restriction, are specified, then the given element declaration will not validate element instances that have types derived from the element declaration's type using the specified derivation methods. Normally, element instances can override their declaration's type by specifying an *xsi:type* attribute.

**Key Constraint** Like [Uniqueness Constraint](#), but additionally requires that the specified value(s) must be provided. See: [http://www.w3.org/TR/xmlschema-1/#cidentity-constraint\\_Definitions](http://www.w3.org/TR/xmlschema-1/#cidentity-constraint_Definitions).

**Key Reference Constraint** Ensures that the specified value(s) must match value(s) from a [Key Constraint](#) or [Uniqueness Constraint](#). See: [http://www.w3.org/TR/xmlschema-1/#cidentity-constraint\\_Definitions](http://www.w3.org/TR/xmlschema-1/#cidentity-constraint_Definitions).

**Model Group** Groups together element content, specifying the order in which the element content can occur and the number of times the group of element content may be repeated. See: [http://www.w3.org/TR/xmlschema-1/#Model\\_Groups](http://www.w3.org/TR/xmlschema-1/#Model_Groups).

**Nillable** (Applies to element declarations). If an element declaration is nillable, instances can use the *xsi:nil* attribute. The *xsi:nil* attribute is the boolean attribute, *nil*, from the <http://www.w3.org/2001/XMLSchema-instance> namespace. If an element instance has an *xsi:nil* attribute set to true, it can be left empty, even though its element declaration may have required content.

**Notation** A notation is used to identify the format of a piece of data. Values of elements and attributes that are of type, NOTATION, must come from the names of declared notations. See: [http://www.w3.org/TR/xmlschema-1/#cNotation\\_Declarations](http://www.w3.org/TR/xmlschema-1/#cNotation_Declarations).

**Preserve Whitespace Policy** Preserve whitespaces exactly as they appear in instances.

**Prohibited Derivations** (Applies to type definitions). Derivation methods that cannot be used to create sub-types from a given type definition.

**Prohibited Substitutions** (Applies to complex type definitions). Prevents sub-types that have been derived using the specified derivation methods from validating element instances in place of the given type definition.

**Replace Whitespace Policy** Replace tab, line feed, and carriage return characters with space character (Unicode character 32).

**Sequence Model Group** Child elements and model groups must be provided *in the specified order* in instances. See: <http://www.w3.org/TR/xmlschema-1/#element-sequence>.

**Substitution Group** Elements that are *members* of a substitution group can be used wherever the *head* element of the substitution group is referenced.

**Substitution Group Exclusions** (Applies to element declarations). Prohibits element declarations from nominating themselves as being able

to substitute a given element declaration, if they have types that are derived from the original element's type using the specified derivation methods.

**Target Namespace** The target namespace identifies the namespace that components in this schema belongs to. If no target namespace is provided, then the schema components do not belong to any namespace.

**Uniqueness Constraint** Ensures uniqueness of an element/attribute value, or a combination of values, within a specified scope. See: [http://www.w3.org/TR/xmlschema-1/#cidentity\\_constraint\\_Definitions](http://www.w3.org/TR/xmlschema-1/#cidentity_constraint_Definitions).

[top](#)

---

Generated by [oXygen](#) XML Editor using a modified version of [xs3p](#) that adds schema diagrams and chunking support.

# XML Schema Documentation

## Table of Contents

- [Schema Document Properties](#)
- [Global Declarations](#)
  - [Element: bulletPayment](#)
  - [Element: capFloor](#)
  - [Element: floatingRateCalculation](#)
  - [Element: fra](#)
  - [Element: inflationRateCalculation](#)
  - [Element: rateCalculation](#)
  - [Element: swap](#)
  - [Element: swaption](#)
- [Global Definitions](#)
  - [Complex Type: BondReference](#)
  - [Complex Type: BulletPayment](#)
  - [Complex Type: Calculation](#)
  - [Complex Type: CalculationPeriod](#)
  - [Complex Type: CalculationPeriodAmount](#)
  - [Complex Type: CalculationPeriodDates](#)
  - [Complex Type: CalculationPeriodDatesReference](#)
  - [Complex Type: CancelableProvision](#)
  - [Complex Type: CancelableProvisionAdjustedDates](#)
  - [Complex Type: CancellationEvent](#)
  - [Complex Type: CapFloor](#)
  - [Complex Type: CashPriceMethod](#)
  - [Complex Type: CashSettlement](#)
  - [Complex Type: CashSettlementPaymentDate](#)
  - [Complex Type: Cashflows](#)
  - [Complex Type: DateRelativeToCalculationPeriodDates](#)
  - [Complex Type: DateRelativeToPaymentDates](#)
  - [Complex Type: Discounting](#)
  - [Complex Type: EarlyTerminationEvent](#)
  - [Complex Type: EarlyTerminationProvision](#)
  - [Complex Type: ExerciseEvent](#)
  - [Complex Type: ExercisePeriod](#)
  - [Complex Type: ExtendibleProvision](#)
  - [Complex Type: ExtendibleProvisionAdjustedDates](#)
  - [Complex Type: ExtensionEvent](#)
  - [Complex Type: FallbackReferencePrice](#)
  - [Complex Type: FinalCalculationPeriodDateAdjustment](#)
  - [Complex Type: FloatingRateDefinition](#)
  - [Complex Type: Fra](#)
  - [Complex Type: FxFixingDate](#)
  - [Complex Type: FxLinkedNotionalAmount](#)
  - [Complex Type: FxLinkedNotionalSchedule](#)
  - [Complex Type: InflationRateCalculation](#)
  - [Complex Type: InterestRateStream](#)
  - [Complex Type: InterestRateStreamReference](#)
  - [Complex Type: MandatoryEarlyTermination](#)
  - [Complex Type: MandatoryEarlyTerminationAdjustedDates](#)
  - [Complex Type: NonDeliverableSettlement](#)
  - [Complex Type: Notional](#)
  - [Complex Type: NotionalStepRule](#)
  - [Complex Type: OptionalEarlyTermination](#)
  - [Complex Type: OptionalEarlyTerminationAdjustedDates](#)
  - [Complex Type: PaymentCalculationPeriod](#)
  - [Complex Type: PaymentDates](#)
  - [Complex Type: PaymentDatesReference](#)
  - [Complex Type: PriceSourceDisruption](#)
  - [Complex Type: PrincipalExchange](#)
  - [Complex Type: RelevantUnderlyingDateReference](#)
  - [Complex Type: ResetDates](#)

- [Complex Type: ResetDatesReference](#)
- [Complex Type: SettlementProvision](#)
- [Complex Type: SettlementRateOption](#)
- [Complex Type: SinglePartyOption](#)
- [Complex Type: StubCalculationPeriodAmount](#)
- [Complex Type: Swap](#)
- [Complex Type: SwapAdditionalTerms](#)
- [Complex Type: Swaption](#)
- [Complex Type: SwaptionAdjustedDates](#)
- [Complex Type: ValuationDatesReference](#)
- [Complex Type: ValuationPostponement](#)
- [Complex Type: YieldCurveMethod](#)
- [Model Group: DiscountRate.model](#)
- [Model Group: MandatoryEarlyTermination.model](#)
- [Model Group: OptionalEarlyTermination.model](#)

- [Legend](#)

- [Glossary](#)

[top](#)

## Schema Document Properties

Target Namespace	<a href="http://www.fpml.org/FpML-5/confirmation">http://www.fpml.org/FpML-5/confirmation</a>
Version	\$Revision: 2806 \$
Element and Attribute Namespaces	<ul style="list-style-type: none"> <li>◦ Global element and attribute declarations belong to this schema's target namespace.</li> <li>◦ By default, local element declarations belong to this schema's target namespace.</li> <li>◦ By default, local attribute declarations have no namespace.</li> </ul>
Schema Composition	<ul style="list-style-type: none"> <li>◦ This schema includes components from the following schema document(s):           <ul style="list-style-type: none"> <li>◦ <a href="#">fpml-asset-5-0.xsd</a></li> </ul> </li> </ul>

## Declared Namespaces

Prefix	Namespace
Default namespace	<a href="http://www.fpml.org/FpML-5/confirmation">http://www.fpml.org/FpML-5/confirmation</a>
xml	<a href="http://www.w3.org/XML/1998/namespace">http://www.w3.org/XML/1998/namespace</a>
fpml-annotation	<a href="http://www.fpml.org/annotation">http://www.fpml.org/annotation</a>
xsd	<a href="http://www.w3.org/2001/XMLSchema">http://www.w3.org/2001/XMLSchema</a>
ecore	<a href="http://www.eclipse.org/emf/2002/Ecore">http://www.eclipse.org/emf/2002/Ecore</a>

## Schema Component Representation

```
<xsd:schema targetNamespace="http://www.fpml.org/FpML-5/confirmation"
documentRoot="FpML" nsPrefix="conf" package="org.fpml.confirmation" version="$Revision: 2806 "
$" elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xsd:include schemaLocation="fpml-asset-5-0.xsd"/>
  ...
</xsd:schema>
```

[top](#)

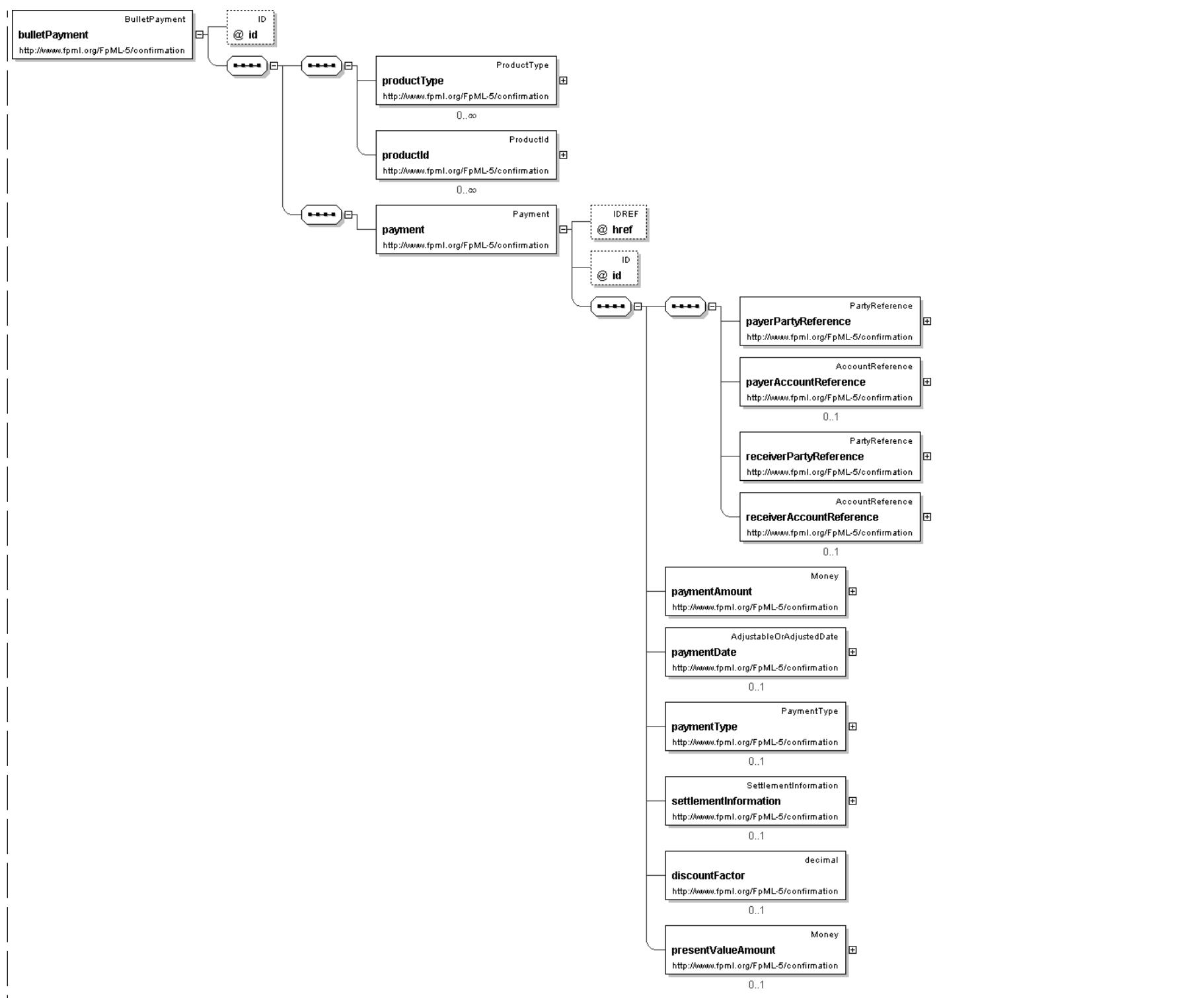
## Global Declarations

### Element: bulletPayment

- This element can be used wherever the following element is referenced:
  - [product](#)

Name	bulletPayment
Type	<a href="#">BulletPayment</a>
Nillable	no
Abstract	no
Documentation	A product to represent a single known payment.

## Logical Diagram



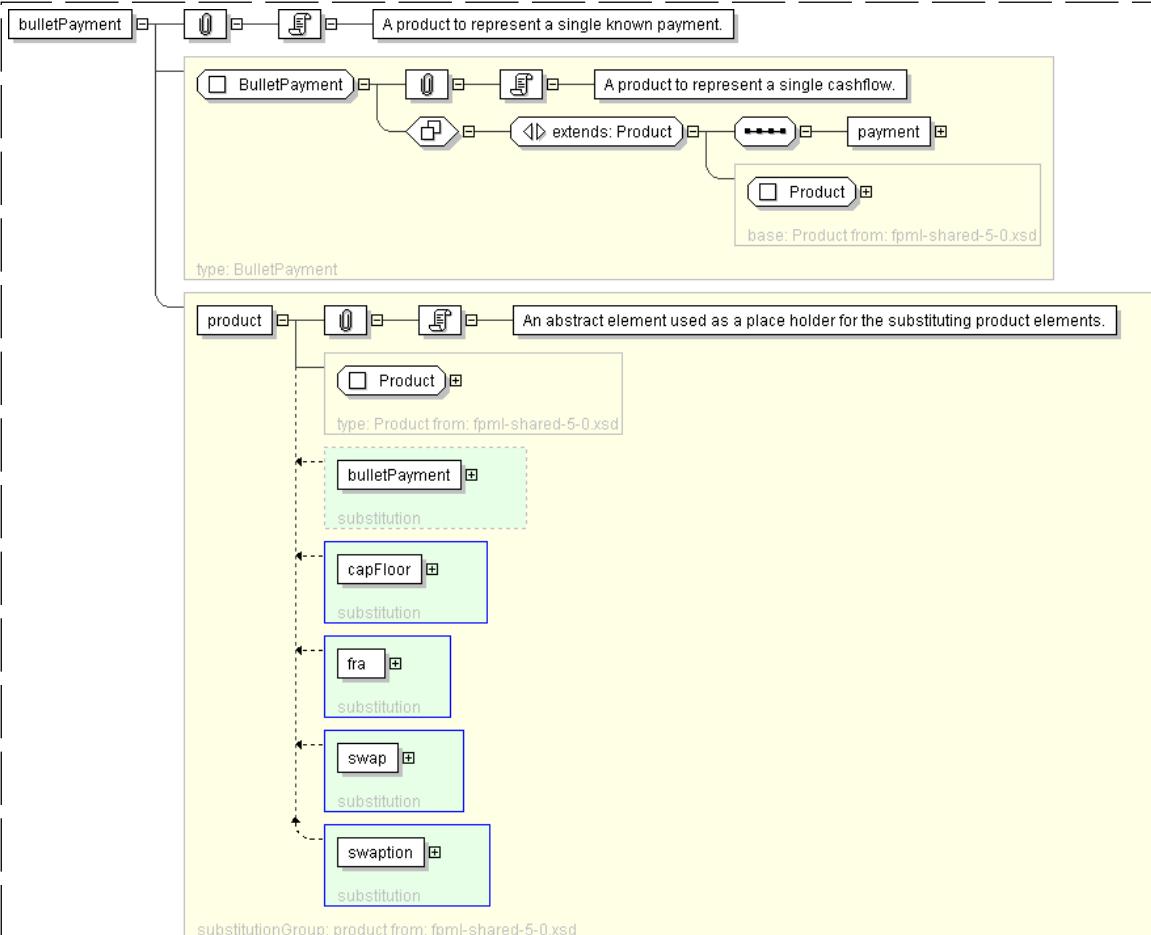
**XML Instance Representation**

```
<bulletPayment
  id=" xsd:ID [0..1]">
  <productType> ProductType </productType> [0..*]
  'A classification of the type of product. FpML defines a simple product categorization using
  a coding scheme.'

  <productId> ProductId </productId> [0..*]
  'A product 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.'

  <payment> Payment </payment> [1]
  'A known payment between two parties.'

</bulletPayment>
```

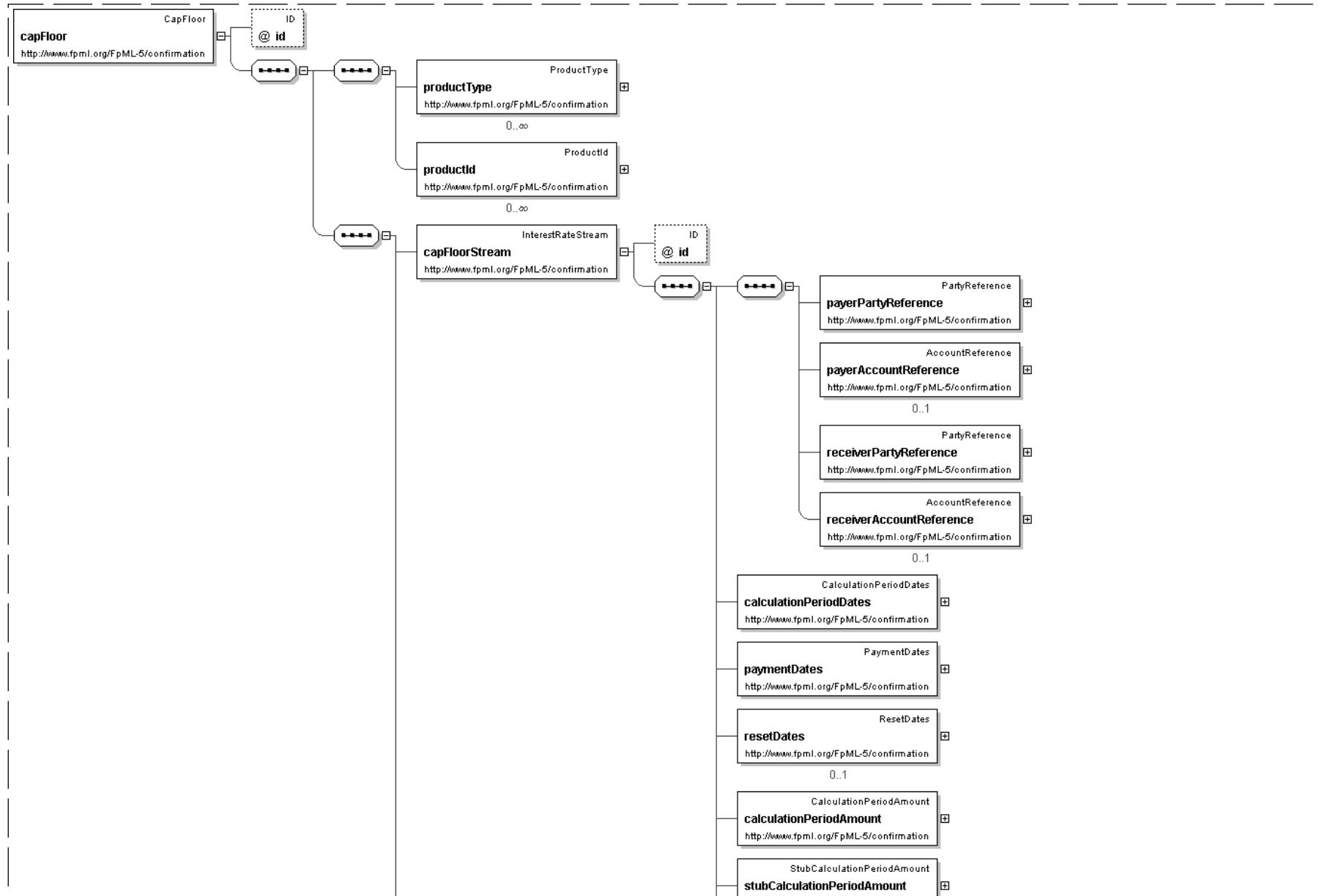
**Diagram****Schema Component Representation**

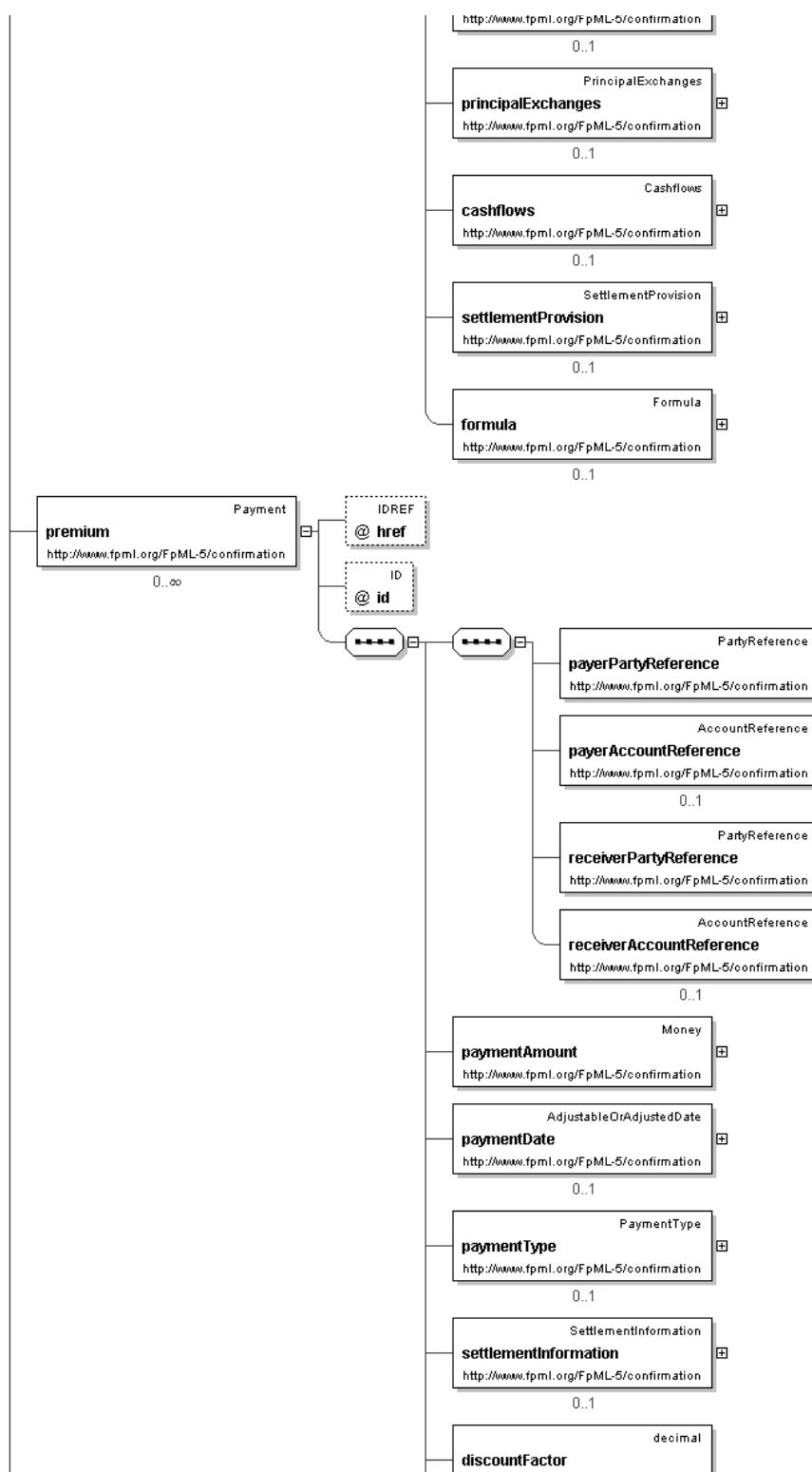
```
<xsd:element name="bulletPayment" type=" BulletPayment " substitutionGroup="product" />
```

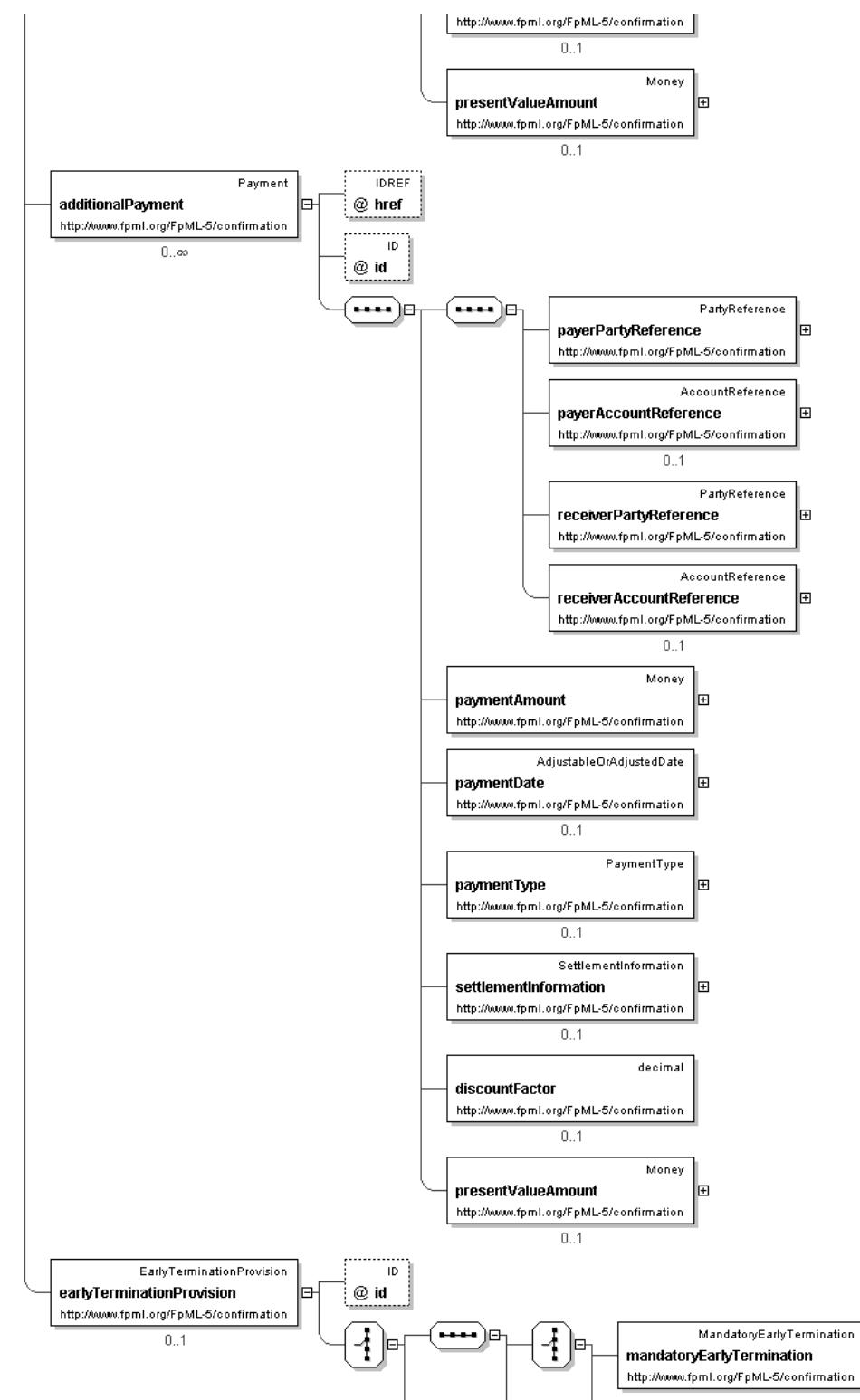
**Element: capFloor**

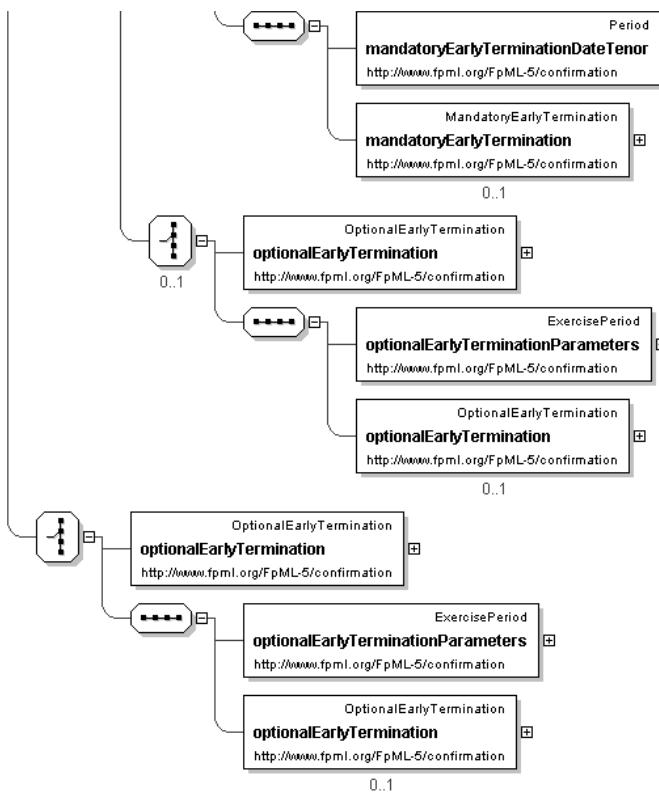
- This element can be used wherever the following element is referenced:
  - product

<b>Name</b>	capFloor
<b>Type</b>	<a href="#">CapFloor</a>
<b>Nillable</b>	no
<b>Abstract</b>	no
<b>Documentation</b>	A cap, floor or cap floor structures product definition.

**Logical Diagram**





**XML Instance Representation**

```

<capFloor
  id=" xsd:ID [0..1]">
  <productType> ProductType </productType> [0..*]
  'A classification of the type of product. FpML defines a simple product categorization using
  a coding scheme.'

  <productId> ProductId </productId> [0..*]
  'A product 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.'

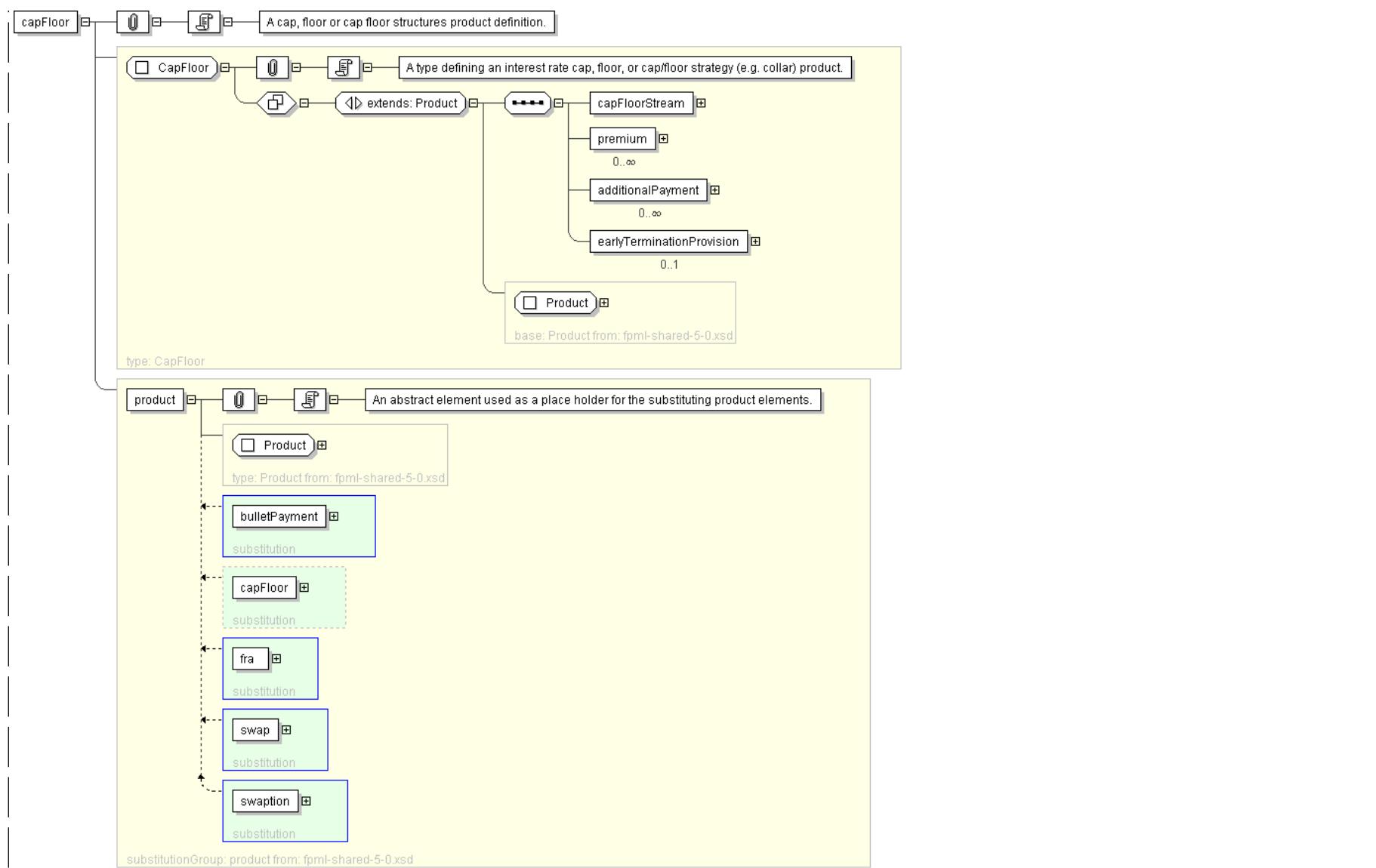
  <capFloorStream> InterestRateStream </capFloorStream> [1]
  <premium> Payment </premium> [0..*]
  'The option premium amount payable by buyer to seller on the specified payment date.'

  <additionalPayment> Payment </additionalPayment> [0..*]
  'Additional payments between the principal parties.'

  <earlyTerminationProvision> EarlyTerminationProvision </earlyTerminationProvision> [0..1]
  'Parameters specifying provisions relating to the optional and mandatory early terminarion of
  a CapFloor transaction.'

</capFloor>
  
```

**Diagram**

**Schema Component Representation**

```
<xsd:element name="capFloor" type=" CapFloor " substitutionGroup="product" />
```

top

**Element: floatingRateCalculation**

- This element can be used wherever the following element is referenced:
  - rateCalculation

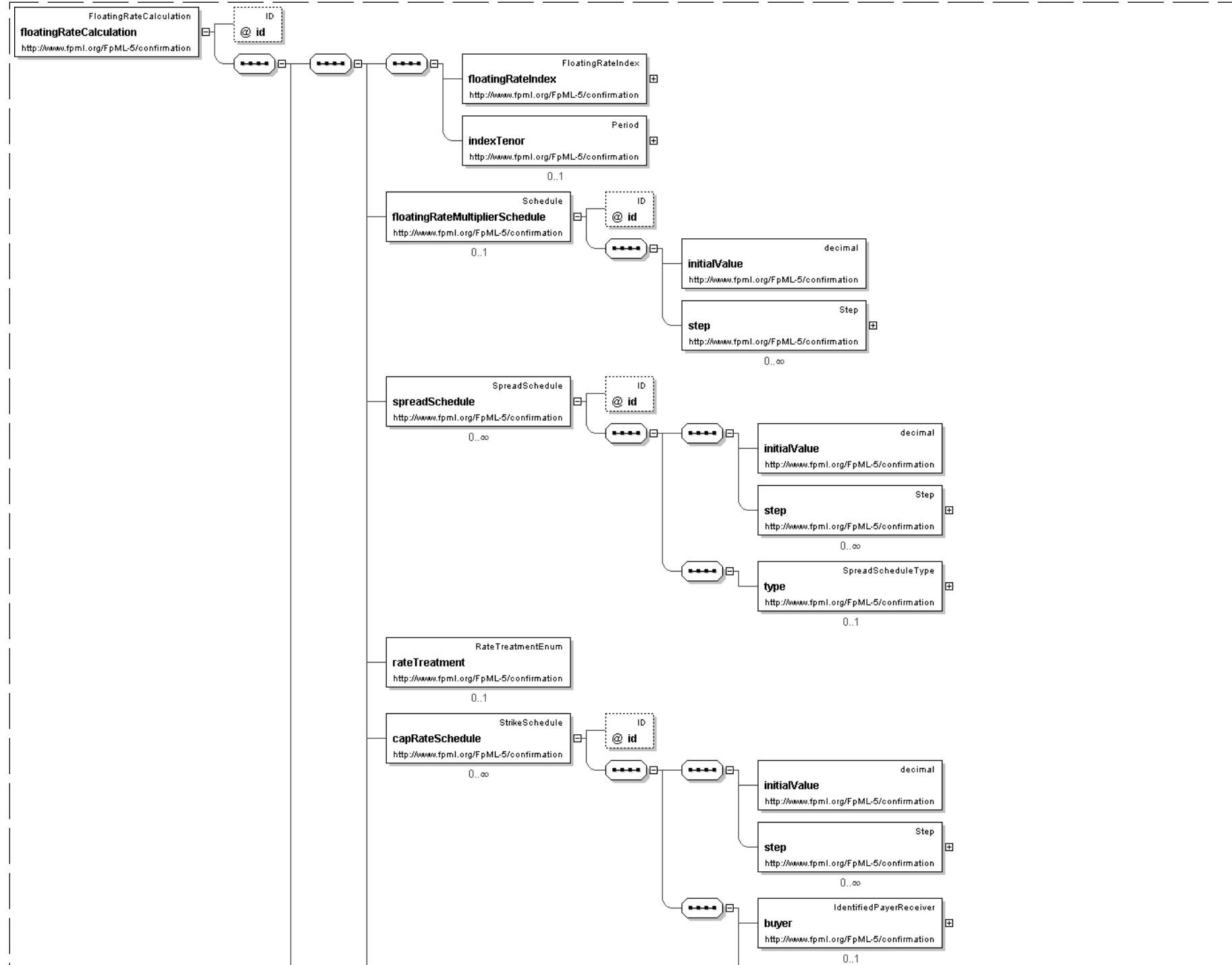
Name	floatingRateCalculation
Type	FloatingRateCalculation
Nullable	no

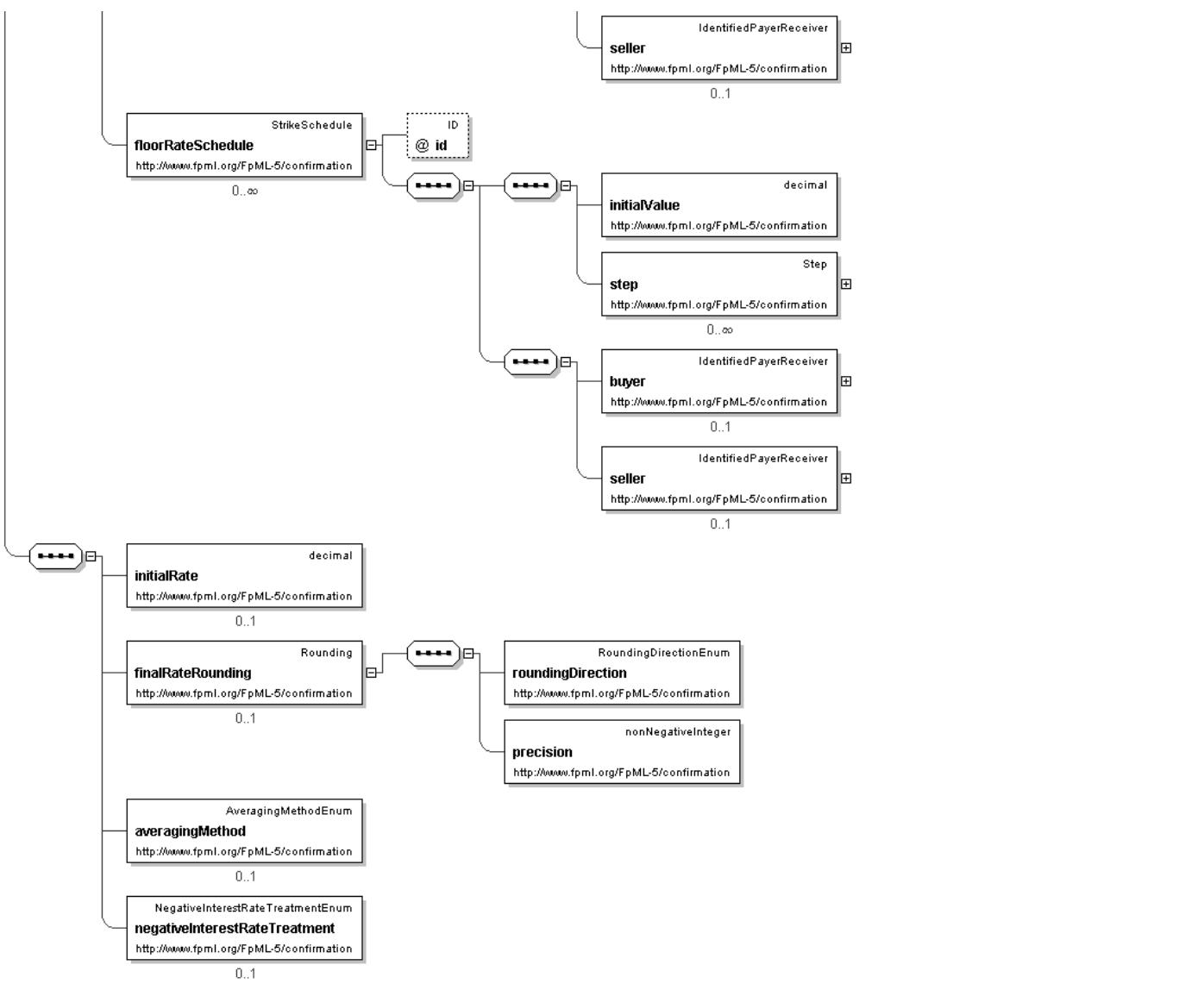
**Abstract**

no

**Documentation**

A floating rate calculation definition.

**Logical Diagram**

**XML Instance Representation**

```

<floatingRateCalculation
id=" xsd:ID [0..1]">
  <floatingRateIndex> FloatingRateIndex </floatingRateIndex> [1]
  <indexTenor> Period </indexTenor> [0..1]
  'The ISDA Designated Maturity, i.e. the tenor of the floating rate.'

  <floatingRateMultiplierSchedule> Schedule </floatingRateMultiplierSchedule> [0..1]
  'A rate multiplier or multiplier schedule to apply to the floating rate. A multiplier
  schedule is expressed as explicit multipliers and dates. In the case of a schedule, the
  step dates may be subject to adjustment in accordance with any adjustments specified in
  the calculationPeriodDatesAdjustments. The multiplier can be a positive or negative
  decimal. This element should only be included if the multiplier is not equal to 1 (one) for
  the term of the stream.'

  <spreadSchedule> SpreadSchedule </spreadSchedule> [0..*]

```

'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> [RateTreatmentEnum](#) </rateTreatment> [0..1]

'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> [StrikeSchedule](#) </capRateSchedule> [0..\*]

'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 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> [StrikeSchedule](#) </floorRateSchedule> [0..\*]

'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.'

<initialRate> [xsd:decimal](#) </initialRate> [0..1]

'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> [Rounding](#) </finalRateRounding> [0..1]

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

<averagingMethod> [AveragingMethodEnum](#) </averagingMethod> [0..1]

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

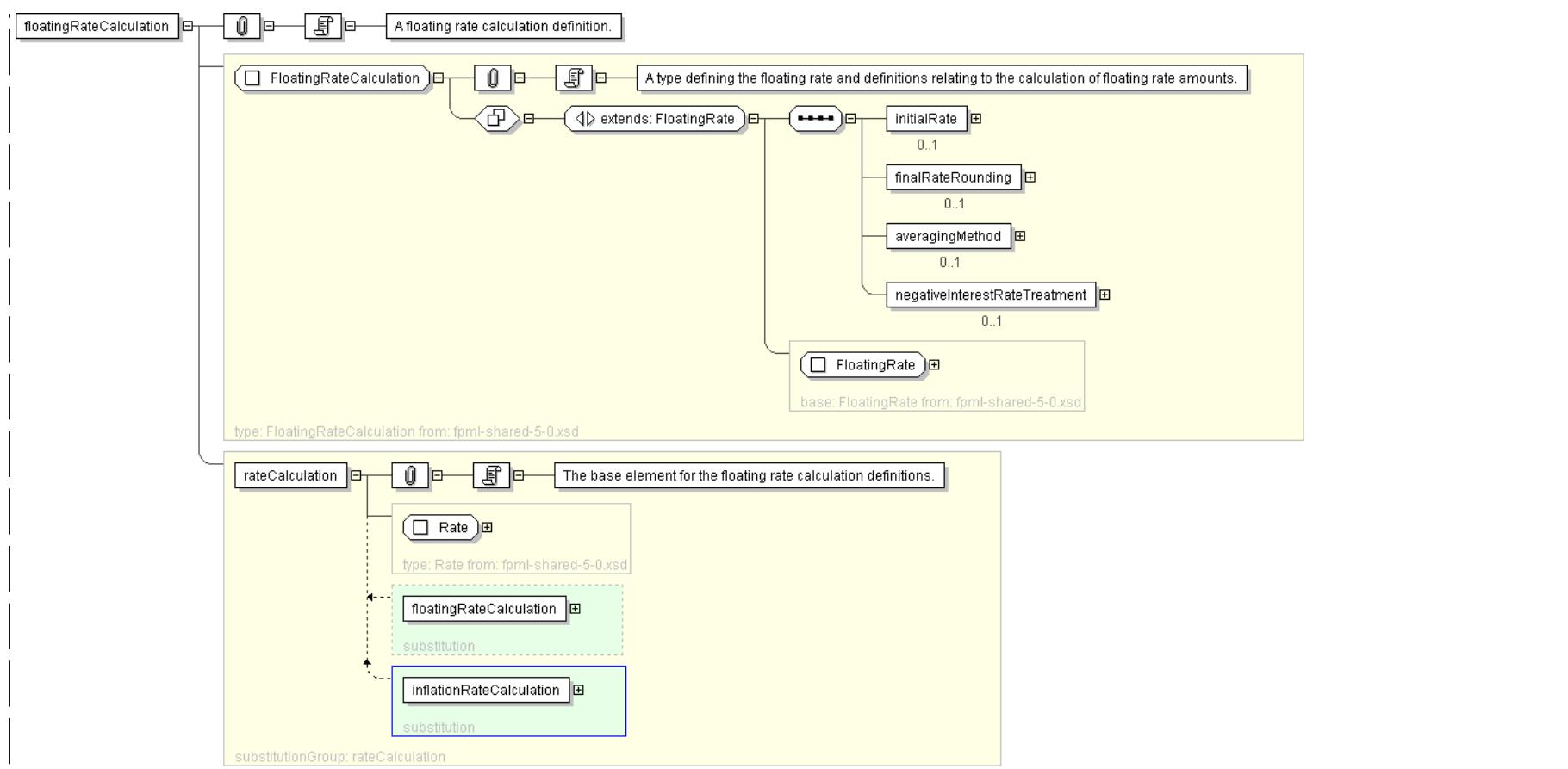
<negativeInterestRateTreatment> [NegativeInterestRateTreatmentEnum](#)

</negativeInterestRateTreatment> [0..1]

'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).'

</floatingRateCalculation>

## Diagram

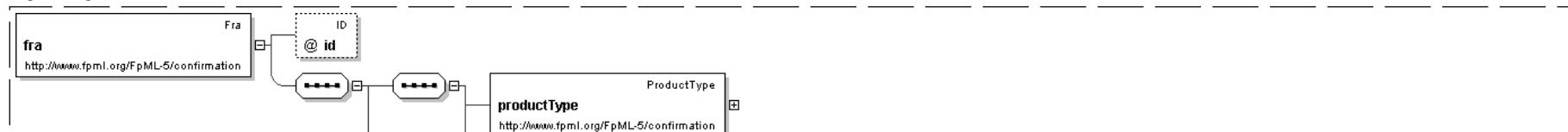
**Schema Component Representation**

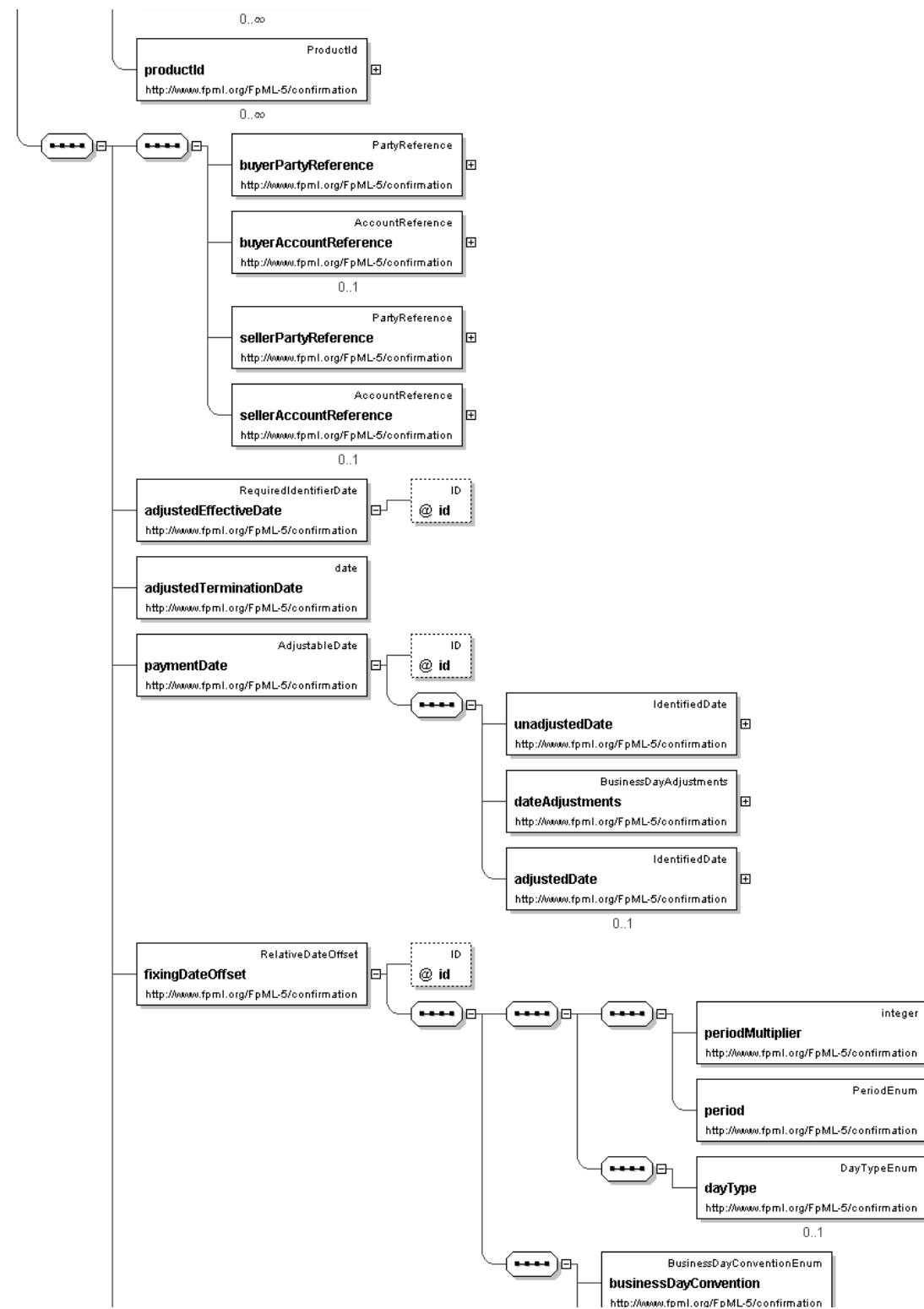
```
<xsd:element name="floatingRateCalculation" type=" FloatingRateCalculation
  " substitutionGroup="rateCalculation"/>
```

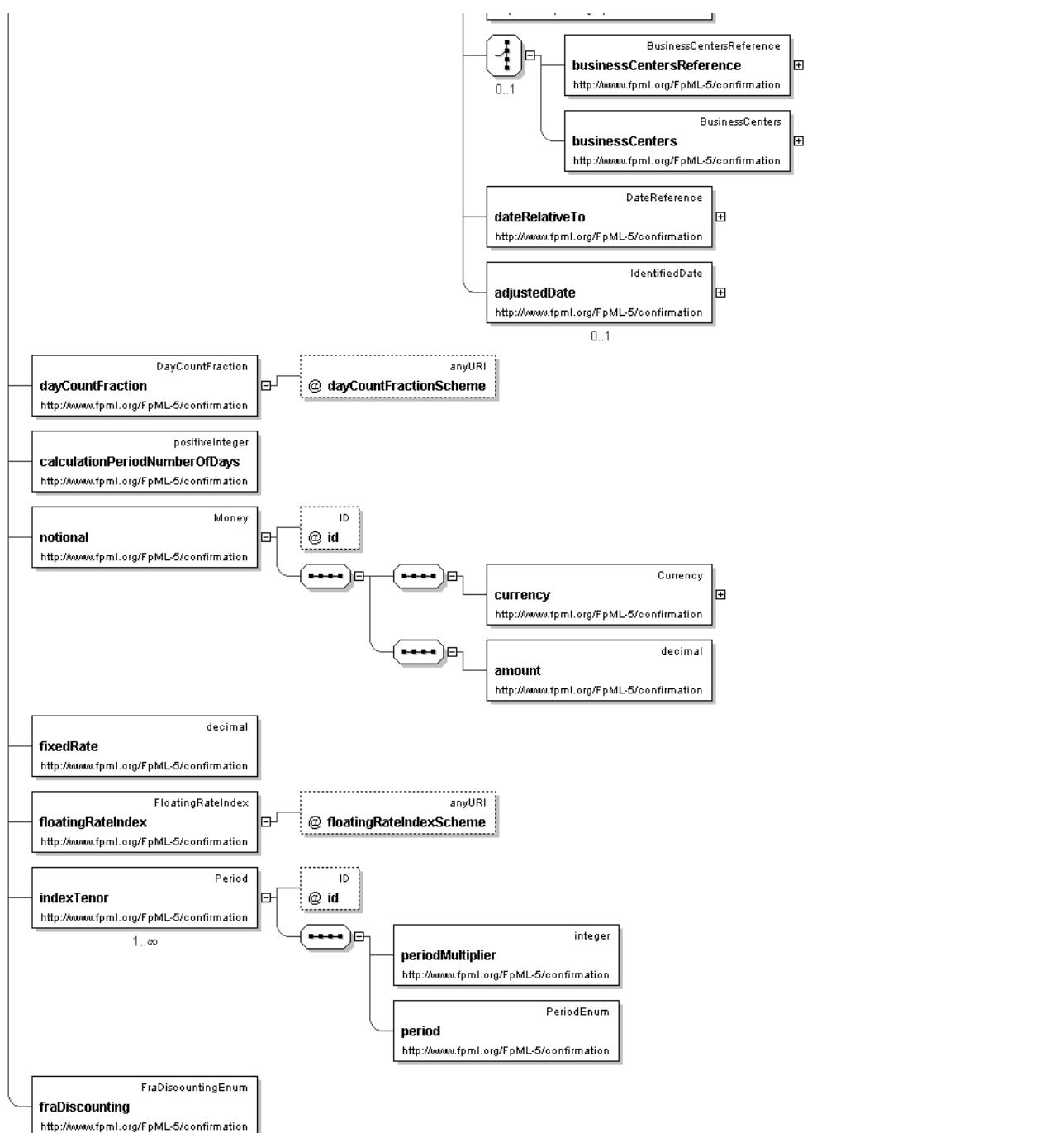
[top](#)**Element: fra**

- This element can be used wherever the following element is referenced:
  - product

Name	fra
Type	Fra
Nillable	no
Abstract	no
Documentation	A forward rate agreement product definition.

**Logical Diagram**



**XML Instance Representation**

```

<fra
  id="#_ID [0..1]">
  <productType> ProductType </productType> [0..*]
  'A classification of the type of product. FpML defines a simple product categorization using

```

*a coding scheme.'*

**<productId>** [ProductId](#) **</productId>** [0..\*]  
*'A product 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.'*

**<buyerPartyReference>** [PartyReference](#) **</buyerPartyReference>** [1]  
*'A reference to the party that buys this instrument, i.e. pays for this instrument and receives the rights defined by it. See 2000 ISDA definitions Article 11.1 (b). In the case of FRAs this is the fixed rate payer.'*

**<buyerAccountReference>** [AccountReference](#) **</buyerAccountReference>** [0..1]  
*'A reference to the account that buys this instrument.'*

**<sellerPartyReference>** [PartyReference](#) **</sellerPartyReference>** [1]  
*'A reference to the party that sells (\"writes\") this instrument, i.e. that grants the rights defined by this instrument and in return receives a payment for it. See 2000 ISDA definitions Article 11.1 (a). In the case of FRAs this is the floating rate payer.'*

**<sellerAccountReference>** [AccountReference](#) **</sellerAccountReference>** [0..1]  
*'A reference to the account that sells this instrument.'*

**<adjustedEffectiveDate>** [RequiredIdentifierDate](#) **</adjustedEffectiveDate>** [1]  
*'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>** [xsd:date](#) **</adjustedTerminationDate>** [1]  
*'The end date of the calculation period. This date should already be adjusted for any applicable business day convention.'*

**<paymentDate>** [AdjustableDate](#) **</paymentDate>** [1]  
*'The payment date. This date is subject to adjustment in accordance with any applicable business day convention.'*

**<fixingDateOffset>** [RelativeDateOffset](#) **</fixingDateOffset>** [1]  
*'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>** [DayCountFraction](#) **</dayCountFraction>** [1]  
*'The day count fraction.'*

**<calculationPeriodNumberOfDays>** [xsd:positiveInteger](#) **</calculationPeriodNumberOfDays>** [1]  
*'The number of days from the adjusted effective date to the adjusted termination date calculated in accordance with the applicable day count fraction.'*

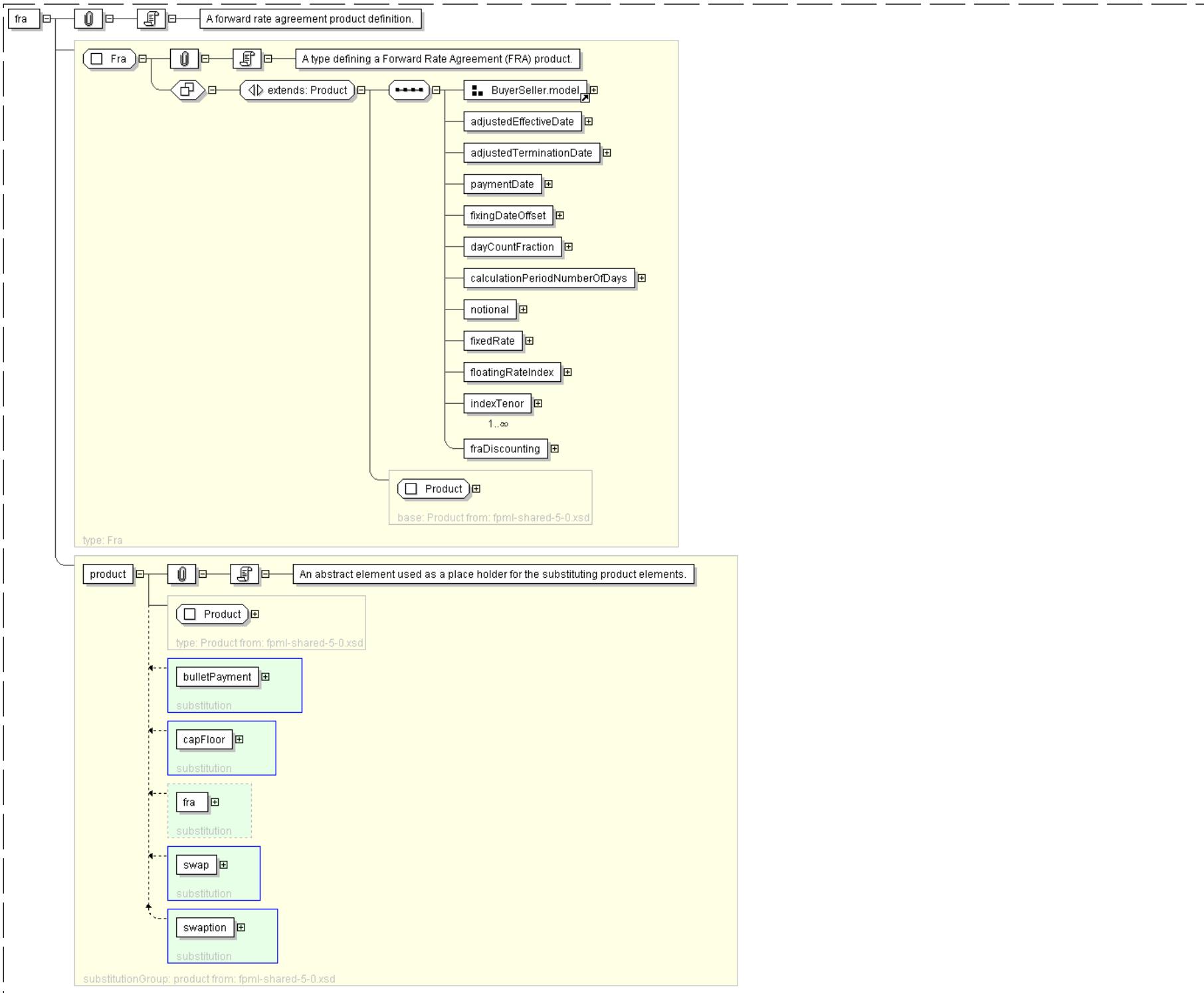
**<notional>** [Money](#) **</notional>** [1]  
*'The notional amount.'*

**<fixedRate>** [xsd:decimal](#) **</fixedRate>** [1]  
*'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>** [FloatingRateIndex](#) **</floatingRateIndex>** [1]  
**<indexTenor>** [Period](#) **</indexTenor>** [1..\*]  
*'The ISDA Designated Maturity, i.e. the tenor of the floating rate.'*

**<fraDiscounting>** [FraDiscountingEnum](#) **</fraDiscounting>** [1]  
*'Specifies whether discounting applies and, if so, what type.'*

**</fra>**

**Diagram**

## Schema Component Representation

```
<xsd:element name="fra" type=" Fra " substitutionGroup="product"/>
```

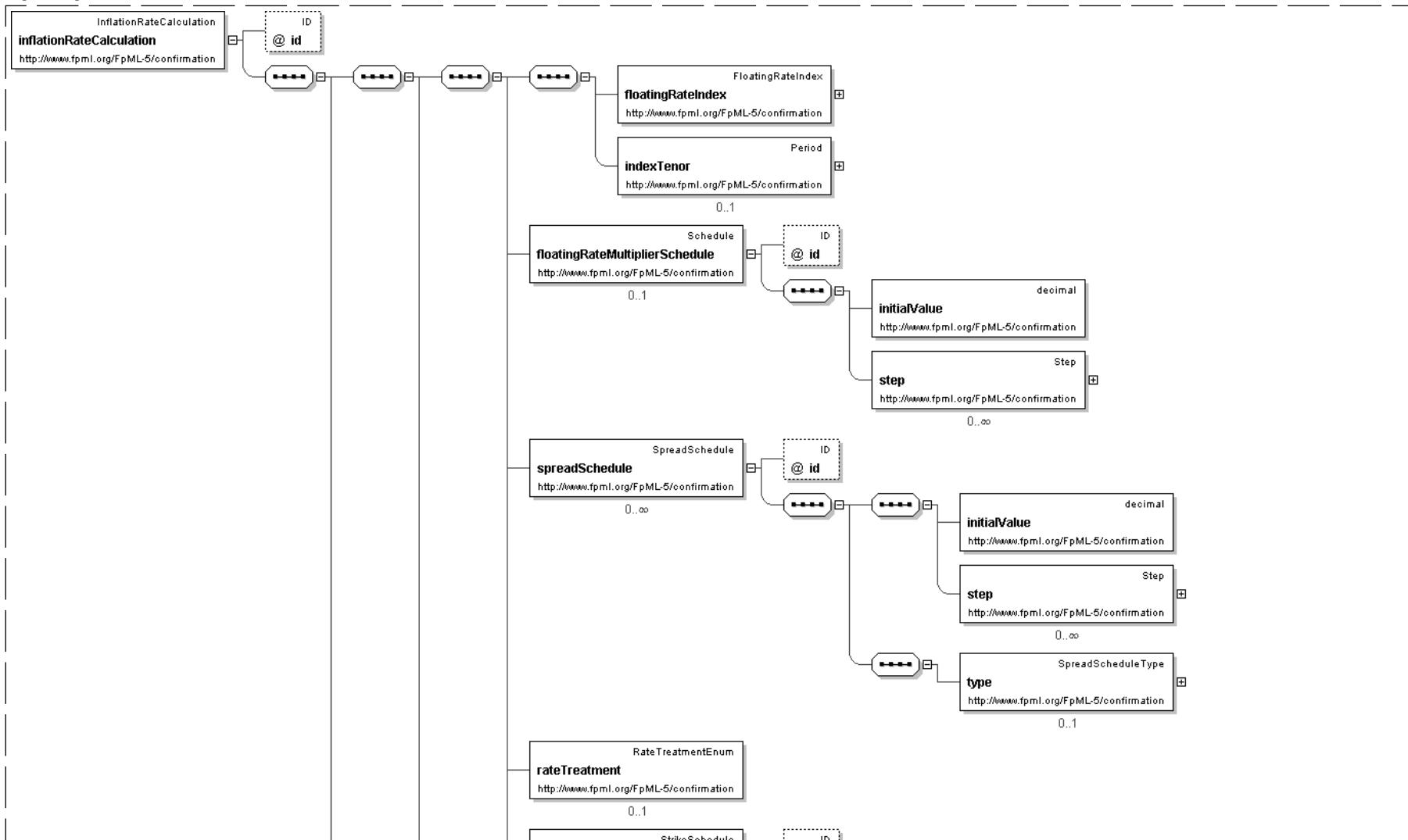
top

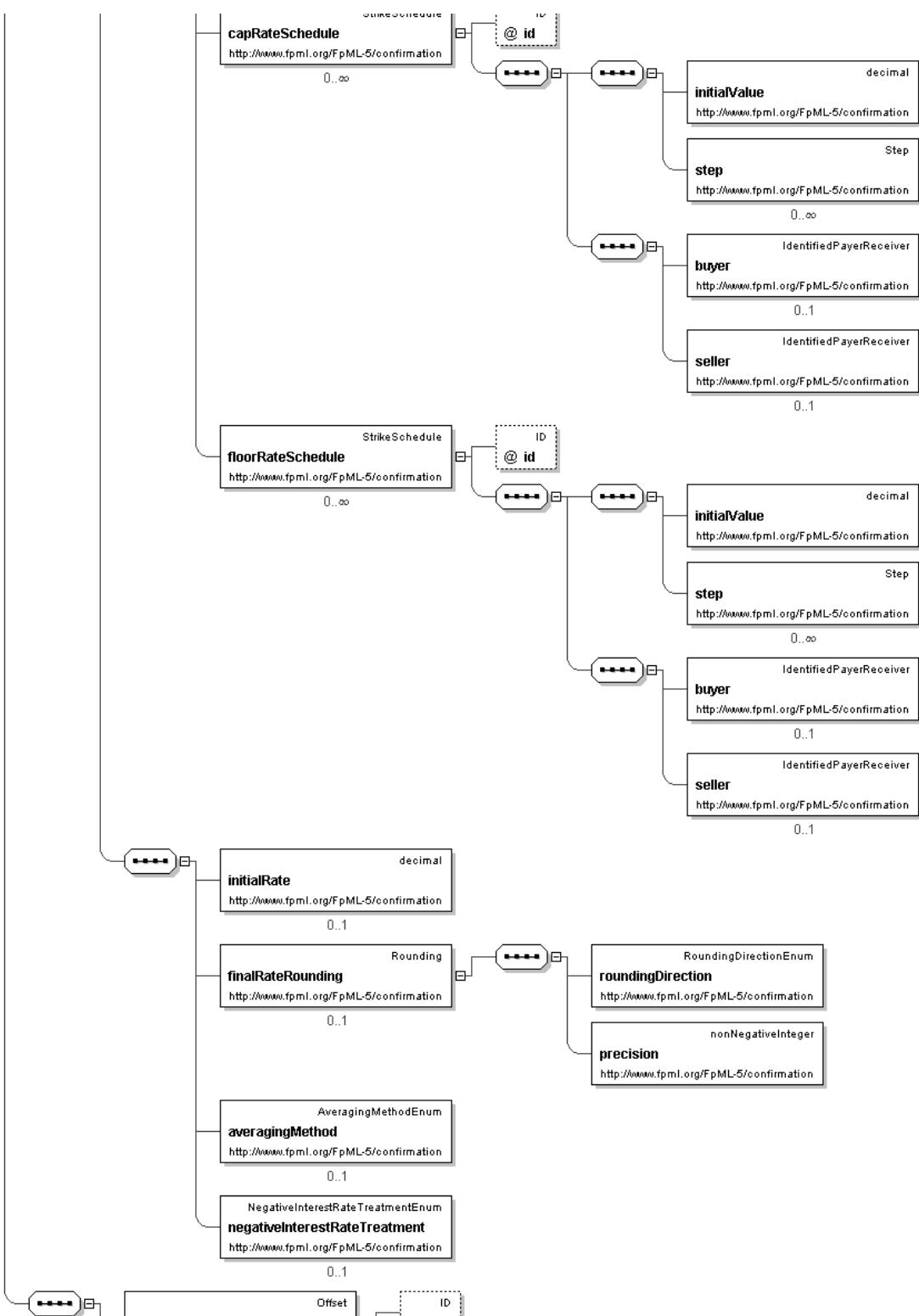
## Element: inflationRateCalculation

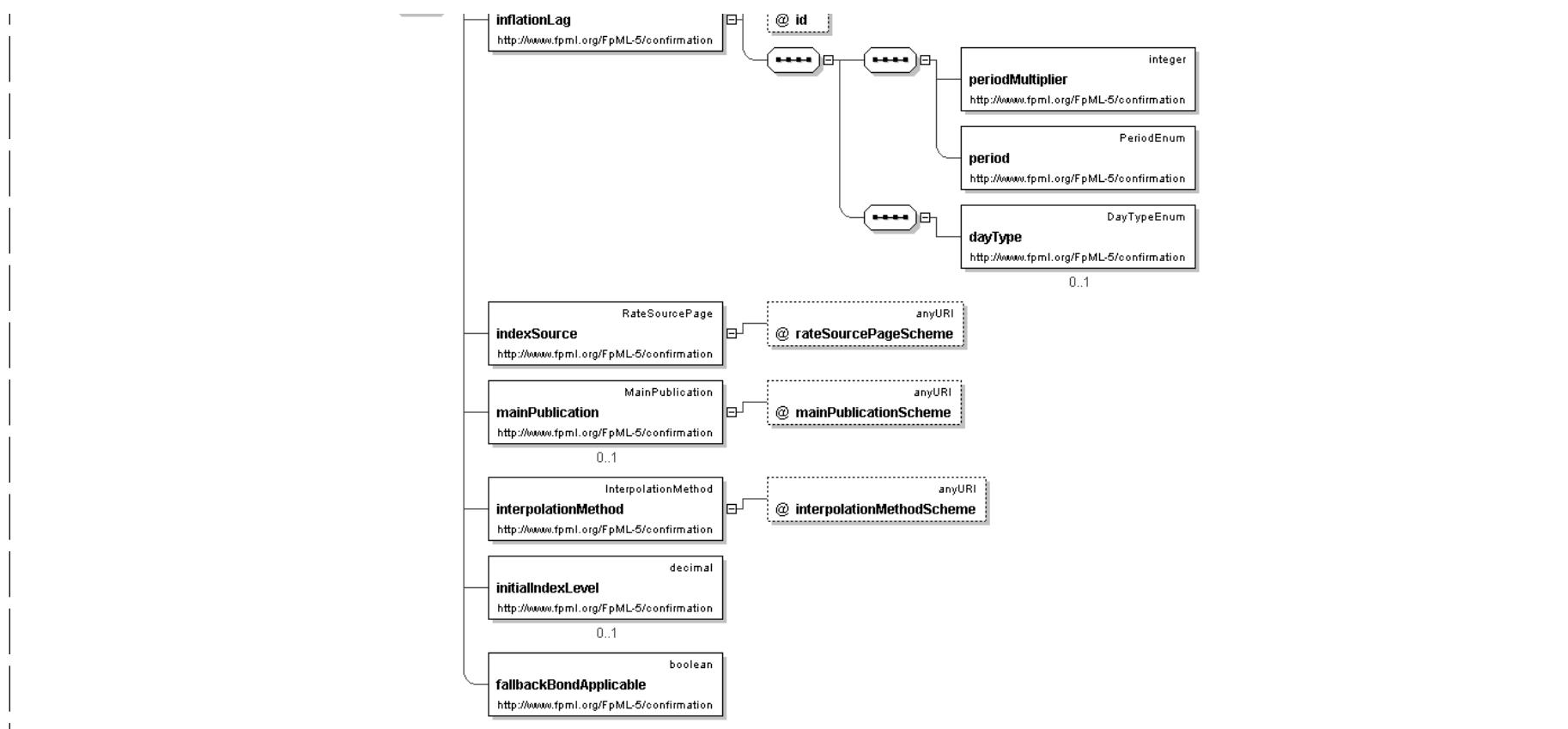
- This element can be used wherever the following element is referenced
    - `rateCalculation`

<b>Name</b>	inflationRateCalculation
<b>Type</b>	<a href="#">InflationRateCalculation</a>
<b>Nillable</b>	no
<b>Abstract</b>	no
<b>Documentation</b>	An inflation rate calculation definition.

## Logical Diagram





**XML Instance Representation**

```

<inflationRateCalculation
  id=" xsd:ID [0..1]">
  <floatingRateIndex> FloatingRateIndex </floatingRateIndex> [1]
  <indexTenor> Period </indexTenor> [0..1]
  'The ISDA Designated Maturity, i.e. the tenor of the floating rate.'

  <floatingRateMultiplierSchedule> Schedule </floatingRateMultiplierSchedule> [0..1]
  'A rate multiplier or multiplier schedule to apply to the floating rate. A multiplier
  schedule is expressed as explicit multipliers and dates. In the case of a schedule, the
  step dates may be subject to adjustment in accordance with any adjustments specified in
  the calculationPeriodDatesAdjustments. The multiplier can be a positive or negative
  decimal. This element should only be included if the multiplier is not equal to 1 (one) for
  the term of the stream.'

  <spreadSchedule> SpreadSchedule </spreadSchedule> [0..*]
  '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> RateTreatmentEnum </rateTreatment> [0..1]
  '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> [StrikeSchedule](#) </capRateSchedule> [0..\*]

'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 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> [StrikeSchedule](#) </floorRateSchedule> [0..\*]

'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.'

<initialRate> [xsd:decimal](#) </initialRate> [0..1]

'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> [Rounding](#) </finalRateRounding> [0..1]

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

<averagingMethod> [AveragingMethodEnum](#) </averagingMethod> [0..1]

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

<negativeInterestRateTreatment> [NegativeInterestRateTreatmentEnum](#)

</negativeInterestRateTreatment> [0..1]

'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).'

<inflationLag> [Offset](#) </inflationLag> [1]

'an offsetting period from the payment date which determines the reference period for which the inflation index is observed.'

<indexSource> [RateSourcePage](#) </indexSource> [1]

'The reference source such as Reuters or Bloomberg.'

<mainPublication> [MainPublication](#) </mainPublication> [0..1]

'The current main publication source such as relevant web site or a government body.'

<interpolationMethod> [InterpolationMethod](#) </interpolationMethod> [1]

'The method used when calculating the Inflation Index Level from multiple points - the most common is Linear.'

<initialIndexLevel> [xsd:decimal](#) </initialIndexLevel> [0..1]

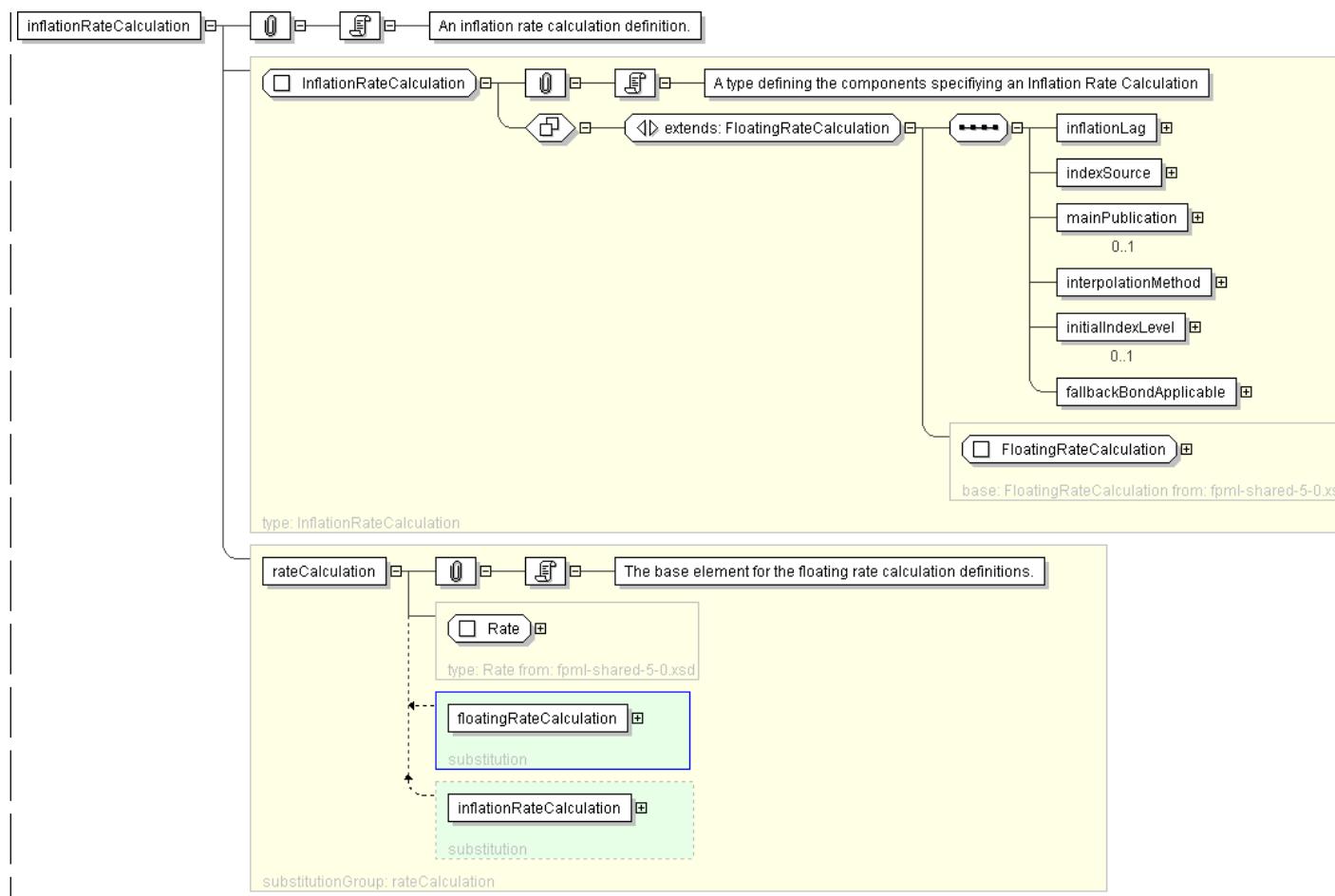
'initial known index level for the first calculation period.'

<fallbackBondApplicable> [xsd:boolean](#) </fallbackBondApplicable> [1]

'The applicability of a fallback bond as defined in the 2006 ISDA Inflation Derivatives Definitions, sections 1.3 and 1.8. Omission of this element implies a value of true.'

</inflationRateCalculation>

## Diagram

**Schema Component Representation**

```
<xsd:element name="inflationRateCalculation" type="InflationRateCalculation"
  "substitutionGroup="rateCalculation"/>
```

top

**Element: rateCalculation**

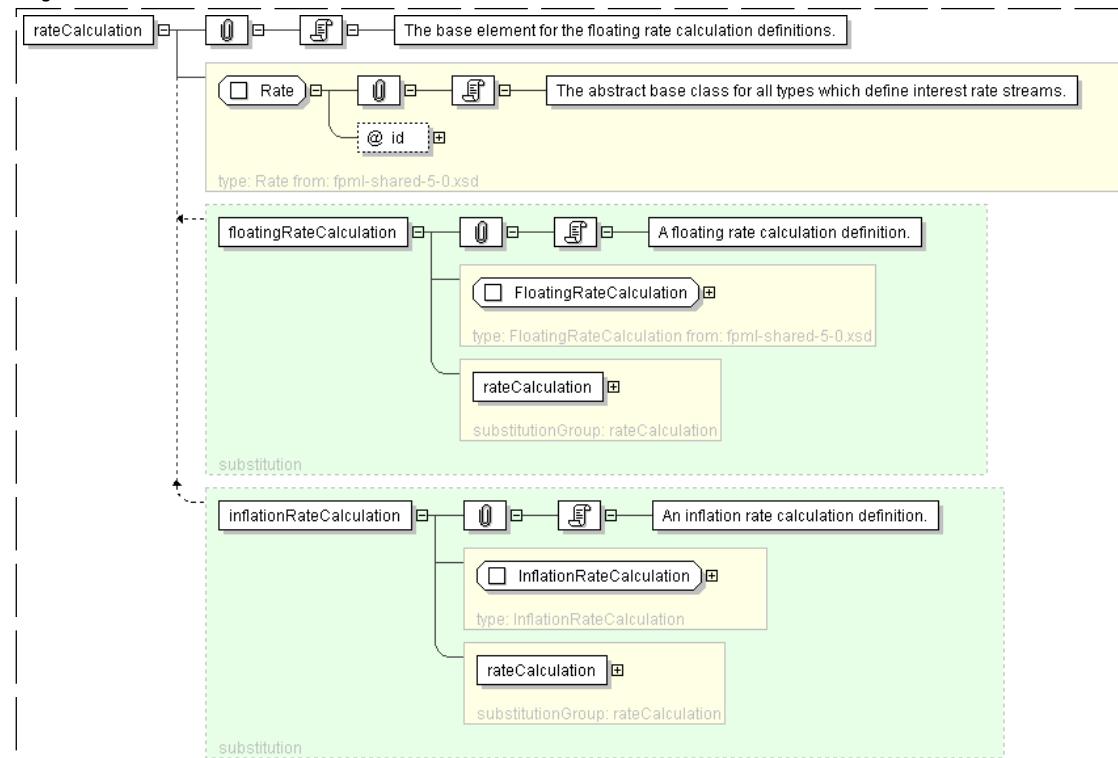
- The following elements can be used wherever this element is referenced:
  - floatingRateCalculation
  - inflationRateCalculation

Name	rateCalculation
Used by (from the same schema document)	Complex Type <a href="#">Calculation</a>
Type	<a href="#">Rate</a>
Nillable	no
Abstract	yes
Documentation	The base element for the floating rate calculation definitions.

**Logical Diagram**

**XML Instance Representation**

```
<rateCalculation
  id=" xsd:id [0..1]" />
```

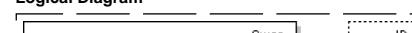
**Diagram****Schema Component Representation**

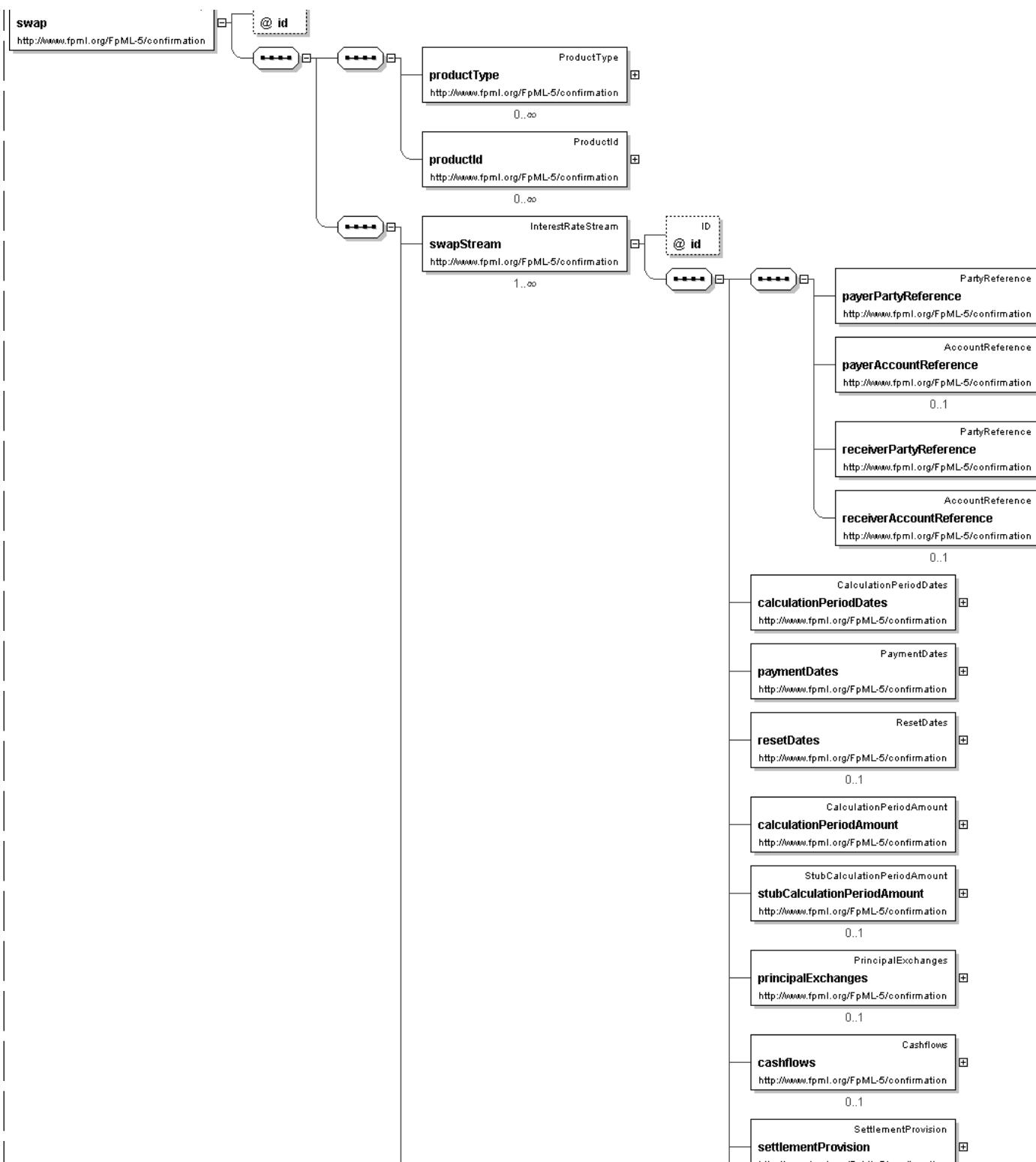
```
<xsd:element name="rateCalculation" type=" Rate " abstract="true"/>
```

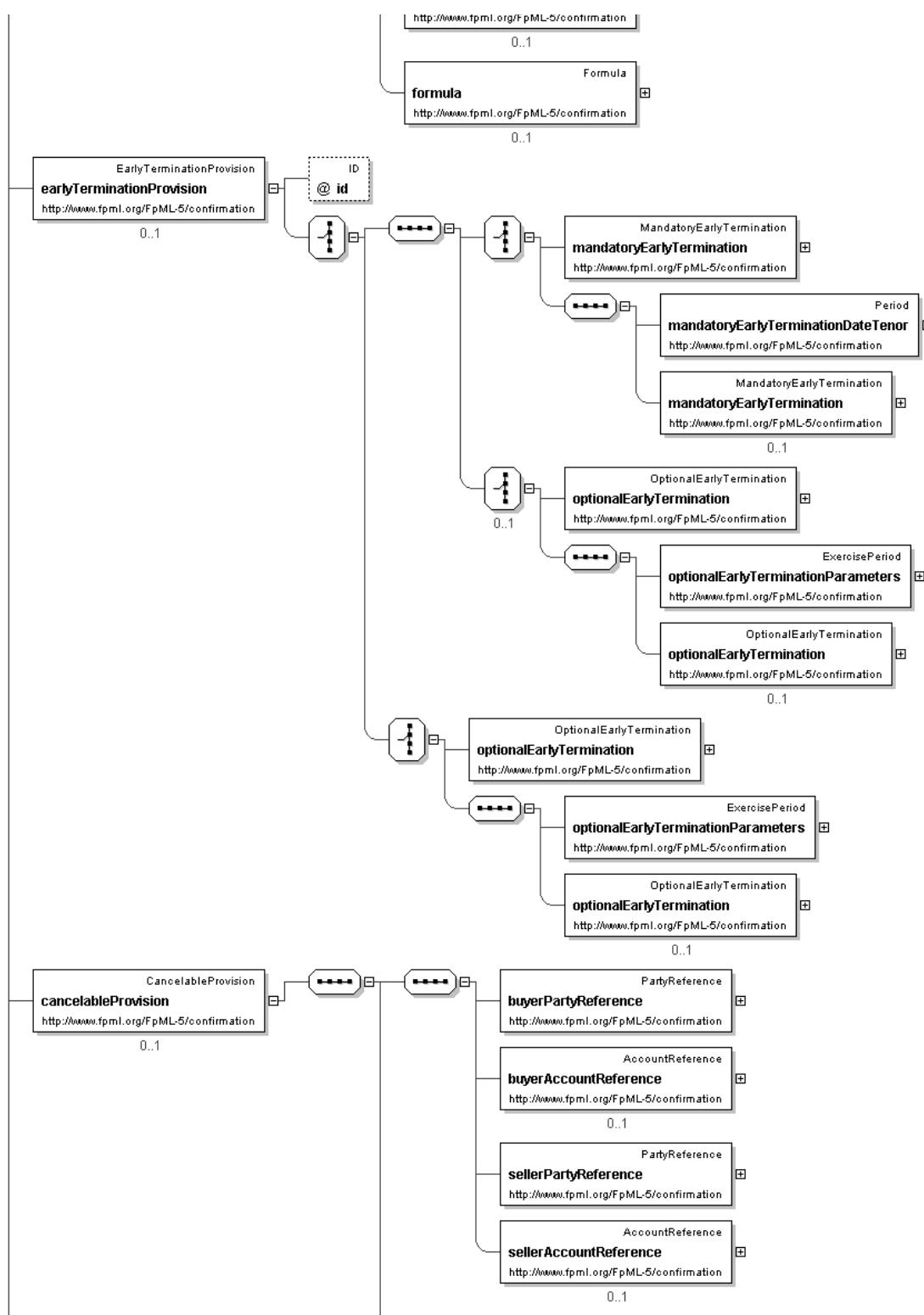
[top](#)**Element: swap**

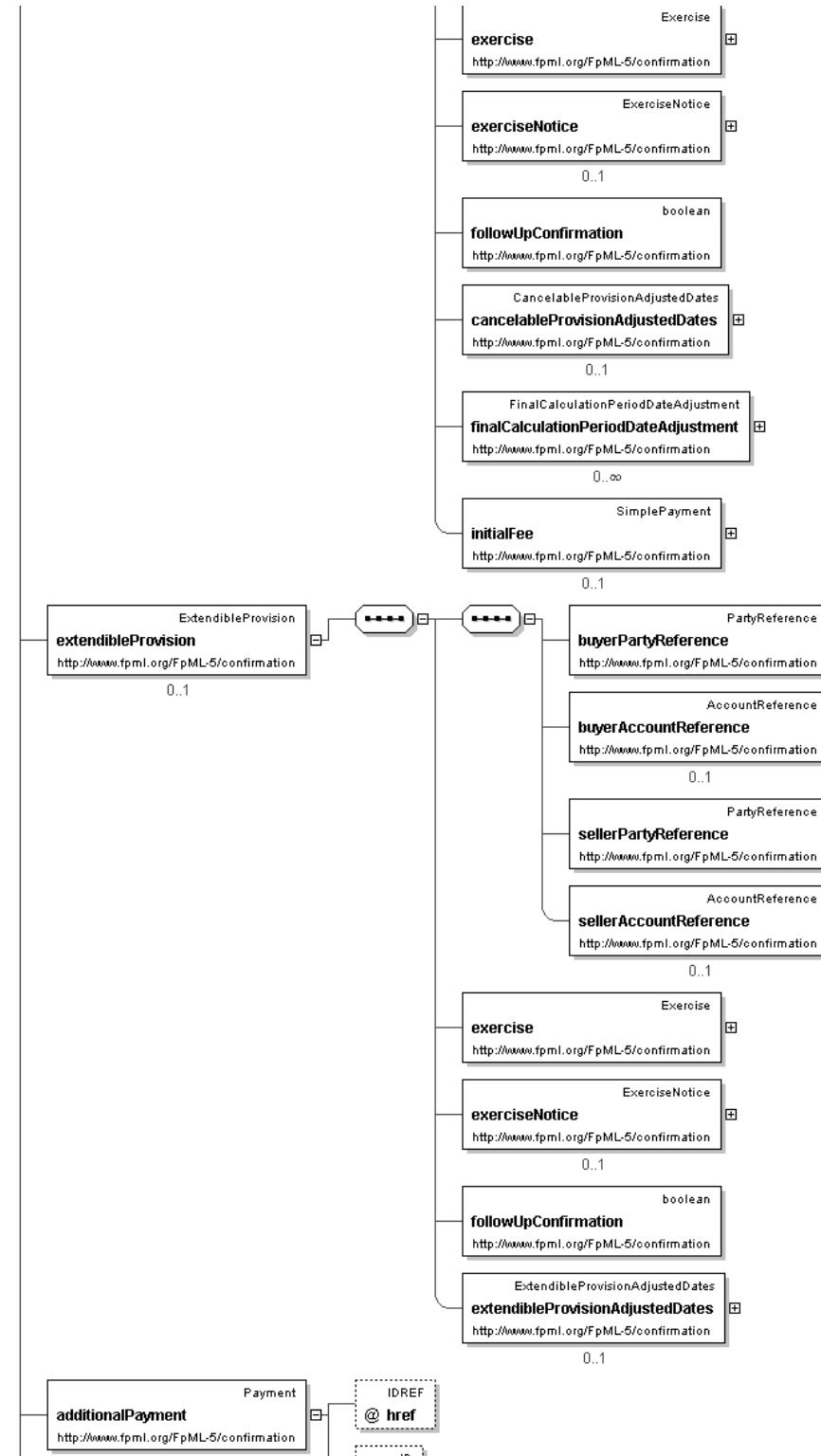
- This element can be used wherever the following element is referenced:
  - product

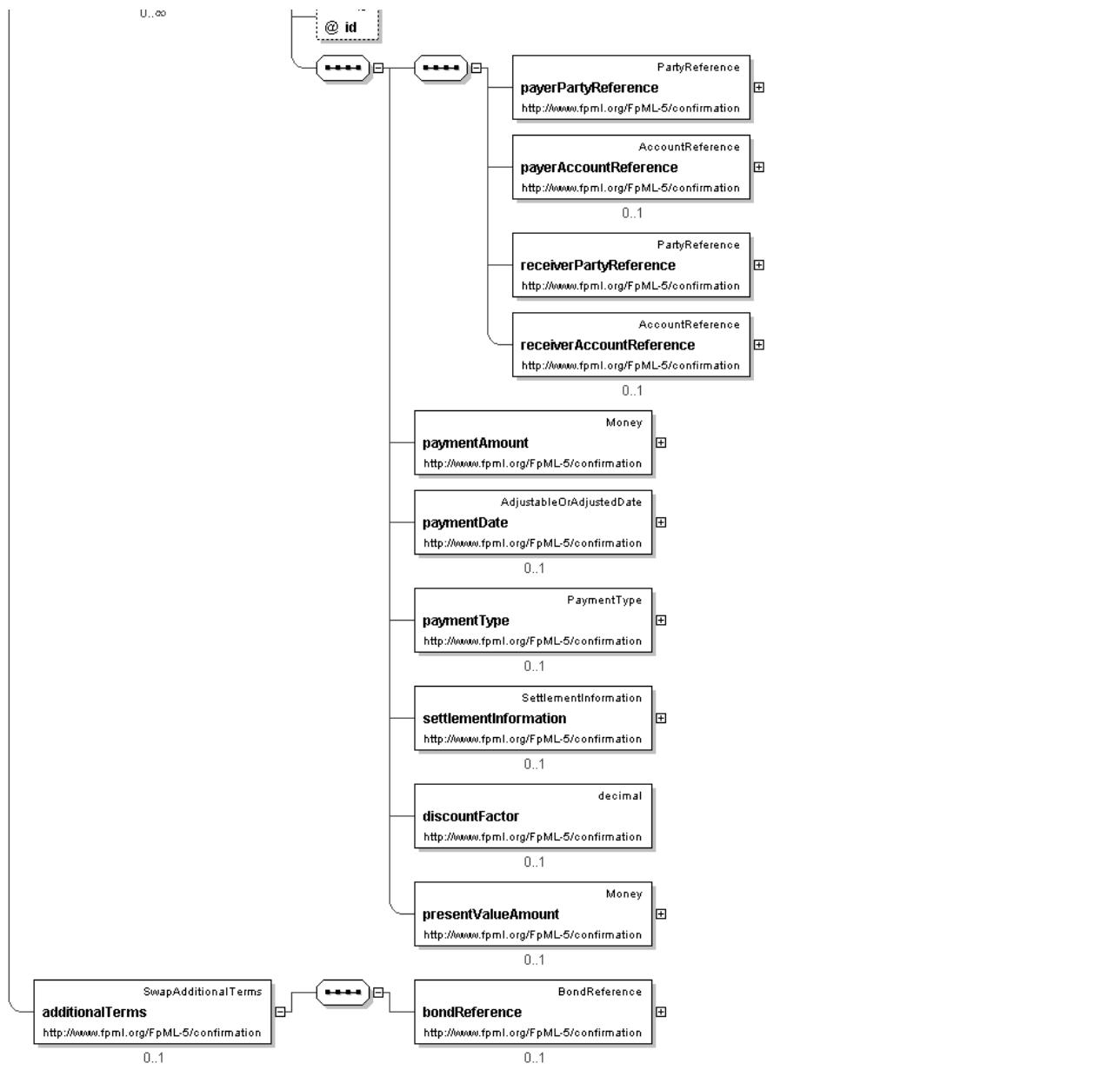
Name	swap
Used by (from the same schema document)	Complex Type <a href="#">Swaption</a>
Type	<a href="#">Swap</a>
Nillable	no
Abstract	no
Documentation	A swap product definition.

**Logical Diagram**







**XML Instance Representation**

```

<swap
  id=" xsd:ID [0..1]">
  <productType> ProductType </productType> [0..*]
  'A classification of the type of product. FpML defines a simple product categorization using
  a coding scheme.'
  <productId> ProductId </productId> [0..*]
  'A product 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.'
  
```

```

<swapStream> InterestRateStream </swapStream> [1..*]
'The swap streams.'

<earlyTerminationProvision> EarlyTerminationProvision </earlyTerminationProvision> [0..1]
'Parameters specifying provisions relating to the optional and mandatory early terminarion of
a swap transaction.'

<cancelableProvision> CancelableProvision </cancelableProvision> [0..1]
'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.'

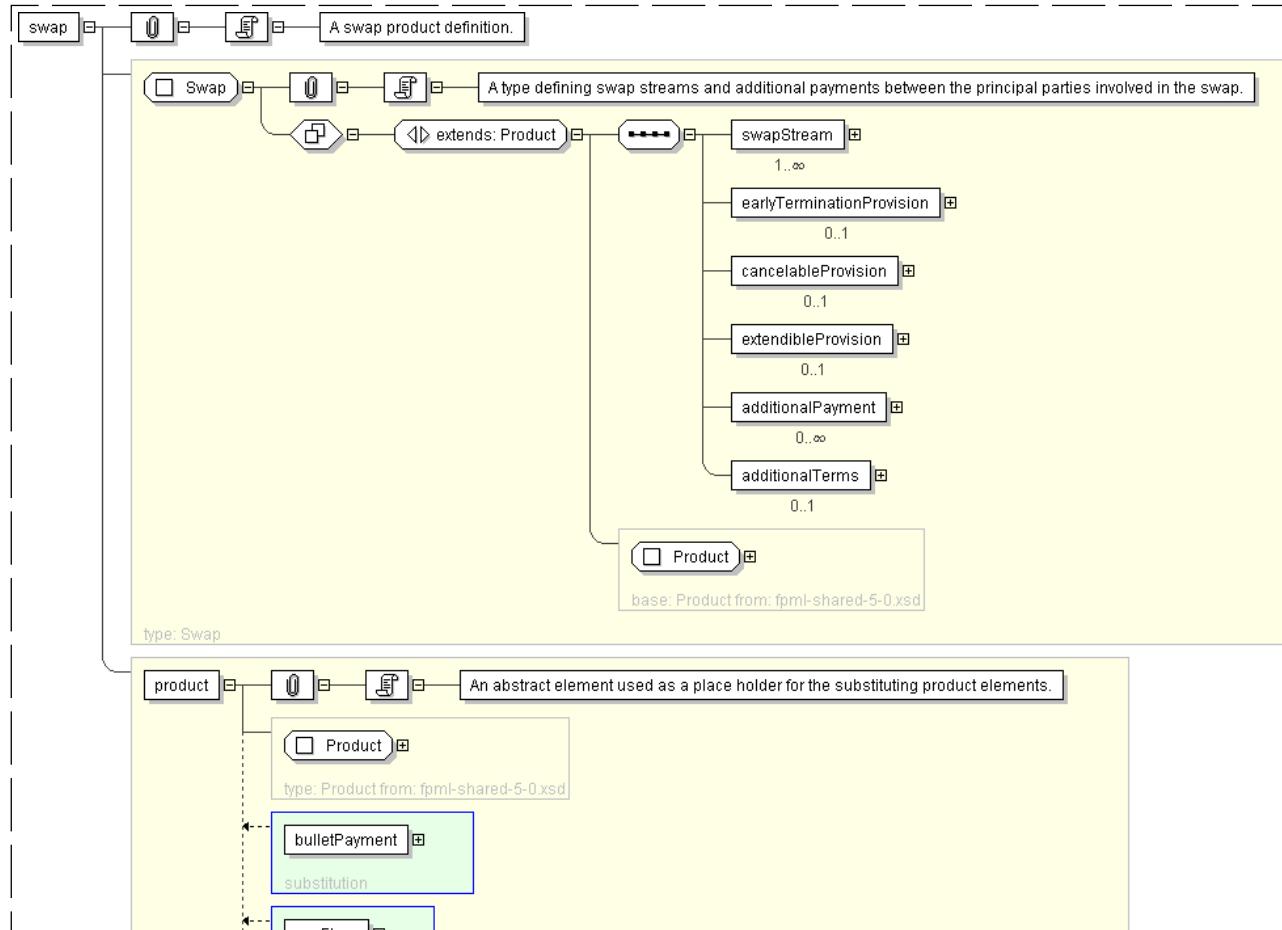
<extendibleProvision> ExtendibleProvision </extendibleProvision> [0..1]
'A provision that allows the specification of an embedded option with a swap giving the
buyer of the option the right to extend the swap, in whole or in part, to the
extended termination date.'

<additionalPayment> Payment </additionalPayment> [0..*]
'Additional payments between the principal parties.'

<additionalTerms> SwapAdditionalTerms </additionalTerms> [0..1]
'Contains any additional terms to the swap contract.'

</swap>

```

**Diagram**

**Schema Component Representation**

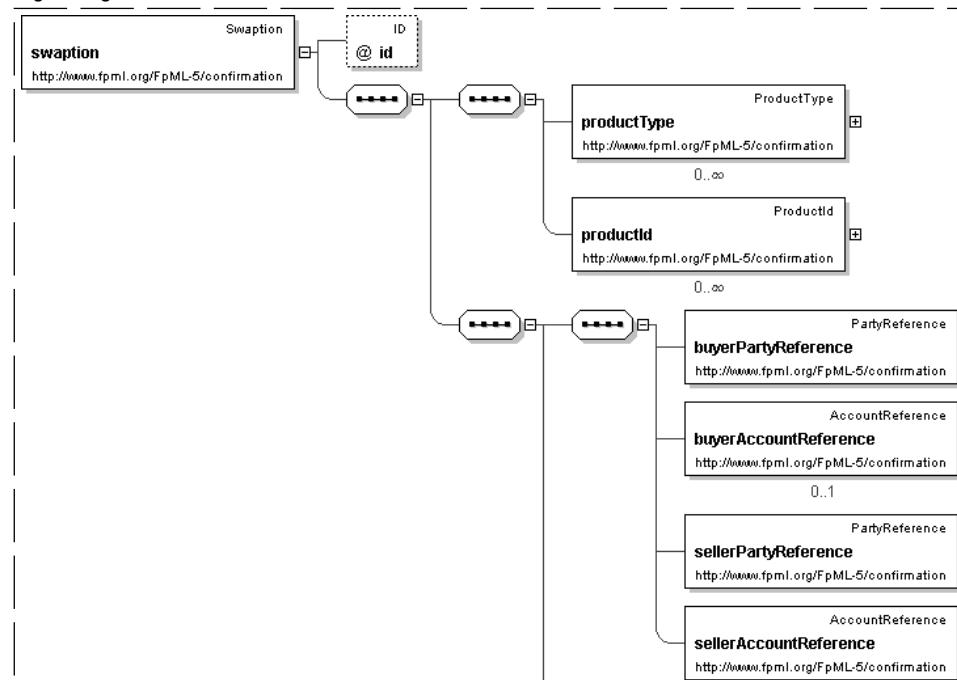
```
<xsd:element name="swaption" type="Swap" substitutionGroup="product"/>
```

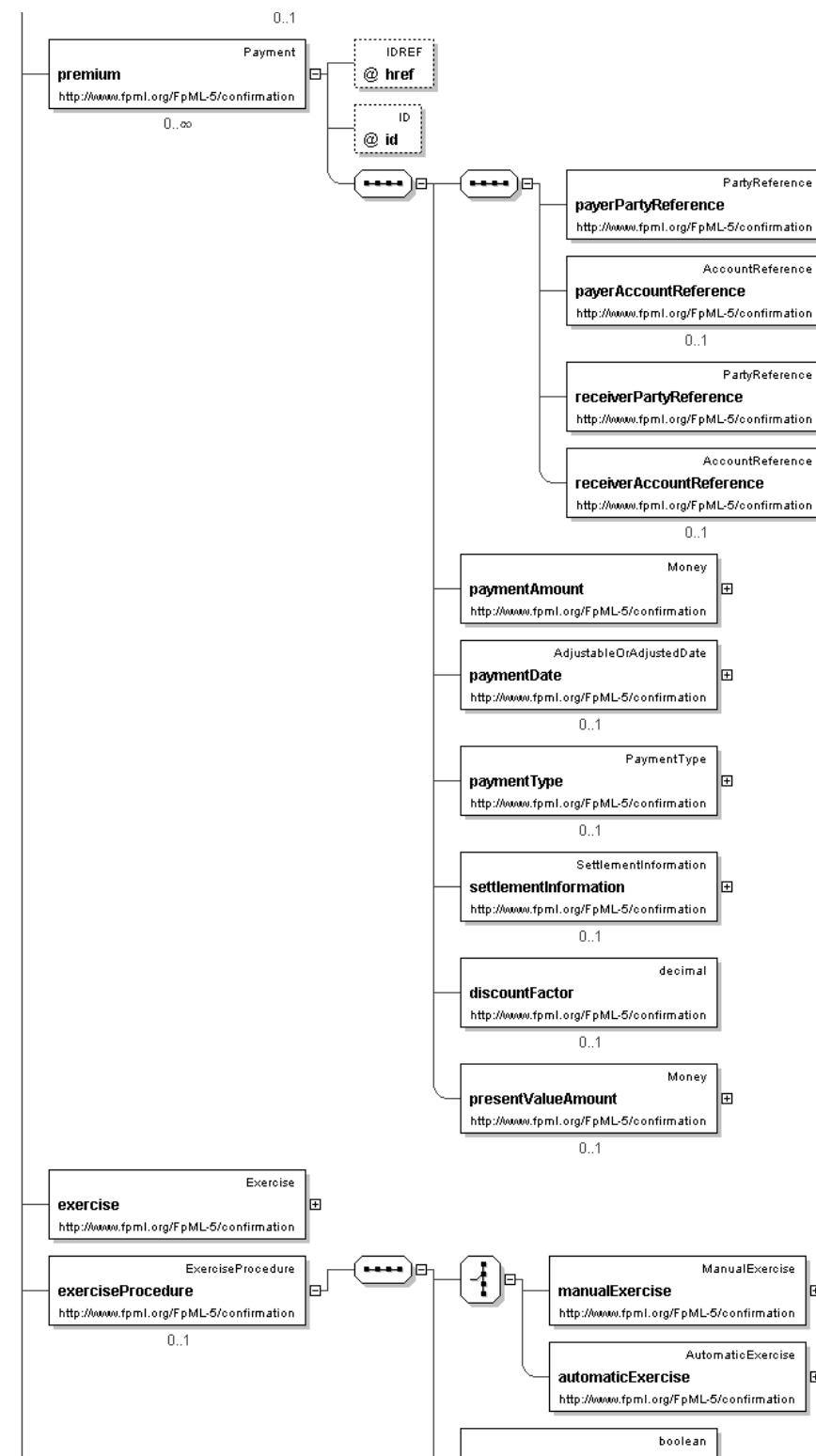
top

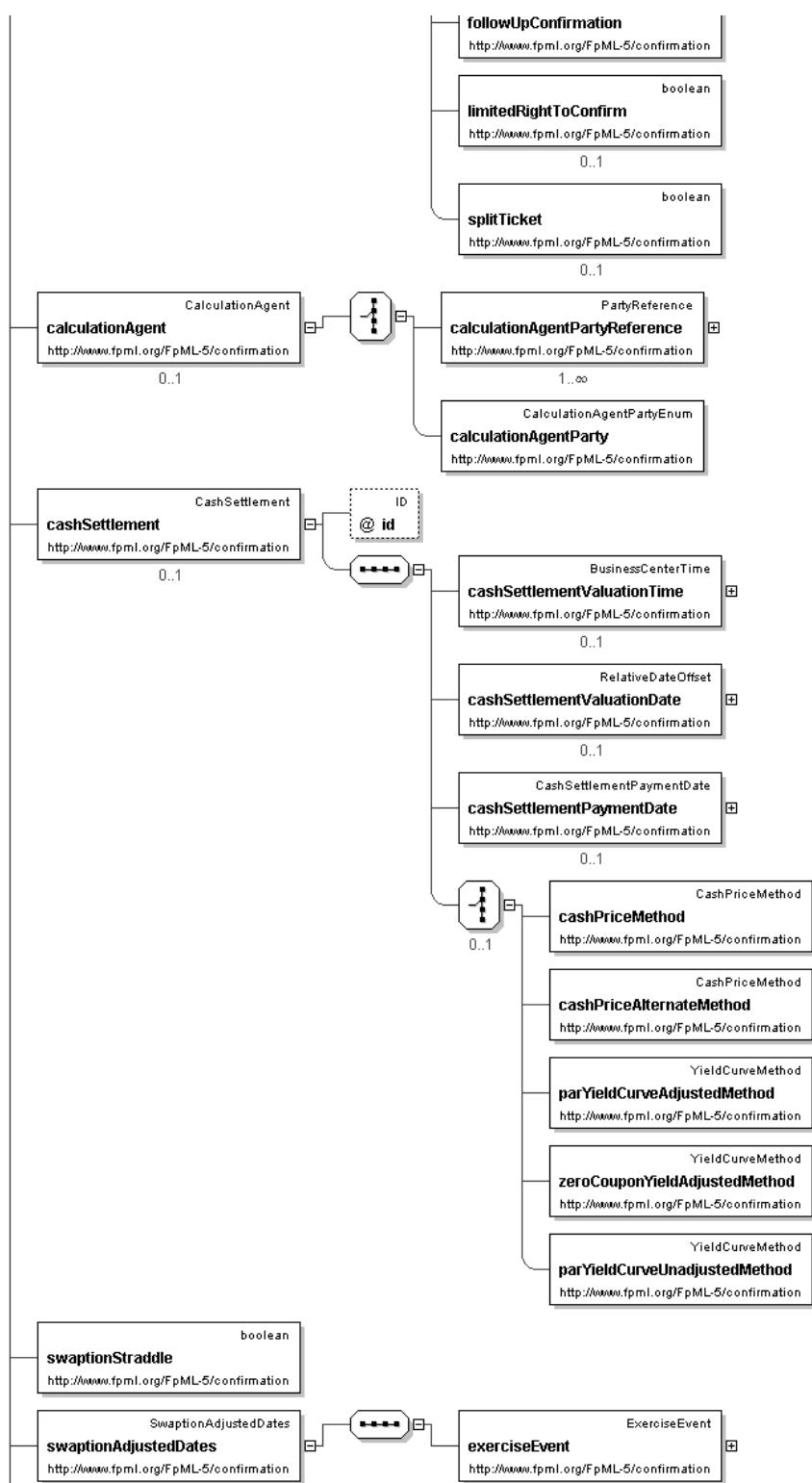
**Element: swaption**

- This element can be used wherever the following element is referenced:
  - product

Name	swaption
Type	Swap
Nullable	no
Abstract	no
Documentation	A swaption product definition.

**Logical Diagram**





**XML Instance Representation**

```

<swaption
  id=" xsd:ID [0..1]">
  <productType> ProductType </productType> [0..*]
  'A classification of the type of product. FpML defines a simple product categorization using
  a coding scheme.'

  <productId> ProductId </productId> [0..*]
  'A product 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.'

  <buyerPartyReference> PartyReference </buyerPartyReference> [1]
  'A reference to the party that buys this instrument, i.e. pays for this instrument and
  receives the rights defined by it. See 2000 ISDA definitions Article 11.1 (b). In the case
  of FRAs this is the fixed rate payer.'

  <buyerAccountReference> AccountReference </buyerAccountReference> [0..1]
  'A reference to the account that buys this instrument.'

  <sellerPartyReference> PartyReference </sellerPartyReference> [1]
  'A reference to the party that sells ("writes") this instrument, i.e. that grants the
  rights defined by this instrument and in return receives a payment for it. See 2000
  ISDA definitions Article 11.1 (a). In the case of FRAs this is the floating rate payer.'

  <sellerAccountReference> AccountReference </sellerAccountReference> [0..1]
  'A reference to the account that sells this instrument.'

  <premium> Payment </premium> [0..*]
  'The option premium amount payable by buyer to seller on the specified payment date.'

  <exercise> ... </exercise> [1]
  <exerciseProcedure> ExerciseProcedure </exerciseProcedure> [0..1]
  'A set of parameters defining procedures associated with the exercise.'

  <calculationAgent> CalculationAgent </calculationAgent> [0..1]
  'The ISDA Calculation Agent responsible for performing duties associated with an optional
  early termination.'

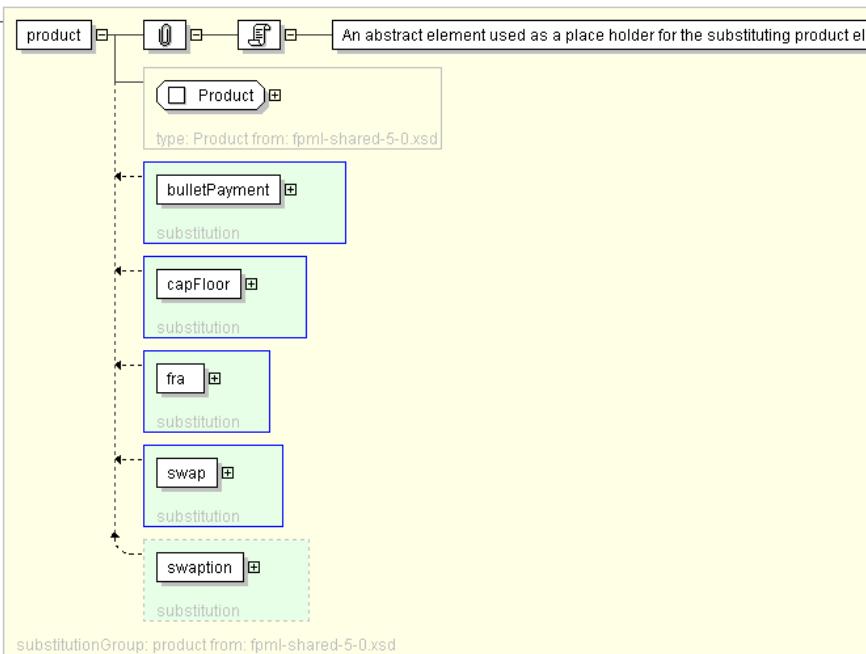
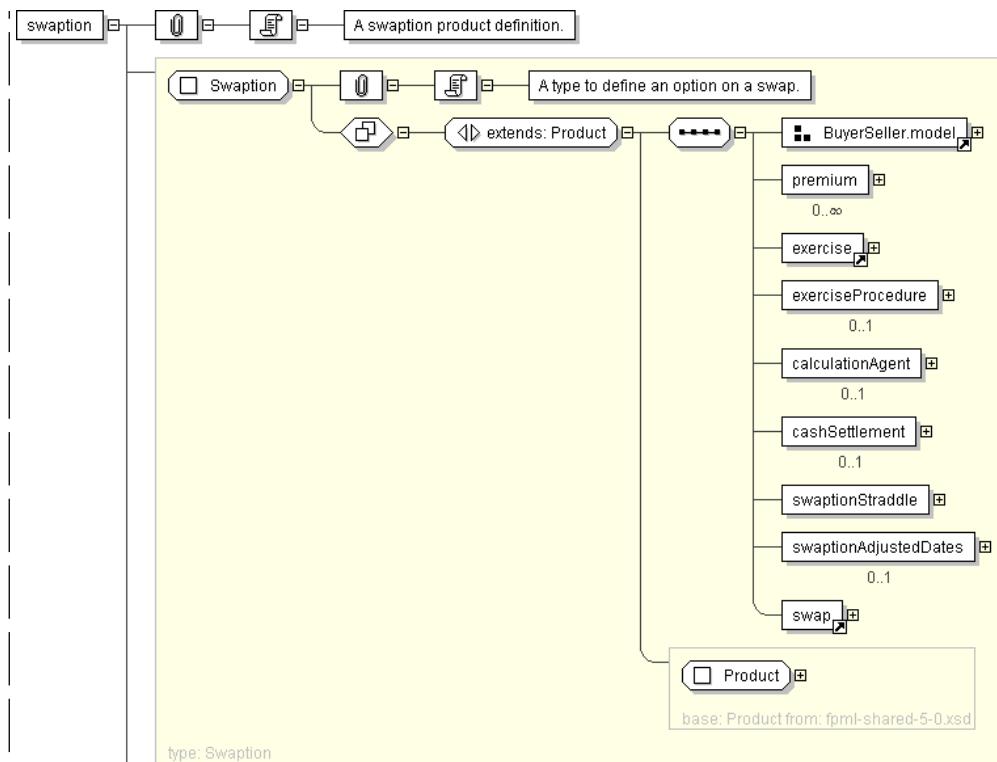
  <cashSettlement> CashSettlement </cashSettlement> [0..1]
  'If specified, this means that cash settlement is applicable to the transaction and defines
  the parameters associated with the cash settlement procedure. If not specified, then
  physical settlement is applicable.'

  <swaptionStraddle> xsd:boolean </swaptionStraddle> [1]
  'Whether the option is a swaption or a swaption straddle.'

  <swaptionAdjustedDates> SwaptionAdjustedDates </swaptionAdjustedDates> [0..1]
  'The adjusted dates associated with swaption exercise. These dates have been adjusted for
  any applicable business day convention.'

  <swap> ... </swap> [1]
</swaption>
  
```

**Diagram**

**Schema Component Representation**

```
<xsd:element name="swaption" type=" Swaption " substitutionGroup="product"/>
```

## Global Definitions

### Complex Type: BondReference

Super-types:	None
Sub-types:	None

Name	BondReference
Used by (from the same schema document)	Complex Type SwapAdditionalTerms
Abstract	no
Documentation	A type including a reference to a bond to support the representation of an asset swap or Condition Precedent Bond.

#### XML Instance Representation

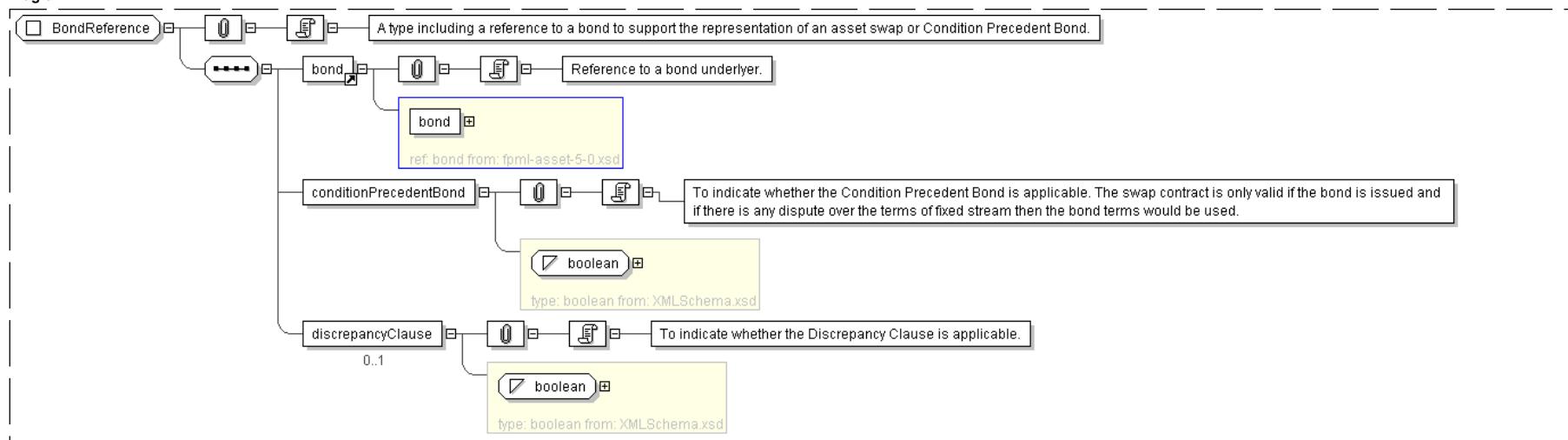
```
<...>
<bond> ... </bond> [1]
'Reference to a bond underlyer.'

<conditionPrecedentBond> xsd:boolean </conditionPrecedentBond> [1]
'To indicate whether the Condition Precedent Bond is applicable. The swap contract is
only valid if the bond is issued and if there is any dispute over the terms of fixed
stream then the bond terms would be used.'

<discrepancyClause> xsd:boolean </discrepancyClause> [0..1]
'To indicate whether the Discrepancy Clause is applicable.'

</...>
```

#### Diagram



#### Schema Component Representation

```
<xsd:complexType name="BondReference">
  <xsd:sequence>
    <xsd:element ref=" bond " />
    <xsd:element name="conditionPrecedentBond" type=" xsd:boolean " />
    <xsd:element name="discrepancyClause" type=" xsd:boolean " minOccurs="0 " />
  </xsd:sequence>
</xsd:complexType>
```

**Complex Type: BulletPayment**

<b>Super-types:</b>	<a href="#">Product</a> < <b>BulletPayment</b> (by extension)
<b>Sub-types:</b>	None
<b>Name</b>	BulletPayment
<b>Used by (from the same schema document)</b>	Element <a href="#">bulletPayment</a>
<b>Abstract</b>	no
<b>Documentation</b>	A product to represent a single cashflow.
<b>XML Instance Representation</b>	
<pre>&lt;...&gt;   id="xsd:ID [0..1]"&gt;     &lt;productType&gt; <a href="#">productType</a> &lt;/productType&gt; [0..*]       'A classification of the type of product. FpML defines a simple product categorization using a coding scheme.'</pre>	
<pre>&lt;productId&gt; <a href="#">ProductId</a> &lt;/productId&gt; [0..*]   'A product 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.'</pre>	
<pre>&lt;payment&gt; <a href="#">Payment</a> &lt;/payment&gt; [1]   'A known payment between two parties.'</pre>	
</...>	
<b>Diagram</b>	
<pre> classDiagram     class BulletPayment     class Product     BulletPayment "1" -- "0..1" Product : extends     BulletPayment "*" -- "1..1" payment   </pre> <p>The diagram illustrates the inheritance relationship between BulletPayment and Product. BulletPayment is shown as a class with a generalization arrow pointing to Product, labeled 'extends: Product'. It also features a directed association named 'payment' with multiplicity '*' at the source end and '1..1' at the target end.</p>	

**Schema Component Representation**

```

<xsd:complexType name="BulletPayment">
  <xsd:complexContent>
    <xsd:extension base=" Product ">
      <xsd:sequence>
        <xsd:element name="payment" type=" Payment " />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
  
```

[top](#)**Complex Type: Calculation**

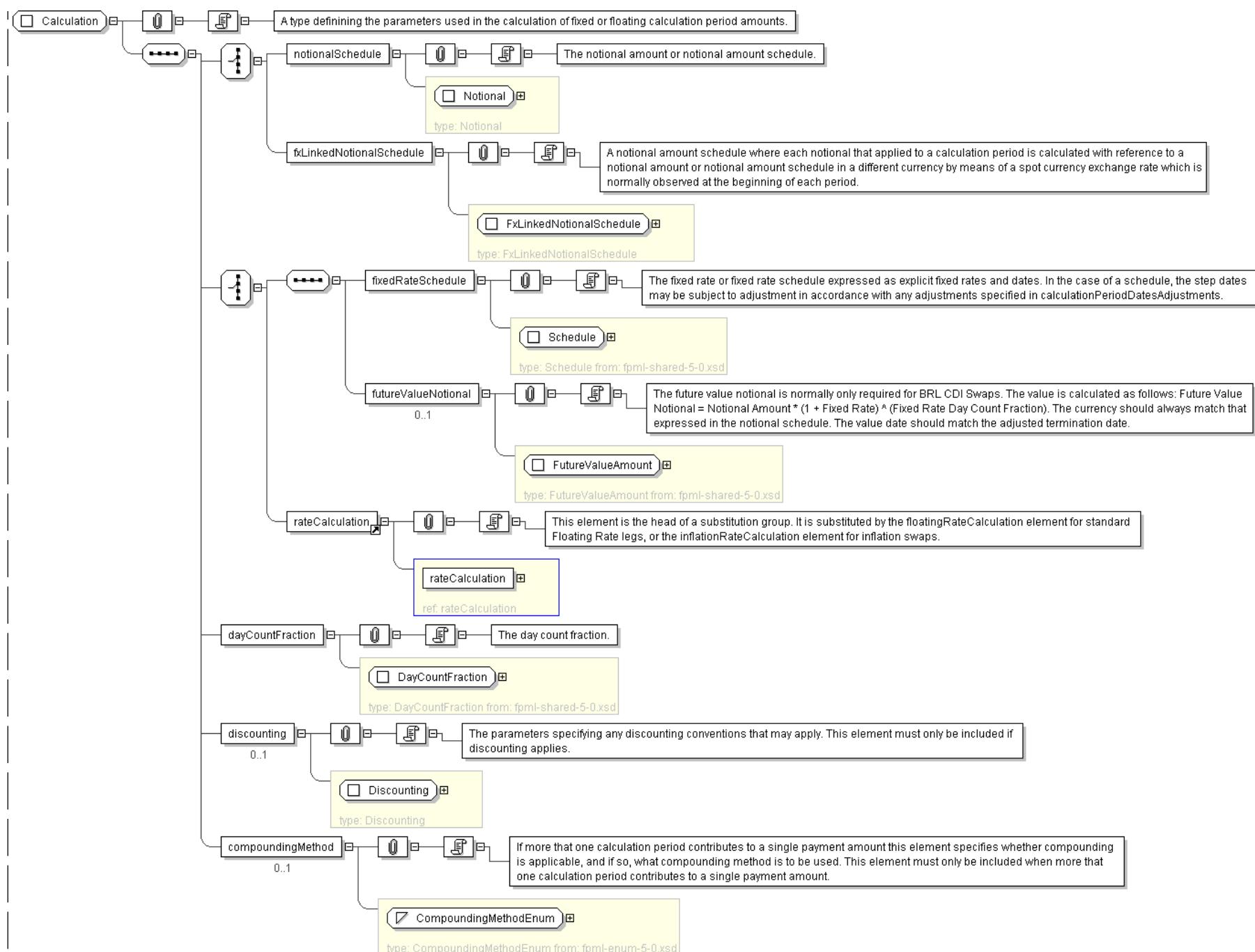
<b>Super-types:</b>	None
<b>Sub-types:</b>	None
<b>Name</b>	Calculation
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">CalculationPeriodAmount</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type defining the parameters used in the calculation of fixed or floating calculation period amounts.
<b>XML Instance Representation</b>	
<pre>&lt;...&gt;  </pre>	

```

| Start Choice [1]
|   <notionalSchedule> Notional </notionalSchedule> [1]
|     'The notional amount or notional amount schedule.'
|
|   <fxLinkedNotionalSchedule> FxLinkedNotionalSchedule </fxLinkedNotionalSchedule> [1]
|     'A notional amount schedule where each notional that applied 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.'
|
| End Choice
| Start Choice [1]
|   <fixedRateSchedule> Schedule </fixedRateSchedule> [1]
|     '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.'
|
|   <futureValueNotional> FutureValueAmount </futureValueNotional> [0..1]
|     'The future value notional is normally only required for BRL CDI Swaps. The value is
|      calculated as follows: Future Value Notional = Notional Amount * (1 + Fixed Rate) ^ (Fixed
|      Rate Day Count Fraction). The currency should always match that expressed in the
|      notional schedule. The value date should match the adjusted termination date.'
|
|   <rateCalculation> ... </rateCalculation> [1]
|     'This element is the head of a substitution group. It is substituted by
|      the floatingRateCalculation element for standard Floating Rate legs, or
|      the inflationRateCalculation element for inflation swaps.'
|
| End Choice
|   <dayCountFraction> DayCountFraction </dayCountFraction> [1]
|     'The day count fraction.'
|
|   <discounting> Discounting </discounting> [0..1]
|     'The parameters specifying any discounting conventions that may apply. This element must
|      only be included if discounting applies.'
|
|   <compoundingMethod> CompoundingMethodEnum </compoundingMethod> [0..1]
|     '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.'
|
| </...>

```

**Diagram**

**Schema Component Representation**

```

<xsd:complexType name="Calculation">
  <xsd:sequence>
    <xsd:choice>
      <xsd:element name="notionalSchedule" type=" Notional " />
      <xsd:element name="fxLinkedNotionalSchedule" type=" FxLinkedNotionalSchedule " />
    </xsd:choice>
  </xsd:sequence>
</xsd:complexType>

```

```

</xsd:choice>
<xsd:choice>
  <xsd:sequence>
    <xsd:element name="fixedRatesSchedule" type=" Schedule " />
    <xsd:element name="futureValueNotional" type=" FutureValueAmount " minOccurs="0" />
  </xsd:sequence>
  <xsd:element ref=" rateCalculation " />
</xsd:choice>
<xsd:element name="dayCountFraction" type=" DayCountFraction " />
<xsd:element name="discounting" type=" Discounting " minOccurs="0" />
<xsd:element name="compoundingMethod" type=" CompoundingMethodEnum " minOccurs="0" />
</xsd:sequence>
</xsd:complexType>

```

top

## Complex Type: CalculationPeriod

<b>Super-types:</b>	None
<b>Sub-types:</b>	None
<b>Name</b>	CalculationPeriod
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">PaymentCalculationPeriod</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type defining the parameters used in the calculation of a fixed or floating rate calculation period amount. This type forms part of cashflows representation of a swap stream.

### XML Instance Representation

```

<...
  id=" xsd:ID [0..1]">
    <unadjustedStartDate> xsd:date </unadjustedStartDate> [0..1]
    <unadjustedEndDate> xsd:date </unadjustedEndDate> [0..1]
    <adjustedStartDate> xsd:date </adjustedStartDate> [0..1]
    'The calculation period start date, adjusted according to any relevant business day convention.'

    <adjustedEndDate> xsd:date </adjustedEndDate> [0..1]
    'The calculation period end date, adjusted according to any relevant business day convention.'

    <calculationPeriodNumberOfDays> xsd:positiveInteger </calculationPeriodNumberOfDays> [0..1]
    'The number of days from the adjusted effective / start date to the adjusted termination / end date calculated in accordance with the applicable day count fraction.'

Start Choice [1]
  <notionalAmount> xsd:decimal </notionalAmount> [1]
  'The amount that a cashflow will accrue interest on.'

  <fxLinkedNotionalAmount> FxLinkedNotionalAmount </fxLinkedNotionalAmount> [1]
  'The amount that a cashflow will accrue interest on. This is the calculated amount of the fx linked - ie the other currency notional amount multiplied by the appropriate fx spot rate.'

End Choice
Start Choice [1]
  <floatingRateDefinition> FloatingRateDefinition </floatingRateDefinition> [1]
  'The floating rate reset information for the calculation period.'

  <fixedRate> xsd:decimal </fixedRate> [1]
  'The calculation period fixed rate. A per annum rate, expressed as a decimal. A fixed rate of 5% would be represented as 0.05.'

End Choice
  <dayCountYearFraction> xsd:decimal </dayCountYearFraction> [0..1]
  'The year fraction value of the calculation period, result of applying the ISDA rules for day count fraction defined in the ISDA Annex.'

  <forecastAmount> Money </forecastAmount> [0..1]

```

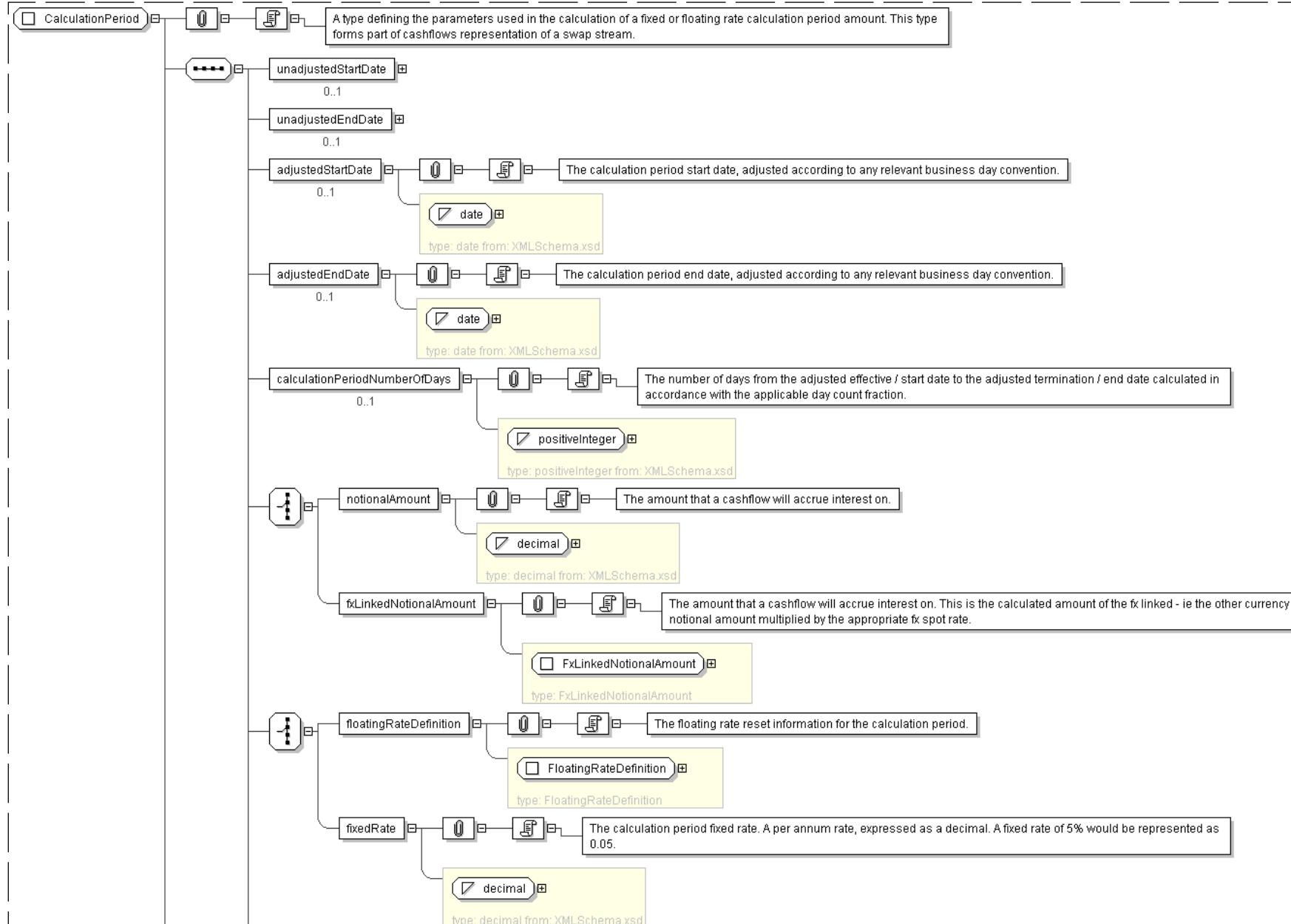
'The amount representing the forecast of the accrued value of the calculation period.'

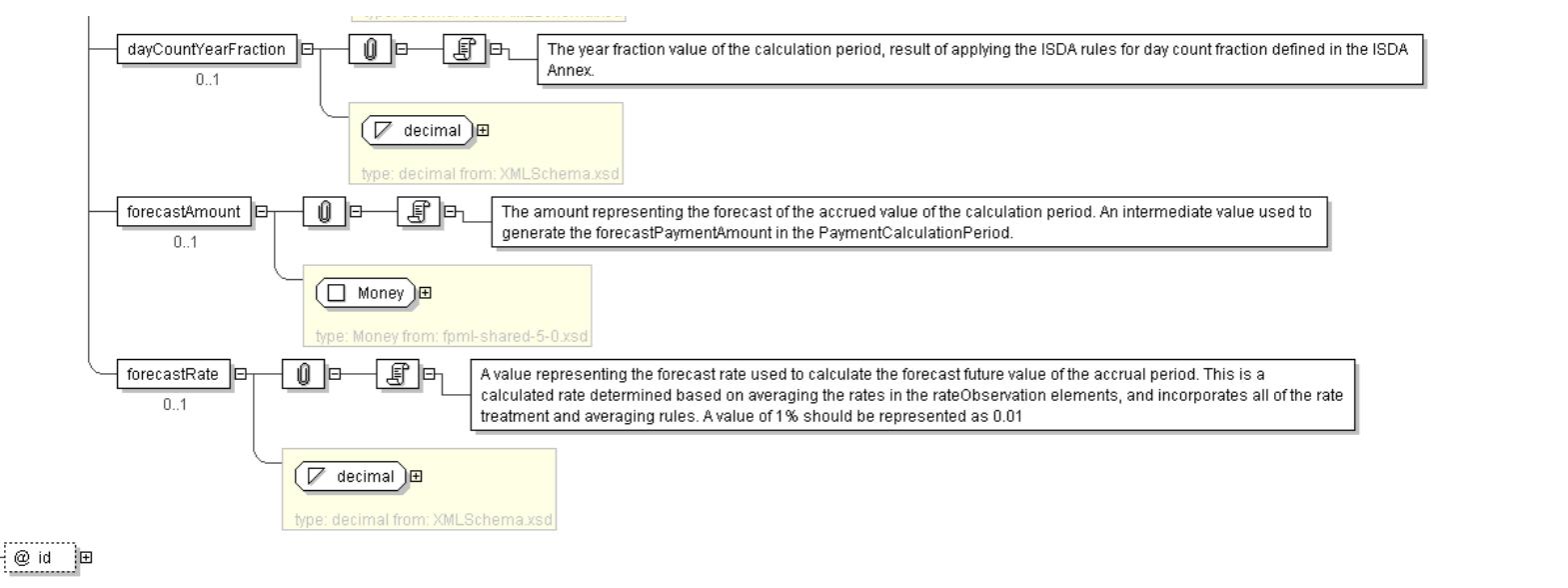
An intermediate value used to generate the forecastPaymentAmount in the PaymentCalculationPeriod.'

```
</forecastRate> xsd:decimal </forecastRate> [0..1]
```

'A value representing the forecast rate used to calculate the forecast future value of the accrual period. This is a calculated rate determined based on averaging the rates in the rateObservation elements, and incorporates all of the rate treatment and averaging rules. A value of 1% should be represented as 0.01'

&lt;/...&gt;

**Diagram**

**Schema Component Representation**

```

<xsd:complexType name="CalculationPeriod">
  <xsd:sequence>
    <xsd:element name="unadjustedStartDate" type="xsd:date" minOccurs="0"/>
    <xsd:element name="unadjustedEndDate" type="xsd:date" minOccurs="0"/>
    <xsd:element name="adjustedStartDate" type="xsd:date" minOccurs="0"/>
    <xsd:element name="adjustedEndDate" type="xsd:date" minOccurs="0"/>
    <xsd:element name="calculationPeriodNumberOfDays" type="xsd:positiveInteger" minOccurs="0"/>
    <xsd:choice>
      <xsd:element name="notionalAmount" type="xsd:decimal"/>
      <xsd:element name="fxLinkedNotionalAmount" type="FxLinkedNotionalAmount"/>
    </xsd:choice>
    <xsd:choice>
      <xsd:element name="floatingRateDefinition" type="FloatingRateDefinition"/>
      <xsd:element name="fixedRate" type="xsd:decimal"/>
    </xsd:choice>
    <xsd:element name="dayCountYearFraction" type="xsd:decimal" minOccurs="0"/>
    <xsd:element name="forecastAmount" type="Money" minOccurs="0"/>
    <xsd:element name="forecastRate" type="xsd:decimal" minOccurs="0"/>
  </xsd:sequence>
  <xsd:attribute name="id" type="xsd:ID"/>
</xsd:complexType>

```

top

**Complex Type: CalculationPeriodAmount**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None

<b>Name</b>	CalculationPeriodAmount
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">InterestRateStream</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type 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.

**XML Instance Representation**

```

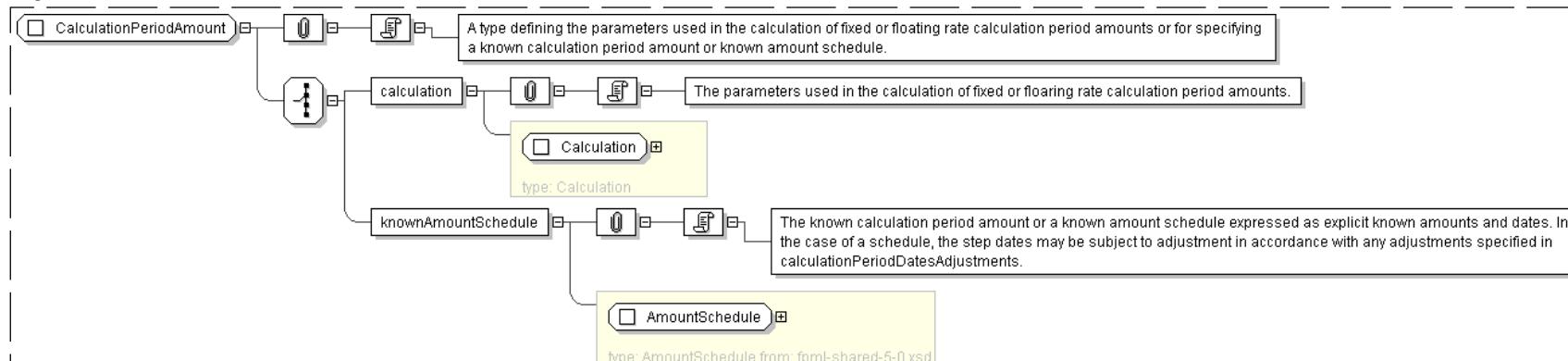
<....>
Start Choice [1]

```

```
<calculation> Calculation </calculation> [1]
'The parameters used in the calculation of fixed or floating rate calculation period amounts.'

<knownAmountSchedule> AmountSchedule </knownAmountSchedule> [1]
'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.'

End Choice
</...>
```

**Diagram****Schema Component Representation**

```
<xsd:complexType name="CalculationPeriodAmount">
  <xsd:choice>
    <xsd:element name="calculation" type="Calculation" />
    <xsd:element name="knownAmountSchedule" type="AmountSchedule" />
  </xsd:choice>
</xsd:complexType>
```

top

**Complex Type: CalculationPeriodDates**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None
<b>Name</b>	CalculationPeriodDates
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">InterestRateStream</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type defining the parameters used to generate the calculation period 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.

**XML Instance Representation**

```
<...
  id=" xsd:ID [1]">
Start Choice [1]
  <effectiveDate> AdjustableDate </effectiveDate> [1]
    'The first day of the term of the trade. This day may be subject to adjustment in accordance with a business day convention.'
  <relativeEffectiveDate> AdjustedRelativeDateOffset </relativeEffectiveDate> [1]
    'Defines the effective date.'
```

```

End Choice
Start Choice [1]
  <terminationDate> AdjustableDate </terminationDate> [1]
  'The last day of the term of the trade. This day may be subject to adjustment in
  accordance with a business day convention.'

  <relativeTerminationDate> RelativeDateOffset </relativeTerminationDate> [1]
  'The term/maturity of the swap, express as a tenor (typically in years).'

End Choice
<calculationPeriodDatesAdjustments> BusinessDayAdjustments </
calculationPeriodDatesAdjustments> [1]
'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> AdjustableDate </firstPeriodStartDate> [0..1]
'The start date of the 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 date may be subject
to adjustment in accordance with a business day convention.'

<firstRegularPeriodStartDate> xsd:date </firstRegularPeriodStartDate> [0..1]
'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.'

<firstCompoundingPeriodEndDate> xsd:date </firstCompoundingPeriodEndDate> [0..1]
'The end date of the initial compounding period when compounding is applicable. It must only
be specified when the compoundingMethod element is present and not equal to a value of
None. This date may be subject to adjustment in accordance with any adjustments specified
in calculationPeriodDatesAdjustments.'

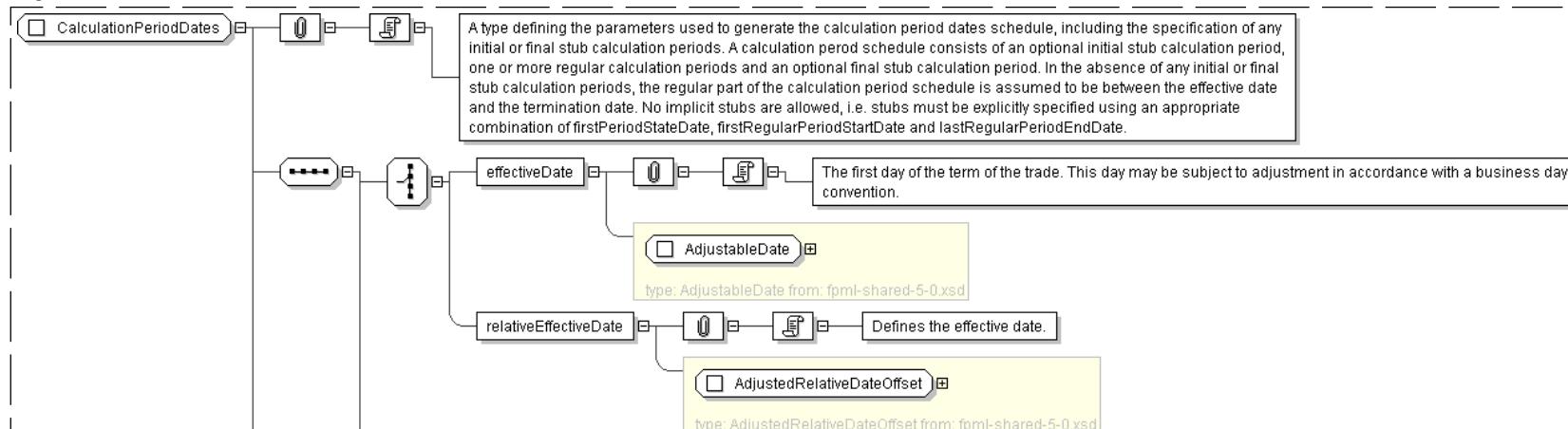
<lastRegularPeriodEndDate> xsd:date </lastRegularPeriodEndDate> [0..1]
'The end date of the regular part of the calculation period schedule. It must only be
specified if there is a final stub calculation period. This day may be subject to adjustment
in accordance with any adjustments specified in calculationPeriodDatesAdjustments.'

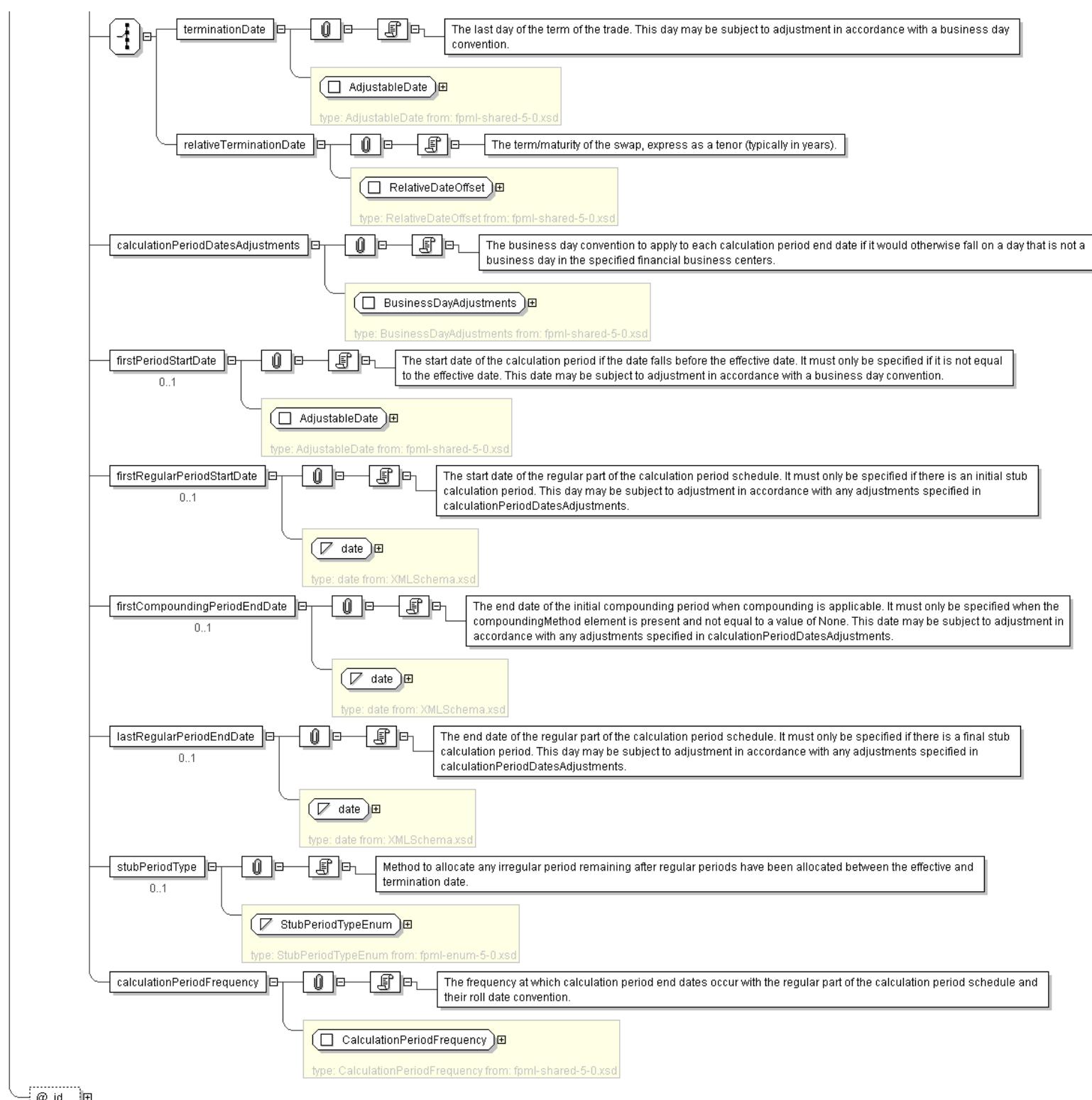
<stubPeriodType> StubPeriodTypeEnum </stubPeriodType> [0..1]
'Method to allocate any irregular period remaining after regular periods have been
allocated between the effective and termination date.'

<calculationPeriodFrequency> CalculationPeriodFrequency </calculationPeriodFrequency> [1]
'The frequency at which calculation period end dates occur with the regular part of
the calculation period schedule and their roll date convention.'

<...>

```

**Diagram**



## Schema Component Representation

```
<xsd:complexType name="CalculationPeriodDates">
  <xsd:sequence>
    <xsd:choice>
      <xsd:element name="effectiveDate" type=" AdjustableDate " />
      <xsd:element name="relativeEffectiveDate" type=" AdjustedRelativeDateOffset " />
    </xsd:choice>
    <xsd:choice>
      <xsd:element name="terminationDate" type=" AdjustableDate " />
      <xsd:element name="relativeTerminationDate" type=" RelativeDateOffset " />
    </xsd:choice>
    <xsd:element name="calculationPeriodDatesAdjustments" type=" BusinessDayAdjustments " />
    <xsd:element name="firstPeriodStartDate" type=" AdjustableDate " minOccurs="0" />
    <xsd:element name="firstRegularPeriodStartDate" type=" xsd:date " minOccurs="0" />
    <xsd:element name="firstCompoundingPeriodEndDate" type=" xsd:date " minOccurs="0" />
    <xsd:element name="lastRegularPeriodEndDate" type=" xsd:date " minOccurs="0" />
    <xsd:element name="stubPeriodType" type=" StubPeriodTypeEnum " minOccurs="0" />
    <xsd:element name="calculationPeriodFrequency" type=" CalculationPeriodFrequency " />
  </xsd:sequence>
  <xsd:attribute name="id" type=" xsd:ID " use="required" />
</xsd:complexType>
```

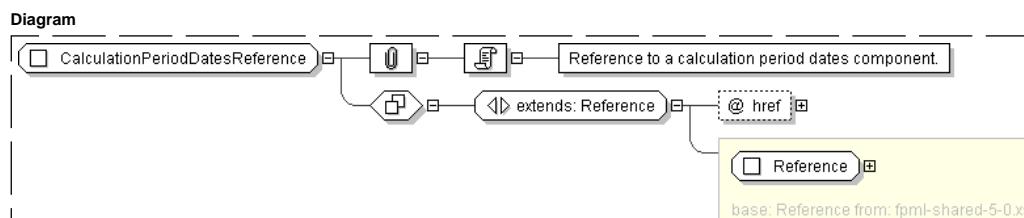
top

**Complex Type:** CalculationPeriodDatesReference

Super-types:	<a href="#">Reference</a> < <b>CalculationPeriodDatesReference</b> (by extension)
Sub-types:	None
<b>Name</b>	CalculationPeriodDatesReference
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">DateRelativeToCalculationPeriodDates</a> , Complex Type <a href="#">NotionalStepRule</a> , Complex Type <a href="#">PaymentDates</a> , Complex Type <a href="#">ResetDates</a> , Complex Type <a href="#">StubCalculationPeriodAmount</a>
<b>Abstract</b>	no
<b>Documentation</b>	Reference to a calculation period dates component.

## XML Instance Representation

```
<...>  
| href="#" xsd:IDREF [1]"/>
```



## Schema Component Representation

```
<xsd:complexType name="CalculationPeriodDatesReference">
  <xsd:complexContent>
    <xsd:extension base="Reference">
      <xsd:attribute name="href" type="xsd:IDREF" use="required">
        reference="CalculationPeriodDates"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

**Complex Type: CancelableProvision**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None
<b>Name</b>	CancelableProvision
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">Swap</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type defining the right of a party to cancel a swap transaction on the specified exercise dates. The provision is for 'walkaway' cancellation (i.e. the fair value of the swap is not paid). A fee payable on exercise can be specified.

**XML Instance Representation**

```

<...>
<buyerPartyReference> PartyReference </buyerPartyReference> [1]
'A reference to the party that buys this instrument, i.e. pays for this instrument and
receives the rights defined by it. See 2000 ISDA definitions Article 11.1 (b). In the case
of FRAs this is the fixed rate payer.'

<buyerAccountReference> AccountReference </buyerAccountReference> [0..1]
'A reference to the account that buys this instrument.'

<sellerPartyReference> PartyReference </sellerPartyReference> [1]
'A reference to the party that sells ("writes") this instrument, i.e. that grants the
rights defined by this instrument and in return receives a payment for it. See 2000
ISDA definitions Article 11.1 (a). In the case of FRAs this is the floating rate payer.'

<sellerAccountReference> AccountReference </sellerAccountReference> [0..1]
'A reference to the account that sells this instrument.'

<exercise> ... </exercise> [1]
<exerciseNotice> ExerciseNotice </exerciseNotice> [0..1]
'Definition of the party to whom notice of exercise should be given.'

<followUpConfirmation> xsd:boolean </followUpConfirmation> [1]
'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> CancelableProvisionAdjustedDates
</cancelableProvisionAdjustedDates> [0..1]
'The adjusted dates associated with a cancelable provision. These dates have been adjusted
for any applicable business day convention.'

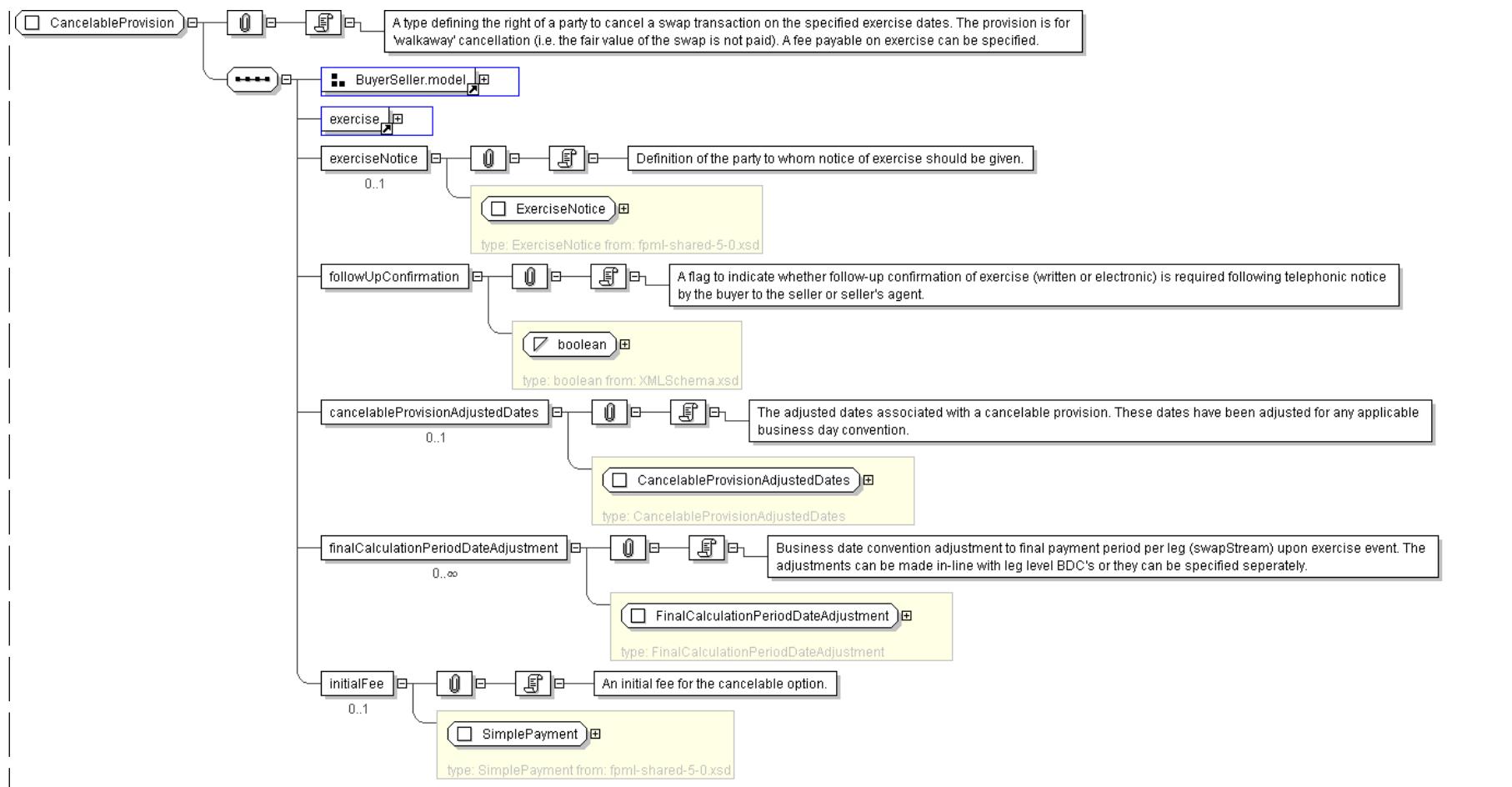
<finalCalculationPeriodDateAdjustment> FinalCalculationPeriodDateAdjustment
</finalCalculationPeriodDateAdjustment> [0..*]
'Business date convention adjustment to final payment period per leg (swapStream) upon
exercise event. The adjustments can be made in-line with leg level BDC's or they can
be specified separately.'

<initialFee> SimplePayment </initialFee> [0..1]
'An initial fee for the cancelable option.'

</...>

```

**Diagram**

**Schema Component Representation**

```

<xsd:complexType name="CancelableProvision">
  <xsd:sequence>
    <xsd:group ref=" BuyerSeller.model " />
    <xsd:element ref=" exercise ">
    <xsd:element name="exerciseNotice" type=" ExerciseNotice " minOccurs="0 " />
    <xsd:element name="followUpConfirmation" type=" xsd:boolean " />
    <xsd:element name="cancelableProvisionAdjustedDates" type=" CancelableProvisionAdjustedDates "
      " minOccurs="0 " />
    <xsd:element name="finalCalculationPeriodDateAdjustment"
      type=" FinalCalculationPeriodDateAdjustment " minOccurs="0" maxOccurs="unbounded" />
    <xsd:element name="initialFee" type=" SimplePayment " minOccurs="0 " />
  </xsd:sequence>
</xsd:complexType>

```

top

**Complex Type: CancelableProvisionAdjustedDates**

Super-types:	None
Sub-types:	None

Name	CancelableProvisionAdjustedDates
Used by (from the same schema document)	Complex Type <a href="#">CancelableProvision</a>

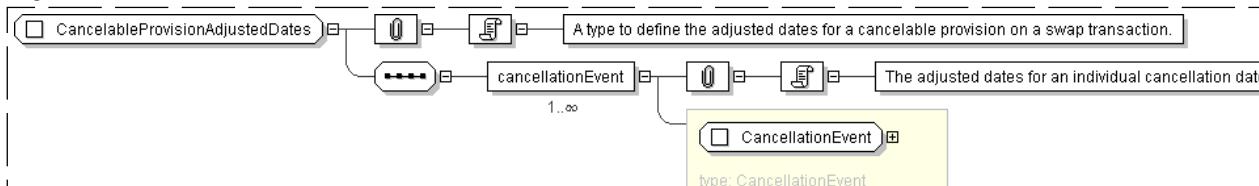
<b>Abstract</b>	no
<b>Documentation</b>	A type to define the adjusted dates for a cancelable provision on a swap transaction.

**XML Instance Representation**

```
<....>
<cancellationEvent> CancellationEvent </cancellationEvent> [1..*]
```

'The adjusted dates for an individual cancellation date.'

</....>

**Diagram****Schema Component Representation**

```
<xsd:complexType name="CancellableProvisionAdjustedDates">
  <xsd:sequence>
    <xsd:element name="cancellationEvent" type="#CancellationEvent" maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>
```

[top](#)

**Complex Type: CancellationEvent**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None

<b>Name</b>	CancellationEvent
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">CancellableProvisionAdjustedDates</a>
<b>Abstract</b>	no
<b>Documentation</b>	The adjusted dates for a specific cancellation date, including the adjusted exercise date and adjusted termination date.

**XML Instance Representation**

```
<...
id="#ID [0..1]">
<adjustedExerciseDate> xsd:date </adjustedExerciseDate> [1]
```

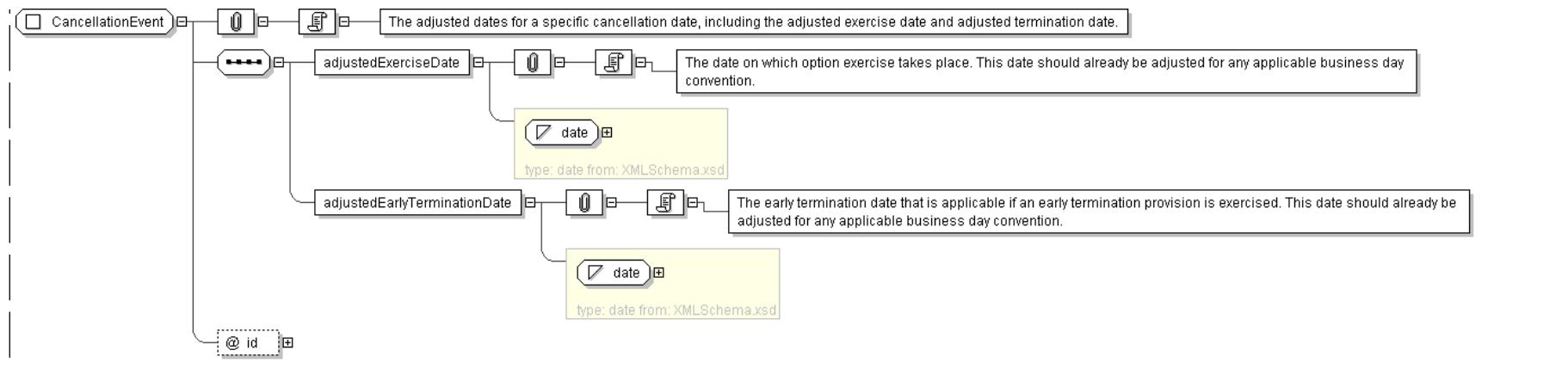
'The date on which option exercise takes place. This date should already be adjusted for any applicable business day convention.'

```
<adjustedEarlyTerminationDate> xsd:date </adjustedEarlyTerminationDate> [1]
```

'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.'

</....>

**Diagram**



## Schema Component Representation

```
<xsd:complexType name="CancellationEvent">
  <xsd:sequence>
    <xsd:element name="adjustedExerciseDate" type="xsd:date" />
    <xsd:element name="adjustedEarlyTerminationDate" type="xsd:date" />
  </xsd:sequence>
  <xsd:attribute name="id" type="xsd:ID" />
</xsd:complexType>
```

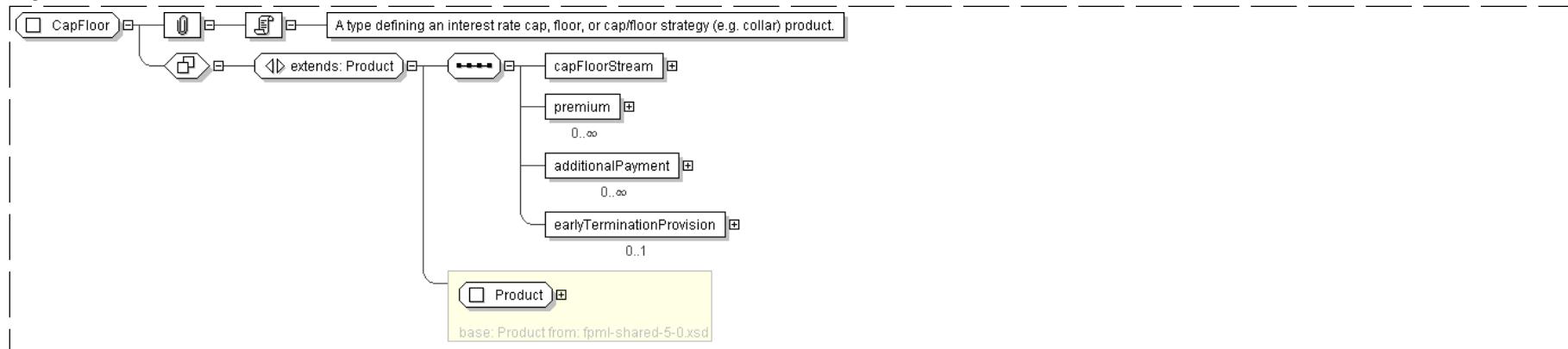
top

## Complex Type: CapFloor

<b>Super-types:</b>	<a href="#">Product</a> < <b>CapFloor</b> (by extension)
<b>Sub-types:</b>	None
<b>Name</b>	CapFloor
<b>Used by (from the same schema document)</b>	Element <a href="#">capFloor</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type defining an interest rate cap, floor, or cap/floor strategy (e.g. collar) product.

## XML Instance Representation

```
<...  
id=" xsd:ID [0..1]">  
<productType> ProductType </productType> [0..*]  
'A classification of the type of product. FpML defines a simple product categorization using  
a coding scheme.'  
  
<productId> ProductId </productId> [0..*]  
'A product 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.'  
  
<capFloorStream> InterestRateStream </capFloorStream> [1]  
<premium> Payment </premium> [0..*]  
'The option premium amount payable by buyer to seller on the specified payment date.'  
  
<additionalPayment> Payment </additionalPayment> [0..*]  
'Additional payments between the principal parties.'  
  
<earlyTerminationProvision> EarlyTerminationProvision </earlyTerminationProvision> [0..1]  
'Parameters specifying provisions relating to the optional and mandatory early terminarion o  
a CapFloor transaction.'  
  
</...>
```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="CapFloor">
  <xsd:complexContent>
    <xsd:extension base=" Product ">
      <xsd:sequence>
        <xsd:element name="capFloorStream" type=" InterestRateStream "/>
        <xsd:element name="premium" type=" Payment " minOccurs="0" maxOccurs="unbounded"/>
        <xsd:element name="additionalPayment" type=" Payment " minOccurs="0" maxOccurs="unbounded"/>
        <xsd:element name="earlyTerminationProvision" type=" EarlyTerminationProvision " minOccurs="0"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
  
```

[top](#)**Complex Type: CashPriceMethod**

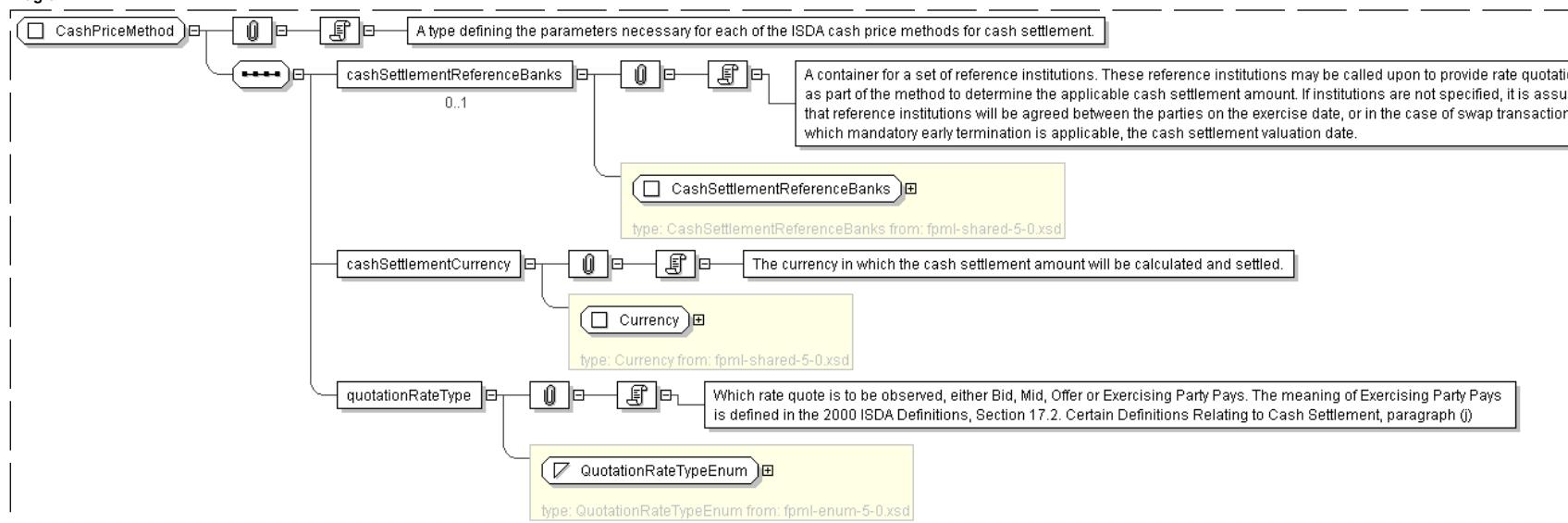
<b>Super-types:</b>	None
<b>Sub-types:</b>	None
<b>Name</b>	CashPriceMethod
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">CashSettlement</a> , Complex Type <a href="#">CashSettlement</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type defining the parameters necessary for each of the ISDA cash price methods for cash settlement.

**XML Instance Representation**

```

<....>
<cashSettlementReferenceBanks> CashSettlementReferenceBanks </
cashSettlementReferenceBanks> [0..1]
'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. If institutions are not specified, it is assumed that
reference institutions will be agreed between the parties on the exercise date, or in the
case of swap transaction to which mandatory early termination is applicable, the
cash settlement valuation date.'
<cashSettlementCurrency> Currency </cashSettlementCurrency> [1]
'The currency in which the cash settlement amount will be calculated and settled.'
<quotationRateType> QuotationRateTypeEnum </quotationRateType> [1]
'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)'
  
```

&lt;/...&gt;

**Diagram****Schema Component Representation**

```

<xsd:complexType name="CashPriceMethod">
  <xsd:sequence>
    <xsd:element name="cashSettlementReferenceBanks" type=" CashSettlementReferenceBanks
      " minOccurs="0" />
    <xsd:element name="cashSettlementCurrency" type=" Currency " />
    <xsd:element name="quotationRateType" type=" QuotationRateTypeEnum " />
  </xsd:sequence>
</xsd:complexType>
  
```

top

**Complex Type: CashSettlement**

Super-types:	None
Sub-types:	None
Name	CashSettlement
Used by (from the same schema document)	Complex Type <a href="#">MandatoryEarlyTermination</a> , Complex Type <a href="#">OptionalEarlyTermination</a> , Complex Type <a href="#">Swaption</a>
Abstract	no
Documentation	A type to define the cash settlement terms for a product where cash settlement is applicable.

**XML Instance Representation**

```

<...
  id=" xsd:ID [0..1]">
  <cashSettlementValuationTime> BusinessCenterTime </cashSettlementValuationTime> [0..1]
  'The time of 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> RelativeDateOffset </cashSettlementValuationDate> [0..1]
  '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> CashSettlementPaymentDate </cashSettlementPaymentDate> [0..1]
  '
  
```

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

Start Choice [0..1]

<cashPriceMethod> CashPriceMethod </cashPriceMethod> [1]

'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).'

<cashPriceAlternateMethod> CashPriceMethod </cashPriceAlternateMethod> [1]

'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).'

<parYieldCurveAdjustedMethod> YieldCurveMethod </parYieldCurveAdjustedMethod> [1]

'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).'

<zeroCouponYieldAdjustedMethod> YieldCurveMethod </zeroCouponYieldAdjustedMethod> [1]

'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).'

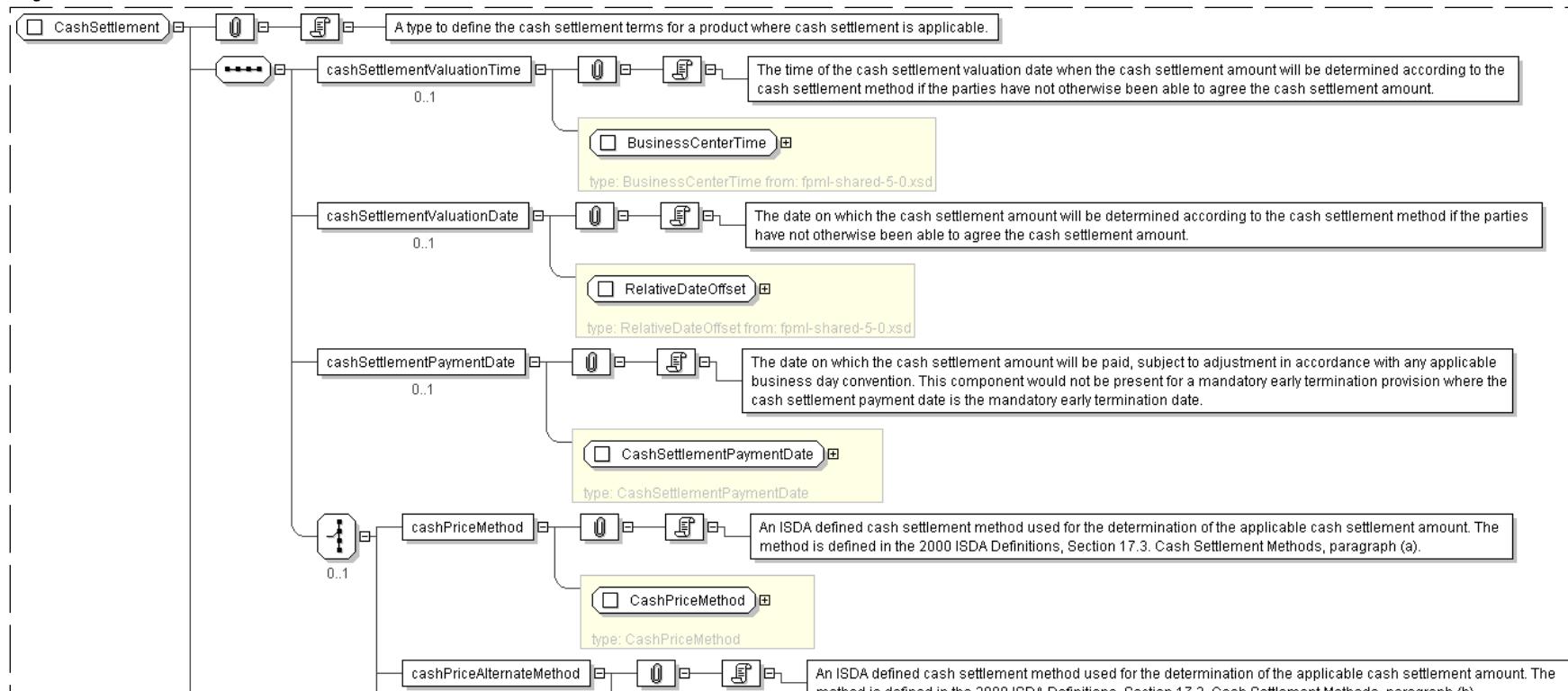
<parYieldCurveUnadjustedMethod> YieldCurveMethod </parYieldCurveUnadjustedMethod> [1]

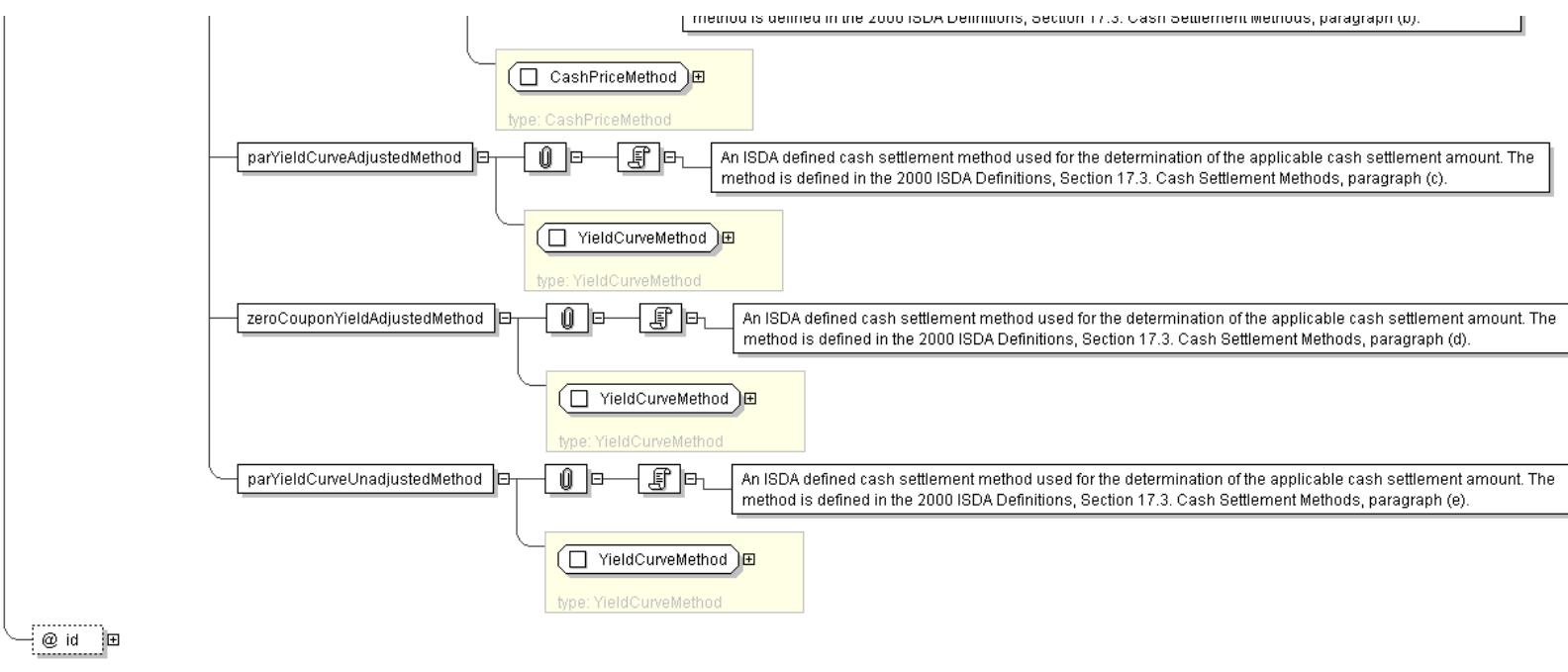
'An ISDA defined cash settlement method used for the determination of the applicable cash settlement amount. The method is defined in the 2000 ISDA Definitions, Section 17.3. Cash Settlement Methods, paragraph (e).'

End Choice

</...>

#### Diagram





## Schema Component Representation

```
<xsd:complexType name="CashSettlement">
  <xsd:sequence>
    <xsd:element name="cashSettlementValuationTime" type=" BusinessCenterTime " minOccurs="0" />
    <xsd:element name="cashSettlementValuationDate" type=" RelativeDateOffset " minOccurs="0" />
    <xsd:element name="cashSettlementPaymentDate" type=" CashSettlementPaymentDate " minOccurs="0" />
    <xsd:choice minOccurs="0">
      <xsd:element name="cashPriceMethod" type=" CashPriceMethod " />
      <xsd:element name="cashPriceAlternateMethod" type=" CashPriceMethod " />
      <xsd:element name="parYieldCurveAdjustedMethod" type=" YieldCurveMethod " />
      <xsd:element name="zeroCouponYieldAdjustedMethod" type=" YieldCurveMethod " />
      <xsd:element name="parYieldCurveUnadjustedMethod" type=" YieldCurveMethod " />
    </xsd:choice>
  </xsd:sequence>
  <xsd:attribute name="id" type=" xsd:ID " />
</xsd:complexType>
```

top

**Complex Type:** [CashSettlementPaymentDate](#)

<b>Super-types:</b>	None
<b>Sub-types:</b>	None

Name	CashSettlementPaymentDate
Used by (from the same schema document)	Complex Type <a href="#">CashSettlement</a>
Abstract	no
Documentation	A type defining the cash settlement payment date(s) as either a set of explicit dates, together with applicable adjustments, or as a date relative to some other (anchor) date, or as any date in a range of contiguous business days.

## XML Instance Representation

```
<...  
id=" xsd:ID [0..1]">  
Start Choice [1]  
    <adjustableDates> AdjustableDates </adjustableDates> [1]
```

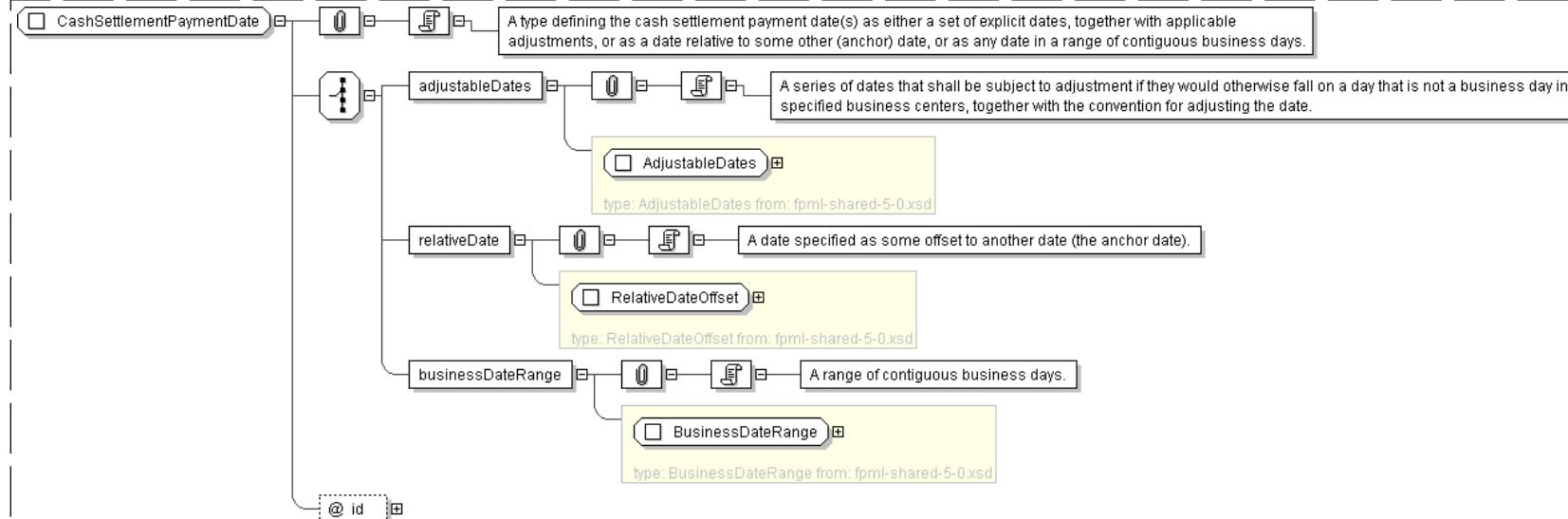
'A series of dates that shall be subject to adjustment if they would otherwise fall on a day that is not a business day in the specified business centers, together with the convention for adjusting the date.'

```
<relativeDate> RelativeDateOffset </relativeDate> [1]
'A date specified as some offset to another date (the anchor date).'

<businessDateRange> BusinessDateRange </businessDateRange> [1]
'A range of contiguous business days.'
```

End Choice

&lt; / ... &gt;

**Diagram****Schema Component Representation**

```

<xsd:complexType name="CashSettlementPaymentDate">
  <xsd:choice>
    <xsd:element name="adjustableDates" type=" AdjustableDates " />
    <xsd:element name="relativeDate" type=" RelativeDateOffset " />
    <xsd:element name="businessDateRange" type=" BusinessDateRange " />
  </xsd:choice>
  <xsd:attribute name="id" type=" xsd:ID ">
</xsd:complexType>
  
```

top

**Complex Type: Cashflows**

Super-types:	None
Sub-types:	None

Name	Cashflows
Used by (from the same schema document)	Complex Type <a href="#">InterestRateStream</a>

Abstract	no
----------	----

Documentation	A type defining the cashflow representation of a swap trade.
---------------	--

**XML Instance Representation**

```

<....>
|  <cashflowsMatchParameters> xsd:boolean </cashflowsMatchParameters> [1]
  
```

'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> PrincipalExchange </principalExchange> [0..\*]

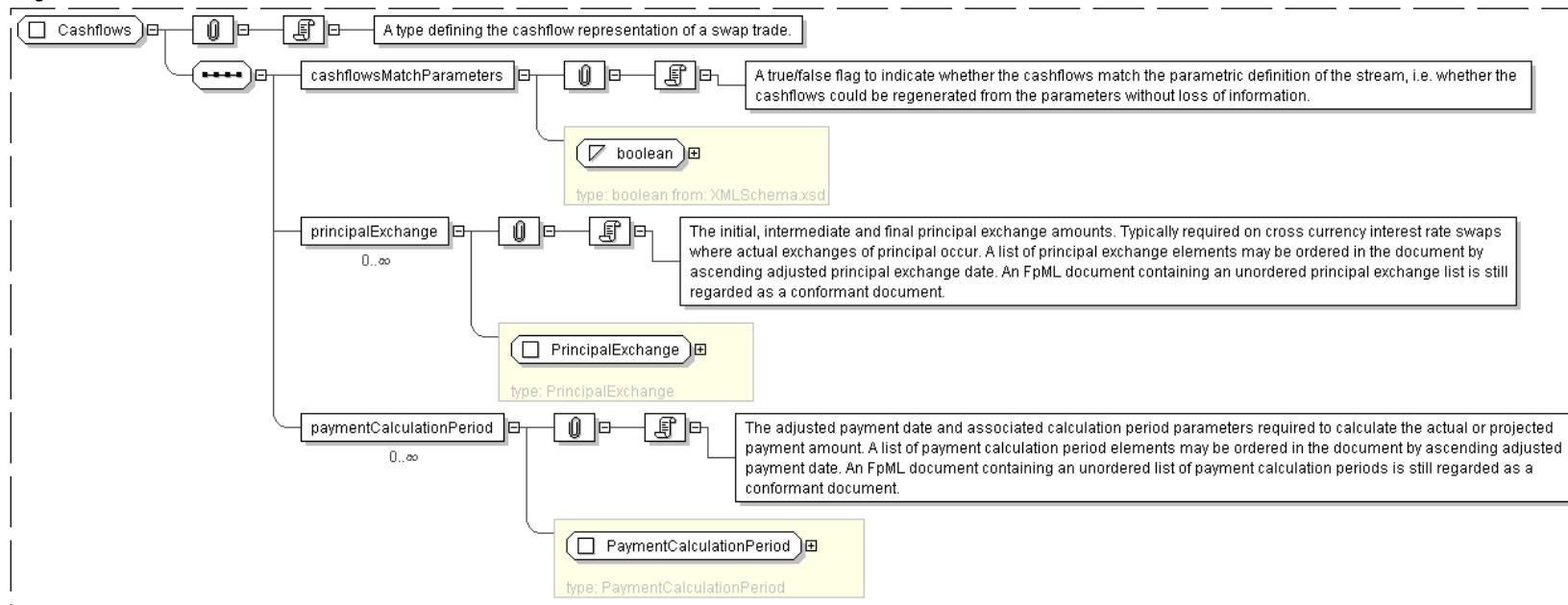
'The initial, intermediate and final principal exchange amounts. Typically required on cross currency interest rate swaps where actual exchanges of principal occur. A list of principal exchange elements may be ordered in the document by ascending adjusted principal exchange date. An FpML document containing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

<paymentCalculationPeriod> PaymentCalculationPeriod </paymentCalculationPeriod> [0..\*]

'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.'

</...>

#### Diagram



#### Schema Component Representation

```

<xsd:complexType name="Cashflows">
  <xsd:sequence>
    <xsd:element name="cashflowsMatchParameters" type="xsd:boolean" />
    <xsd:element name="principalExchange" type="PrincipalExchange" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element name="paymentCalculationPeriod" type="PaymentCalculationPeriod" minOccurs="0" maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>
  
```

top

#### Complex Type: DateRelativeToCalculationPeriodDates

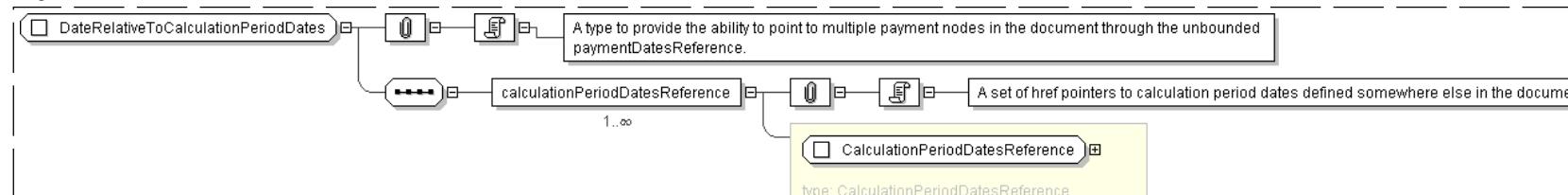
Super-types:	None
Sub-types:	None

<b>Name</b>	DateRelativeToCalculationPeriodDates
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">FxFixingDate</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type to provide the ability to point to multiple payment nodes in the document through the unbounded paymentDatesReference.

**XML Instance Representation**

```
<....>
<calculationPeriodDatesReference> CalculationPeriodDatesReference
</calculationPeriodDatesReference> [1..*]
'A set of href pointers to calculation period dates defined somewhere else in the document.'

</....>
```

**Diagram****Schema Component Representation**

```
<xsd:complexType name="DateRelativeToCalculationPeriodDates">
  <xsd:sequence>
    <xsd:element name="calculationPeriodDatesReference" type=" CalculationPeriodDatesReference
      " maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>
```

top

**Complex Type: DateRelativeToPaymentDates**

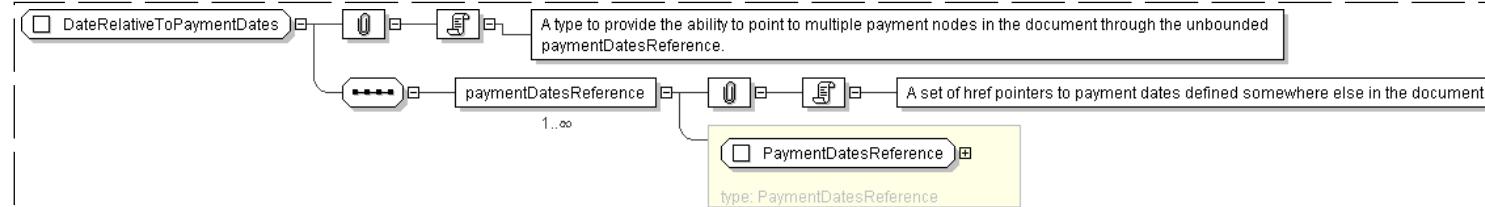
<b>Super-types:</b>	None
<b>Sub-types:</b>	None

<b>Name</b>	DateRelativeToPaymentDates
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">FxFixingDate</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type to provide the ability to point to multiple payment nodes in the document through the unbounded paymentDatesReference.

**XML Instance Representation**

```
<....>
<paymentDatesReference> PaymentDatesReference </paymentDatesReference> [1..*]
'A set of href pointers to payment dates defined somewhere else in the document.'

</....>
```

**Diagram**

## Schema Component Representation

```
<xsd:complexType name="DateRelativeToPaymentDates">
  <xsd:sequence>
    <xsd:element name="paymentDatesReference" type=" PaymentDatesReference " maxOccurs="unbounded" />
  </xsd:sequence>
</xsd:complexType>
```

[top](#)Complex Type: **Discounting**

Super-types:

None

Sub-types:

None

Name	Discounting
Used by (from the same schema document)	Complex Type <a href="#">Calculation</a>
Abstract	no
Documentation	A type 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 type must only be included if discounting applies.

## XML Instance Representation

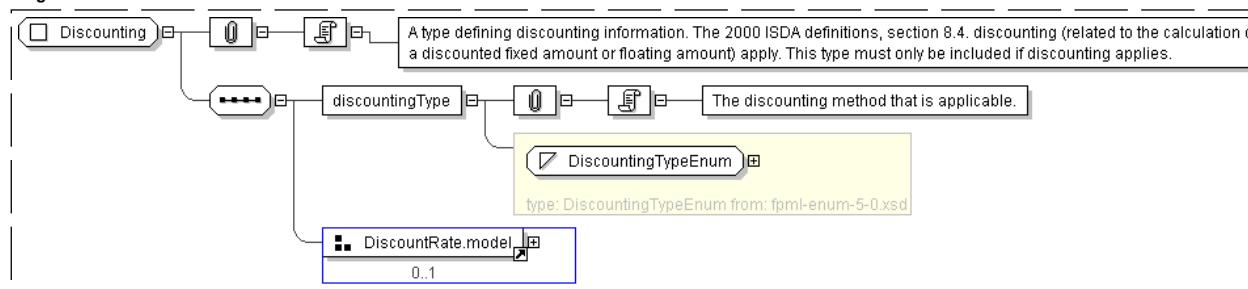
```
<...>
  <discountingType> DiscountingTypeEnum </discountingType> [1]
  'The discounting method that is applicable.'

Start Group: DiscountRate.model [0..1]
  <discountRate> xsd:decimal </discountRate> [1]
  'A discount rate, expressed as a decimal, to be used in the calculation of a discounted amount. A discount amount of 5% would be represented as 0.05.'

  <discountRateDayCountFraction> DayCountFraction </discountRateDayCountFraction> [0..1]
  'A discount day count fraction to be used in the calculation of a discounted amount.'

End Group: DiscountRate.model
</...>
```

## Diagram



## Schema Component Representation

```
<xsd:complexType name="Discounting">
  <xsd:sequence>
    <xsd:element name="discountingType" type=" DiscountingTypeEnum " />
    <xsd:group ref=" DiscountRate.model " minOccurs="0 " />
  </xsd:sequence>
</xsd:complexType>
```

[top](#)Complex Type: **EarlyTerminationEvent**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None

<b>Name</b>	EarlyTerminationEvent
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">OptionalEarlyTerminationAdjustedDates</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type to define the adjusted dates associated with an early termination provision.

**XML Instance Representation**

```
<...>
  id=" xsd:ID [0..1]">
    <adjustedExerciseDate> xsd:date </adjustedExerciseDate> [1]
    'The date on which option exercise takes place. This date should already be adjusted for
    any applicable business day convention.'

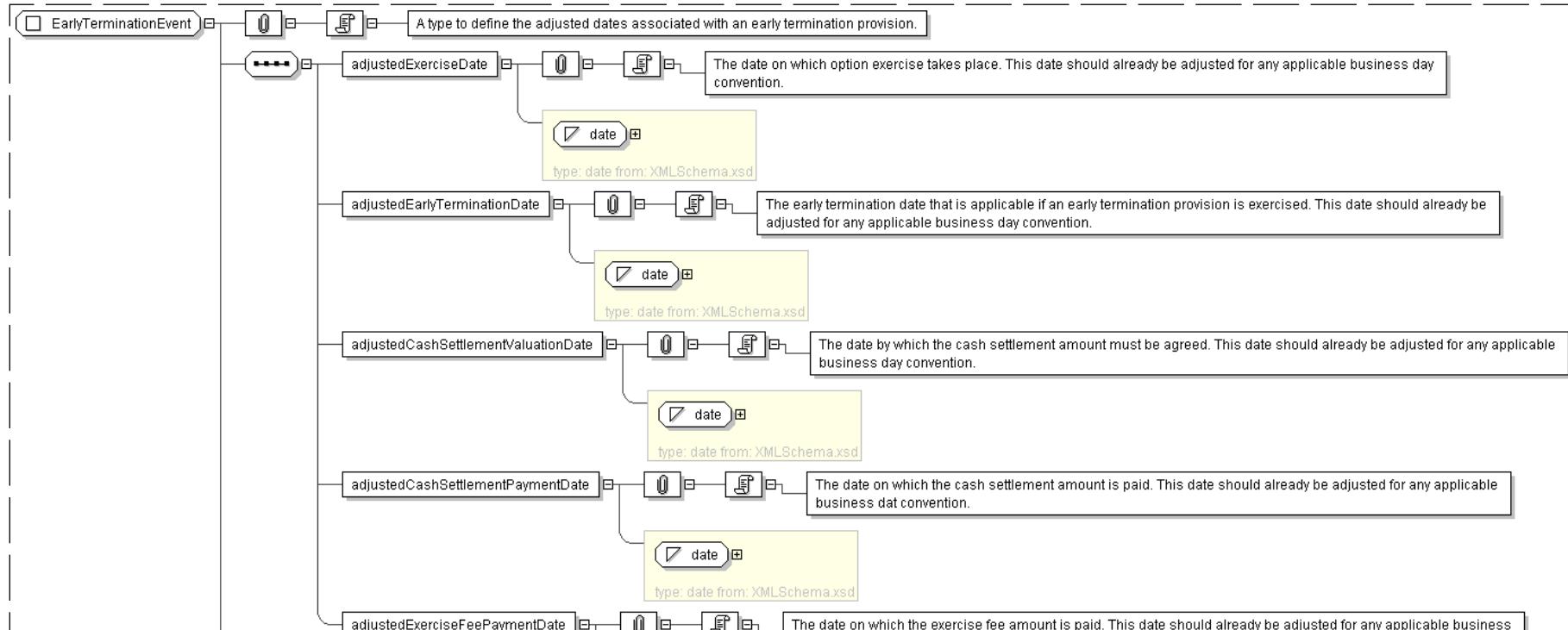
    <adjustedEarlyTerminationDate> xsd:date </adjustedEarlyTerminationDate> [1]
    'The early termination date that is applicable if an early termination provision is
    exercised. This date should already be adjusted for any applicable business day convention.'

    <adjustedCashSettlementValuationDate> xsd:date </adjustedCashSettlementValuationDate> [1]
    'The date by which the cash settlement amount must be agreed. This date should already
    be adjusted for any applicable business day convention.'

    <adjustedCashSettlementPaymentDate> xsd:date </adjustedCashSettlementPaymentDate> [1]
    'The date on which the cash settlement amount is paid. This date should already be adjusted
    for any applicable business dat convention.'

    <adjustedExerciseFeePaymentDate> xsd:date </adjustedExerciseFeePaymentDate> [0..1]
    'The date on which the exercise fee amount is paid. This date should already be adjusted
    for any applicable business day convention.'

</...>
```

**Diagram**

**Schema Component Representation**

```

<xsd:complexType name="EarlyTerminationEvent">
  <xsd:sequence>
    <xsd:element name="adjustedExerciseDate" type="xsd:date" />
    <xsd:element name="adjustedEarlyTerminationDate" type="xsd:date" />
    <xsd:element name="adjustedCashSettlementValuationDate" type="xsd:date" />
    <xsd:element name="adjustedCashSettlementPaymentDate" type="xsd:date" />
    <xsd:element name="adjustedExerciseFeePaymentDate" type="xsd:date" minOccurs="0"/>
  </xsd:sequence>
  <xsd:attribute name="id" type="xsd:ID" />
</xsd:complexType>
  
```

top

**Complex Type: EarlyTerminationProvision**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None
<b>Name</b>	EarlyTerminationProvision
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">CapFloor</a> , Complex Type <a href="#">Swap</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type defining an early termination provision for a swap. This early termination is at fair value, i.e. on termination the fair value of the product must be settled between the parties.

**XML Instance Representation**

```

<...
  id="xsd:ID [0..1]">
  Start Choice [1]
  Start Choice [1]
    <mandatoryEarlyTermination> MandatoryEarlyTermination </mandatoryEarlyTermination> [1]
    'A mandatory early termination provision to terminate the swap at fair value.'

    <mandatoryEarlyTerminationDateTenor> Period </mandatoryEarlyTerminationDateTenor> [1]
    'Period after trade date of the mandatory early termination date.'

    <mandatoryEarlyTermination> MandatoryEarlyTermination </mandatoryEarlyTermination> [0..1]
    'A mandatory early termination provision to terminate the swap at fair value.'

  End Choice
  Start Group: OptionalEarlyTermination.model [0..1]
  Start Choice [1]
    <optionalEarlyTermination> OptionalEarlyTermination </optionalEarlyTermination> [1]
    'An option for either or both parties to terminate the swap at fair value.'

    <optionalEarlyTerminationParameters> ExercisePeriod </optionalEarlyTerminationParameters> [1]
    'Definition of the first early termination date and the frequency of the termination
    dates subsequent to that. American exercise is defined by having a frequency of one day.'

    <optionalEarlyTermination> OptionalEarlyTermination </optionalEarlyTermination> [0..1]
    'An option for either or both parties to terminate the swap at fair value.'

  End Choice
  End Group: OptionalEarlyTermination.model
  Start Choice [1]
  
```

```

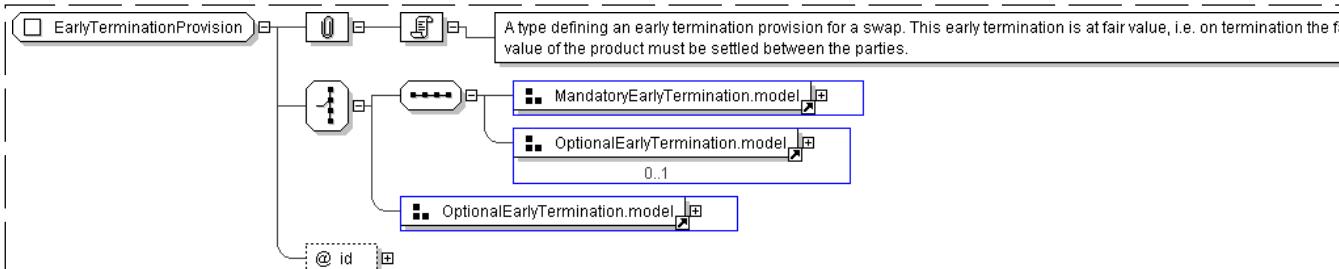
<optionalEarlyTermination> OptionalEarlyTermination </optionalEarlyTermination> [1]
  'An option for either or both parties to terminate the swap at fair value.'

<optionalEarlyTerminationParameters> ExercisePeriod </optionalEarlyTerminationParameters> [1]
  'Definition of the first early termination date and the frequency of the termination
  dates subsequent to that. American exercise is defined by having a frequency of one day.'

<optionalEarlyTermination> OptionalEarlyTermination </optionalEarlyTermination> [0..1]
  'An option for either or both parties to terminate the swap at fair value.'

```

End Choice  
End Choice  
...>

**Diagram****Schema Component Representation**

```

<xsd:complexType name="EarlyTerminationProvision">
  <xsd:choice>
    <xsd:sequence>
      <xsd:group ref=" MandatoryEarlyTermination.model " />
      <xsd:group ref=" OptionalEarlyTermination.model " minOccurs="0" />
    </xsd:sequence>
    <xsd:group ref=" OptionalEarlyTermination.model " />
  </xsd:choice>
  <xsd:attribute name="id" type=" xsd:ID " />
</xsd:complexType>

```

top

**Complex Type: ExerciseEvent**

Super-types:

None

Sub-types:

None

Name

ExerciseEvent

Used by (from the same schema document)

Complex Type [SwaptionAdjustedDates](#)

Abstract

no

Documentation

A type defining the adjusted dates associated with a particular exercise event.

**XML Instance Representation**

```

<...
  id=" xsd:ID [0..1]">
  <adjustedExerciseDate> xsd:date </adjustedExerciseDate> [1]
  'The date on which option exercise takes place. This date should already be adjusted for
  any applicable business day convention.'

  <adjustedRelevantSwapEffectiveDate> xsd:date </adjustedRelevantSwapEffectiveDate> [1]
  '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> xsd:date </adjustedCashSettlementValuationDate> [0..1]

```

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

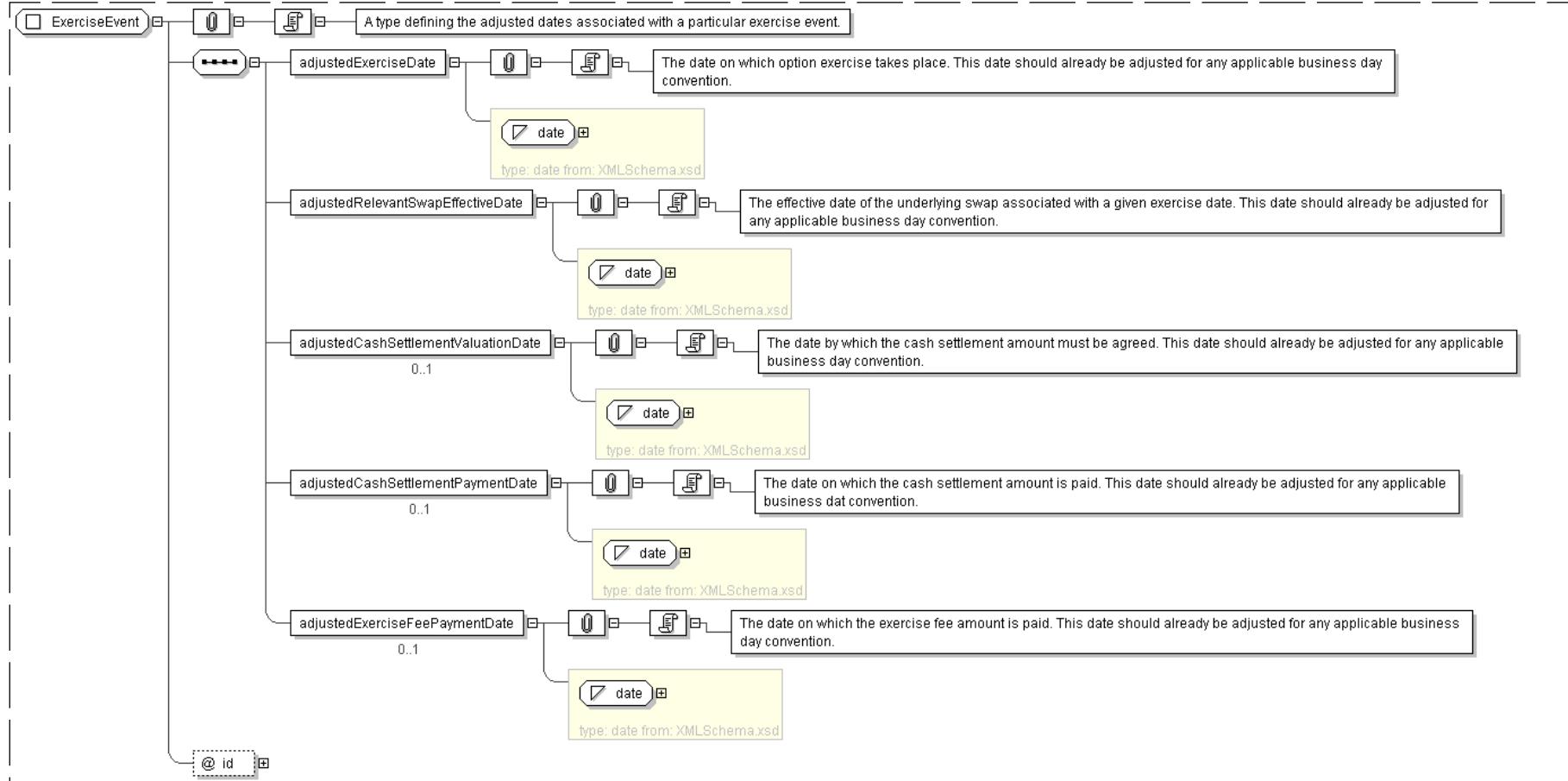
<adjustedCashSettlementPaymentDate> xsd:date </adjustedCashSettlementPaymentDate> [0..1]

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

<adjustedExerciseFeePaymentDate> xsd:date </adjustedExerciseFeePaymentDate> [0..1]

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

&lt;/...&gt;

**Diagram****Schema Component Representation**

```

<xsd:complexType name="ExerciseEvent">
  <xsd:sequence>
    <xsd:element name="adjustedExerciseDate" type="xsd:date" />
    <xsd:element name="adjustedRelevantSwapEffectiveDate" type="xsd:date" />
    <xsd:element name="adjustedCashSettlementValuationDate" type="xsd:date" minOccurs="0"/>
    <xsd:element name="adjustedCashSettlementPaymentDate" type="xsd:date" minOccurs="0"/>
    <xsd:element name="adjustedExerciseFeePaymentDate" type="xsd:date" minOccurs="0"/>
  </xsd:sequence>
  <xsd:attribute name="id" type="xsd:ID" />
</xsd:complexType>
  
```

## Complex Type: ExercisePeriod

Super-types:	None
Sub-types:	None
Name	ExercisePeriod
Used by (from the same schema document)	Model Group <a href="#">OptionalEarlyTermination.model</a>
Abstract	no
Documentation	This defines the time interval to the start of the exercise period, i.e. the earliest exercise date, and the frequency of subsequent exercise dates (if any).

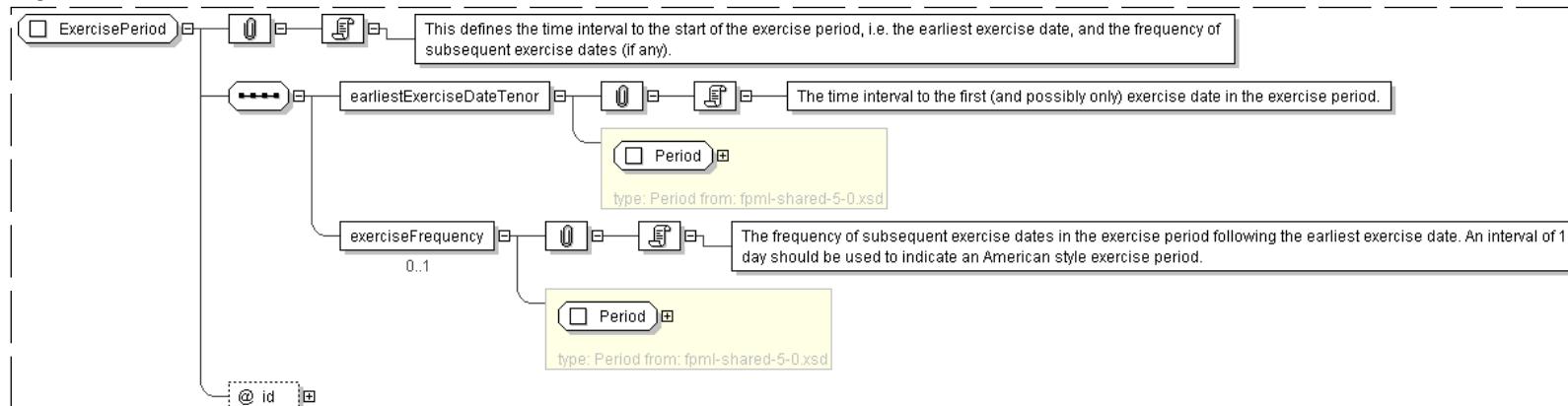
### XML Instance Representation

```
<...>
  id="xsd:ID [0..1]">
    <earliestExerciseDateTenor> Period </earliestExerciseDateTenor> [1]
    'The time interval to the first (and possibly only) exercise date in the exercise period.'

    <exerciseFrequency> Period </exerciseFrequency> [0..1]
    'The frequency of subsequent exercise dates in the exercise period following the
    earliest exercise date. An interval of 1 day should be used to indicate an American
    style exercise period.'

</...>
```

### Diagram



### Schema Component Representation

```
<xsd:complexType name="ExercisePeriod">
  <xsd:sequence>
    <xsd:element name="earliestExerciseDateTenor" type="Period"/>
    <xsd:element name="exerciseFrequency" type="Period" minOccurs="0"/>
  </xsd:sequence>
  <xsd:attribute name="id" type="xsd:ID"/>
</xsd:complexType>
```

## Complex Type: ExtendibleProvision

Super-types:	None
Sub-types:	None
Name	ExtendibleProvision
Description	ExtendibleProvision

<b>Used by (from the same schema document)</b>	Complex Type <a href="#">Swap</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type defining an option to extend an existing swap transaction on the specified exercise dates for a term ending on the specified new termination date.

**XML Instance Representation**

```

<....>
<buyerPartyReference> PartyReference </buyerPartyReference> [1]
'A reference to the party that buys this instrument, i.e. pays for this instrument and
receives the rights defined by it. See 2000 ISDA definitions Article 11.1 (b). In the case
of FRAs this is the fixed rate payer.'

<buyerAccountReference> AccountReference </buyerAccountReference> [0..1]
'A reference to the account that buys this instrument.'

<sellerPartyReference> PartyReference </sellerPartyReference> [1]
'A reference to the party that sells ("writes") this instrument, i.e. that grants the
rights defined by this instrument and in return receives a payment for it. See 2000
ISDA definitions Article 11.1 (a). In the case of FRAs this is the floating rate payer.'

<sellerAccountReference> AccountReference </sellerAccountReference> [0..1]
'A reference to the account that sells this instrument.'

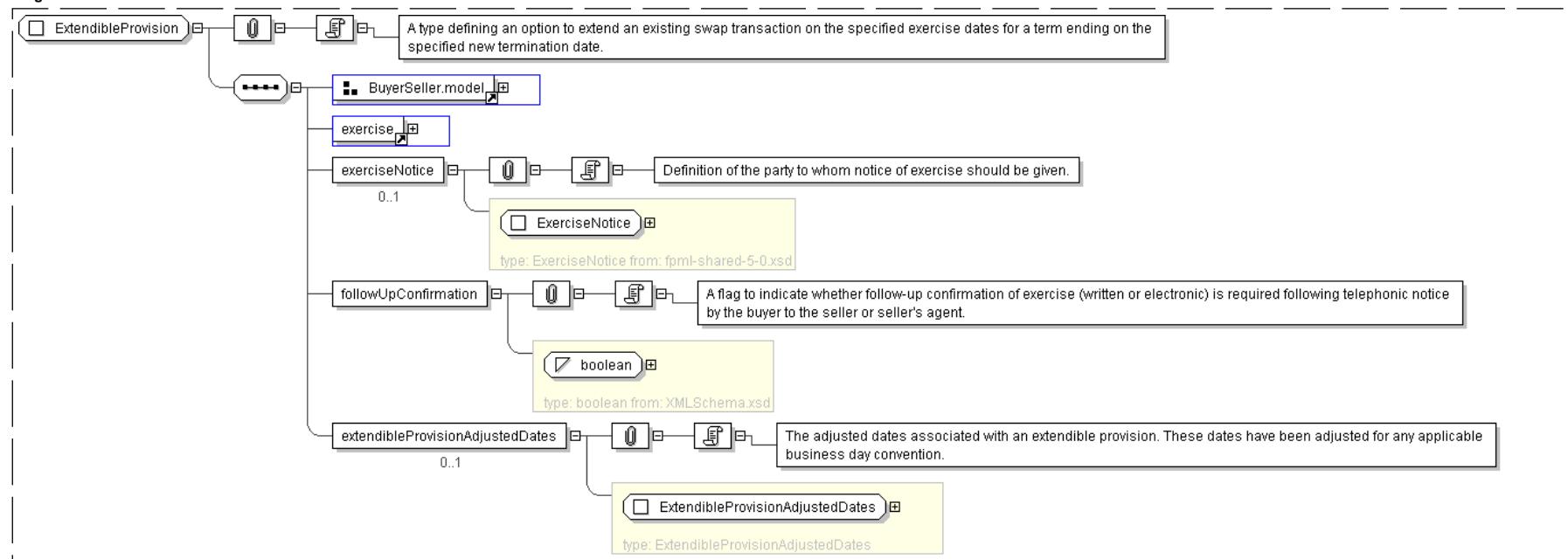
<exercise> ... </exercise> [1]
<exerciseNotice> ExerciseNotice </exerciseNotice> [0..1]
'Definition of the party to whom notice of exercise should be given.'

<followUpConfirmation> xsd:boolean </followUpConfirmation> [1]
'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> ExtendibleProvisionAdjustedDates
</extendibleProvisionAdjustedDates> [0..1]
'The adjusted dates associated with an extendible provision. These dates have been adjusted
for any applicable business day convention.'

</....>

```

**Diagram**

## Schema Component Representation

```
<xsd:complexType name="ExtendibleProvision">
  <xsd:sequence>
    <xsd:group ref=" BuyerSeller.model ">/>
    <xsd:element ref=" exercise ">/>
    <xsd:element name="exerciseNotice" type=" ExerciseNotice " minOccurs="0 "/>
    <xsd:element name="followUpConfirmation" type=" xsd:boolean ">/>
    <xsd:element name="extendibleProvisionAdjustedDates" type=" ExtendibleProvisionAdjustedDates "
      " minOccurs="0 "/>
  </xsd:sequence>
</xsd:complexType>
```

[top](#)

## Complex Type: ExtendibleProvisionAdjustedDates

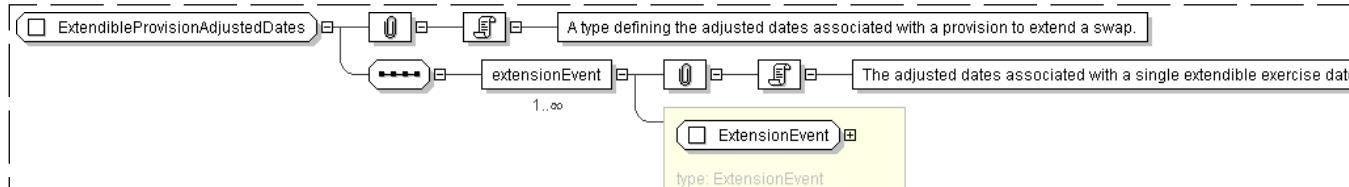
Super-types: None  
 Sub-types: None

Name	ExtendibleProvisionAdjustedDates
Used by (from the same schema document)	Complex Type <a href="#">ExtendibleProvision</a>
Abstract	no
Documentation	A type defining the adjusted dates associated with a provision to extend a swap.

## XML Instance Representation

```
<....>
  <extensionEvent> ExtensionEvent </extensionEvent> [1..*]
    'The adjusted dates associated with a single extendible exercise date.'
</....>
```

## Diagram



## Schema Component Representation

```
<xsd:complexType name="ExtendibleProvisionAdjustedDates">
  <xsd:sequence>
    <xsd:element name="extensionEvent" type=" ExtensionEvent " maxOccurs="unbounded "/>
  </xsd:sequence>
</xsd:complexType>
```

[top](#)

## Complex Type: ExtensionEvent

Super-types: None  
 Sub-types: None

Name	ExtensionEvent
Used by (from the same schema document)	Complex Type <a href="#">ExtendibleProvisionAdjustedDates</a>
Abstract	no
Documentation	A type to define the adjusted dates associated with an individual extension event.

## XML Instance Representation

```
file:///C:/Irina-Local/Subversion/branches/FpML-5.0/pdf/confirmation/fpm-ird-5-0.xsd.html (63 of 113) [7/7/2010 11:06:50 AM]
```

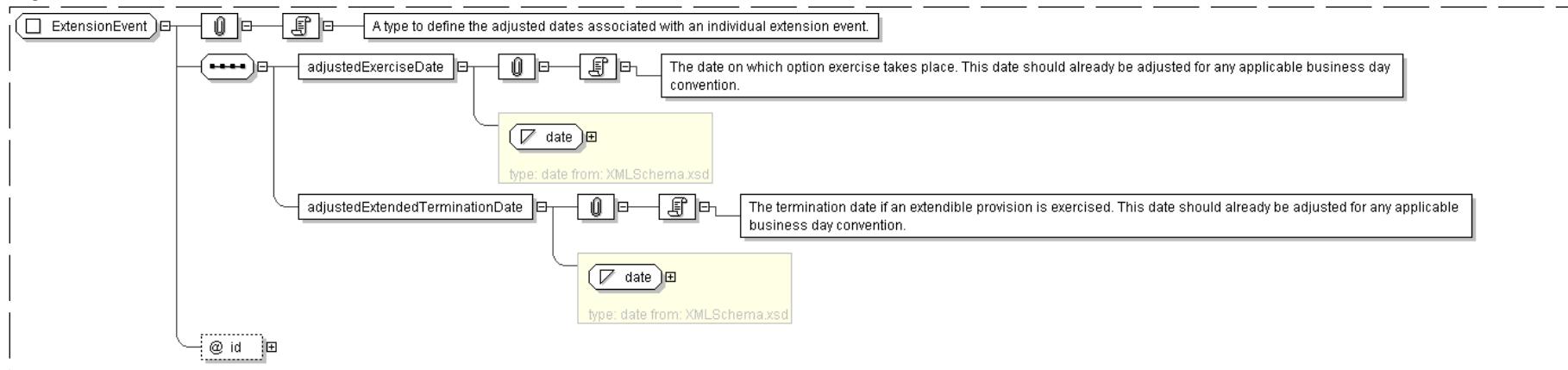
```

<...
  id="xsd:ID [0..1]">
  <adjustedExerciseDate> xsd:date </adjustedExerciseDate> [1]
  'The date on which option exercise takes place. This date should already be adjusted for
  any applicable business day convention.'

  <adjustedExtendedTerminationDate> xsd:date </adjustedExtendedTerminationDate> [1]
  'The termination date if an extendible provision is exercised. This date should already
  be adjusted for any applicable business day convention.'

</...>

```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="ExtensionEvent">
  <xsd:sequence>
    <xsd:element name="adjustedExerciseDate" type="xsd:date" />
    <xsd:element name="adjustedExtendedTerminationDate" type="xsd:date" />
  </xsd:sequence>
  <xsd:attribute name="id" type="xsd:ID" />
</xsd:complexType>

```

[top](#)**Complex Type: FallbackReferencePrice**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None
<b>Name</b>	FallbackReferencePrice
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">PriceSourceDisruption</a>
<b>Abstract</b>	no
<b>Documentation</b>	The method, prioritized by the order it is listed in this element, to get a replacement rate for the disrupted settlement rate option.

**XML Instance Representation**

```

<...
  <valuationPostponement> ValuationPostponement </valuationPostponement> [0..1]
  'Specifies how long to wait to get a quote from a settlement rate option upon a price
  source disruption'

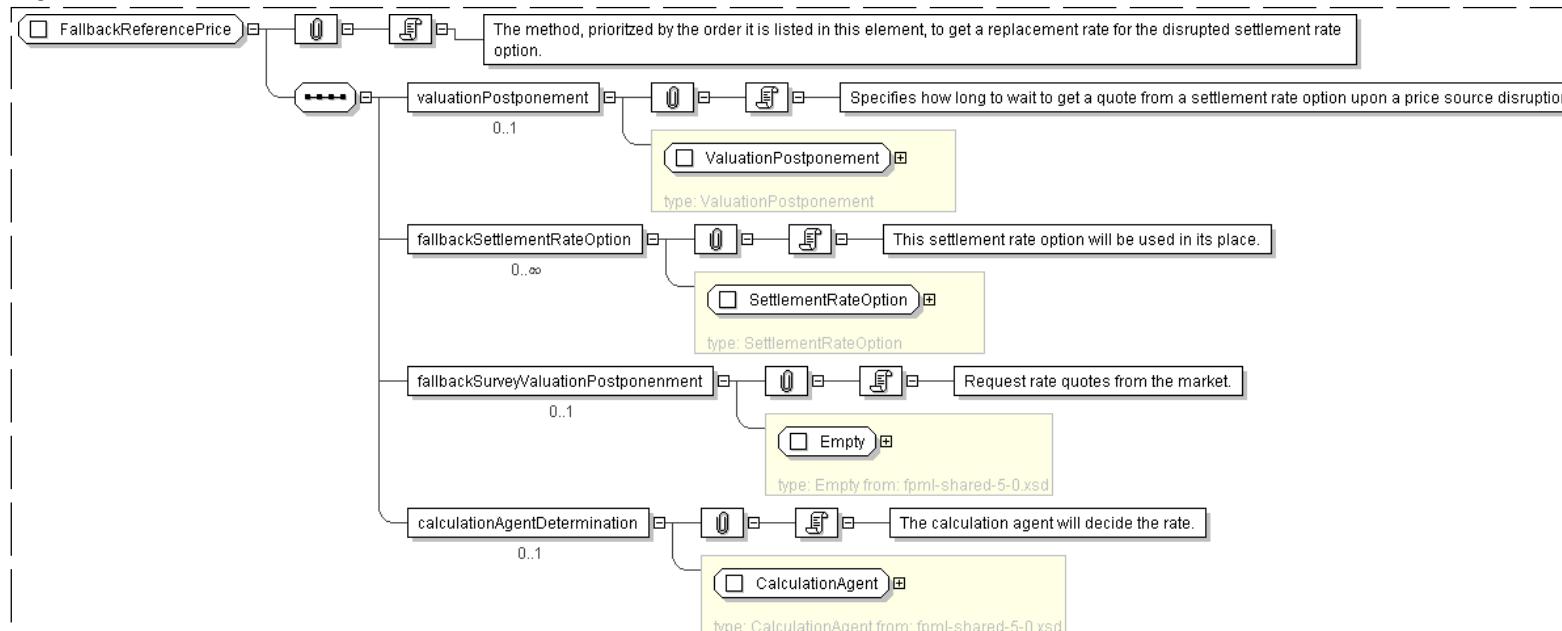
  <fallbackSettlementRateOption> SettlementRateOption </fallbackSettlementRateOption> [0..*]
  'This settlement rate option will be used in its place.'

  <fallbackSurveyValuationPostponement> Empty </fallbackSurveyValuationPostponement> [0..1]
  'Request rate quotes from the market.'
</...>

```

```
<calculationAgentDetermination> CalculationAgent </calculationAgentDetermination> [0..1]
  The calculation agent will decide the rate.'
```

&lt;...&gt;

**Diagram****Schema Component Representation**

```

<xsd:complexType name="FallbackReferencePrice">
  <xsd:sequence>
    <xsd:element name="valuationPostponement" type=" ValuationPostponement " minOccurs="0"/>
    <xsd:element name="fallbackSettlementRateOption" type=" SettlementRateOption "
      minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element name="fallbackSurveyValuationPostponement" type=" Empty " minOccurs="0"/>
    <xsd:element name="calculationAgentDetermination" type=" CalculationAgent " minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>
  
```

top

**Complex Type: FinalCalculationPeriodDateAdjustment**

Super-types:

None

Sub-types:

None

**Name**

FinalCalculationPeriodDateAdjustment

**Used by (from the same schema document)**Complex Type [CancelableProvision](#)**Abstract**

no

**Documentation**

A type to define business date convention adjustment to final payment period per leg.

**XML Instance Representation**

```

<...>
  <relevantUnderlyingDateReference> RelevantUnderlyingDateReference
  </relevantUnderlyingDateReference> [1]
  'Reference to the unadjusted cancellation effective dates.'
  
```

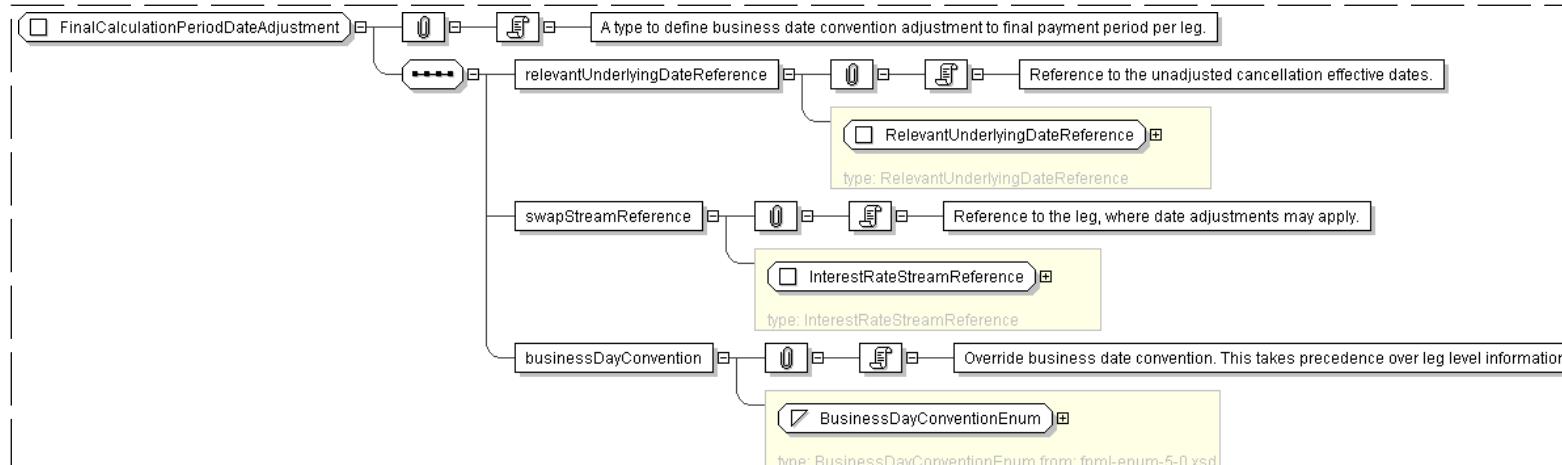
```
<swapStreamReference> InterestRateStreamReference </swapStreamReference> [1]
```

'Reference to the leg, where date adjustments may apply.'

```
<businessDayConvention> BusinessDayConventionEnum </businessDayConvention> [1]
```

'Override business date convention. This takes precedence over leg level information.'

&lt;/...&gt;

**Diagram****Schema Component Representation**

```

<xsd:complexType name="FinalCalculationPeriodDateAdjustment">
  <xsd:sequence>
    <xsd:element name="relevantUnderlyingDateReference" type=" RelevantUnderlyingDateReference " />
    <xsd:element name="swapStreamReference" type=" InterestRateStreamReference " />
    <xsd:element name="businessDayConvention" type=" BusinessDayConventionEnum " />
  </xsd:sequence>
</xsd:complexType>
  
```

top

**Complex Type: FloatingRateDefinition**

Super-types:

None

Sub-types:

None

Name	FloatingRateDefinition
------	------------------------

Used by (from the same schema document)	Complex Type <a href="#">CalculationPeriod</a>
---	--

Abstract	no
----------	----

Documentation	A type defining parameters associated with a floating rate reset. This type forms part of the cashflows representation of a stream.
---------------	---

**XML Instance Representation**

```
<...>
<calculatedRate> xsd:decimal </calculatedRate> [0..1]
```

'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> RateObservation </rateObservation> [0..*]
```

'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> xsd:decimal </floatingRateMultiplier> [0..1]
'A rate multiplier to apply to the floating rate. The multiplier can be a positive or
negative decimal. This element should only be included if the multiplier is not equal to
1 (one).'

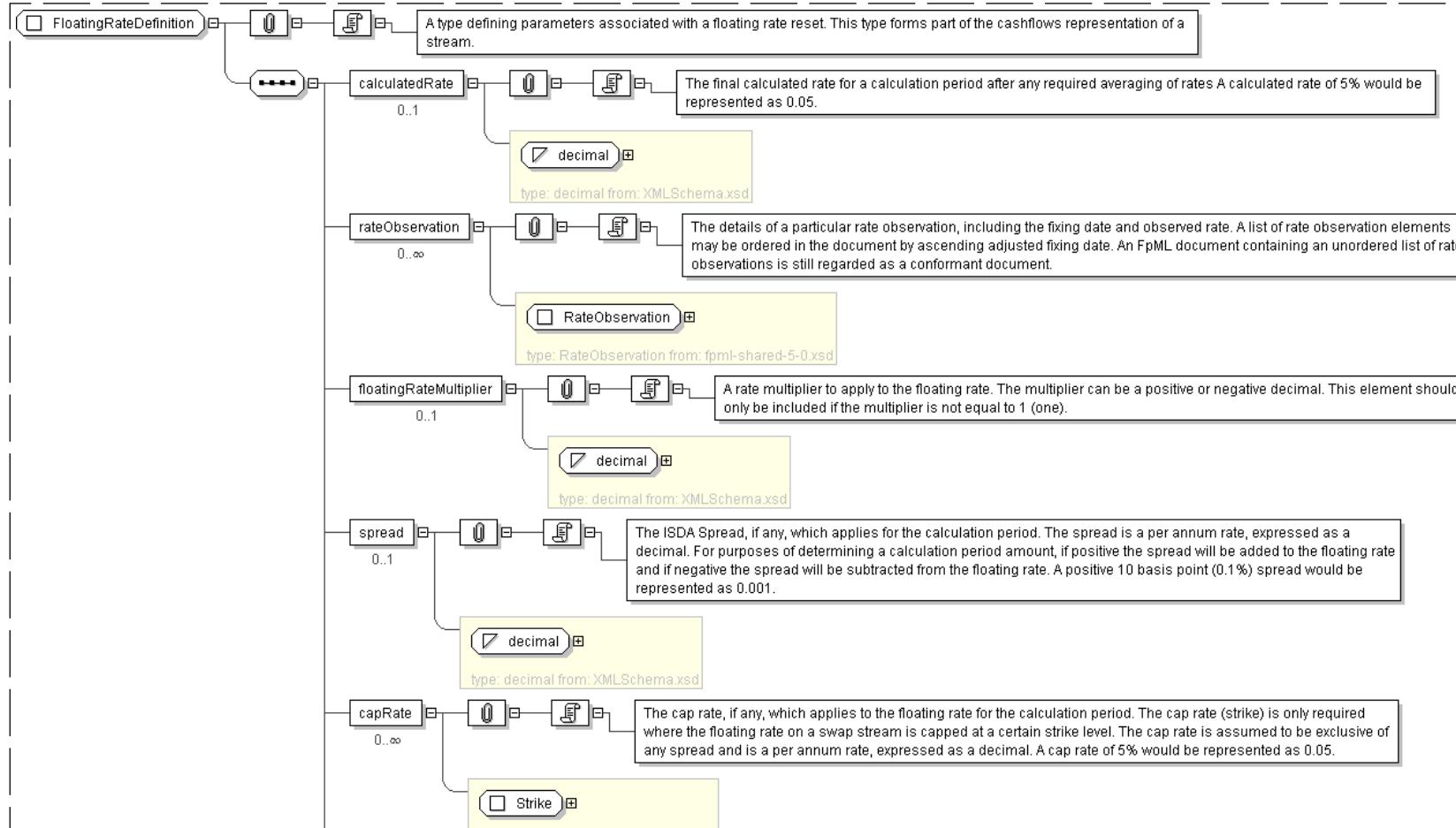
<spread> xsd:decimal </spread> [0..1]
'The ISDA Spread, if any, which applies for the calculation period. The spread is a per
annum rate, expressed as a decimal. For purposes of determining a calculation period amount,
if positive the spread will be added to the floating rate and if negative the spread will
be subtracted from the floating rate. A positive 10 basis point (0.1%) spread would
be represented as 0.001.'

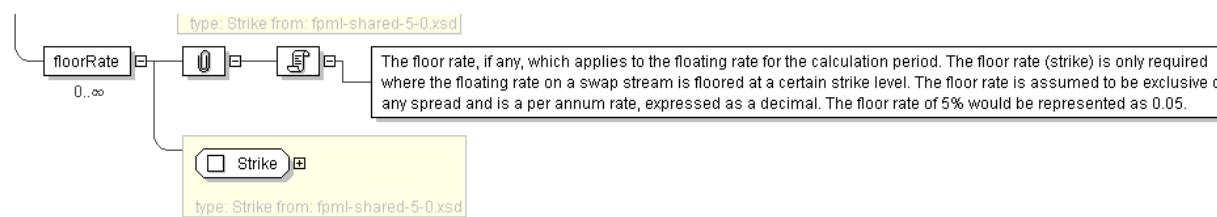
<capRate> Strike </capRate> [0..*]
'The cap rate, if any, which applies to the floating rate for the calculation period. The
cap rate (strike) is only required where the floating rate on a swap stream is capped at
a certain strike level. The cap rate is assumed to be exclusive of any spread and is a
per annum rate, expressed as a decimal. A cap rate of 5% would be represented as 0.05.'

<floorRate> Strike </floorRate> [0..*]
'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. The floor rate of 5% would be represented as 0.05.'

</...>

```

**Diagram**

**Schema Component Representation**

```

<xsd:complexType name="FloatingRateDefinition">
  <xsd:sequence>
    <xsd:element name="calculatedRate" type="xsd:decimal" minOccurs="0"/>
    <xsd:element name="rateObservation" type="RateObservation" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element name="floatingRateMultiplier" type="xsd:decimal" minOccurs="0"/>
    <xsd:element name="spread" type="xsd:decimal" minOccurs="0"/>
    <xsd:element name="capRate" type="Strike" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element name="floorRate" type="Strike" minOccurs="0" maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>

```

[top](#)**Complex Type: Fra**

Super-types:	<a href="#">Product</a> < <b>Fra</b> (by extension)
Sub-types:	None

Name	Fra
Used by (from the same schema document)	Element <a href="#">fra</a>
Abstract	no
Documentation	A type defining a Forward Rate Agreement (FRA) product.

**XML Instance Representation**

```

<...
  id="xsd:ID [0..1]">
  <productType> ProductType </productType> [0..*]
  'A classification of the type of product. FpML defines a simple product categorization using a coding scheme.'
  <productId> ProductId </productId> [0..*]
  'A product 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.'
  <buyerPartyReference> PartyReference </buyerPartyReference> [1]
  'A reference to the party that buys this instrument, i.e. pays for this instrument and receives the rights defined by it. See 2000 ISDA definitions Article 11.1 (b). In the case of FRAs this is the fixed rate payer.'
  <buyerAccountReference> AccountReference </buyerAccountReference> [0..1]
  'A reference to the account that buys this instrument.'
  <sellerPartyReference> PartyReference </sellerPartyReference> [1]
  'A reference to the party that sells ("writes") this instrument, i.e. that grants the rights defined by this instrument and in return receives a payment for it. See 2000 ISDA definitions Article 11.1 (a). In the case of FRAs this is the floating rate payer.'
  <sellerAccountReference> AccountReference </sellerAccountReference> [0..1]
  'A reference to the account that sells this instrument.'
  <adjustedEffectiveDate> RequiredIdentifierDate </adjustedEffectiveDate> [1]

```

'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> xsd:date </adjustedTerminationDate> [1]

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

<paymentDate> AdjustableDate </paymentDate> [1]

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

<fixingDateOffset> RelativeDateOffset </fixingDateOffset> [1]

'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> DayCountFraction </dayCountFraction> [1]

'The day count fraction.'

<calculationPeriodNumberOfDays> xsd:positiveInteger </calculationPeriodNumberOfDays> [1]

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

<notional> Money </notional> [1]

'The notional amount.'

<fixedRate> xsd:decimal </fixedRate> [1]

'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> FloatingRateIndex </floatingRateIndex> [1]

<indexTenor> Period </indexTenor> [1..\*]

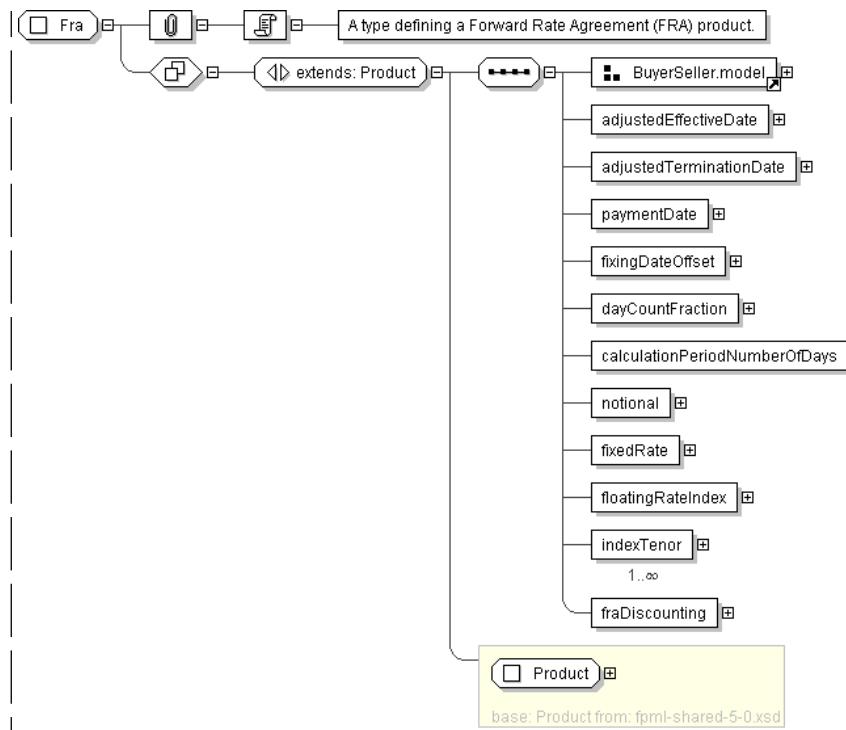
'The ISDA Designated Maturity, i.e. the tenor of the floating rate.'

<fraDiscounting> FraDiscountingEnum </fraDiscounting> [1]

'Specifies whether discounting applies and, if so, what type.'

</...>

Diagram



## Schema Component Representation

```

<xsd:complexType name="Fra">
  <xsd:complexContent>
    <xsd:extension base="Product">
      <xsd:sequence>
        <xsd:group ref=" BuyerSeller.model " />
        <xsd:element name="adjustedEffectiveDate" type="RequiredIdentifierDate " />
        <xsd:element name="adjustedTerminationDate" type="xsd:date" />
        <xsd:element name="paymentDate" type="AdjustableDate " />
        <xsd:element name="fixingDateOffset" type="RelativeDateOffset " />
        <xsd:element name="dayCountFraction" type="DayCountFraction " />
        <xsd:element name="calculationPeriodNumberOfDays" type="xsd:positiveInteger" />
        <xsd:element name="notional" type="Money " />
        <xsd:element name="fixedRate" type="xsd:decimal" />
        <xsd:element name="floatingRateIndex" type="FloatingRateIndex " />
        <xsd:element name="indexTenor" type="Period" maxOccurs="unbounded"/>
        <xsd:element name="fraDiscounting" type="FraDiscountingEnum " />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
  
```

top

Complex Type: [FxFixingDate](#)

Super-types:	<a href="#">Offset</a> < <a href="#">FxFixingDate</a> (by extension)
Sub-types:	None
Name	<a href="#">FxFixingDate</a>
Used by (from the same schema document)	Complex Type <a href="#">NonDeliverableSettlement</a>
Abstract	no

**Documentation**

A type that is extending the Offset structure for providing the ability to specify an FX fixing date as an offset to dates specified somewhere else in the document.

**XML Instance Representation**

```
<...>
  id=" xsd:ID [0..1]">
    <periodMultiplier> xsd:integer </periodMultiplier> [1]
    '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.'

    <period> PeriodEnum </period> [1]
    'A time period, e.g. a day, week, month or year of the stream. If the periodMultiplier value is 0 (zero) then period must contain the value D (day).'

    <dayType> DayTypeEnum </dayType> [0..1]
    '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.'

    <businessDayConvention> BusinessDayConventionEnum </businessDayConvention> [1]
    'The convention for adjusting a date if it would otherwise fall on a day that is not a business day.'
```

Start Group: BusinessCentersOrReference.model [0..1]

Start Choice [1]

- <businessCentersReference> BusinessCentersReference </businessCentersReference> [1]
 '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.'
- <businessCenters> BusinessCenters </businessCenters> [1]

End Choice

End Group: BusinessCentersOrReference.model

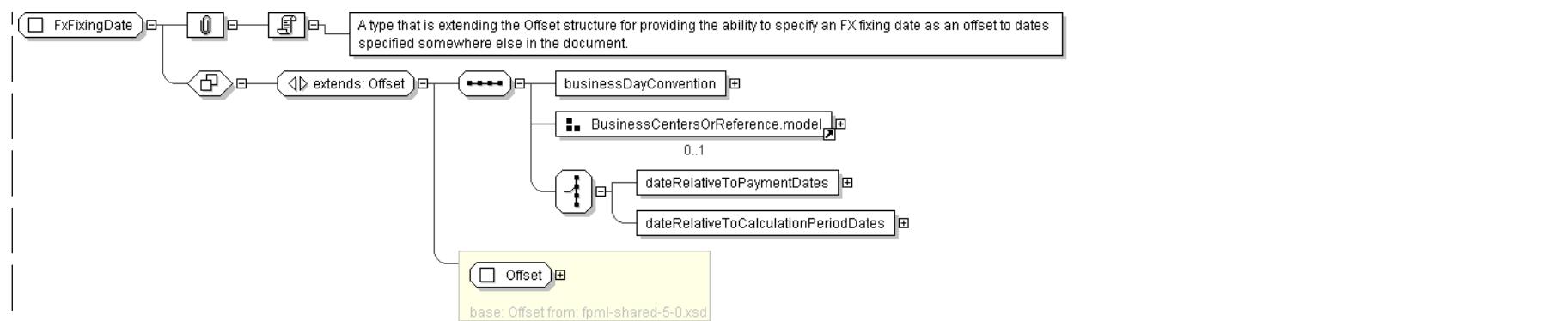
Start Choice [1]

- <dateRelativeToPaymentDates> DateRelativeToPaymentDates </dateRelativeToPaymentDates> [1]
 'The payment date references on which settlements in non-deliverable currency are due and will then have to be converted according to the terms specified through the other parts of the nonDeliverableSettlement structure.'
- <dateRelativeToCalculationPeriodDates> DateRelativeToCalculationPeriodDates </dateRelativeToCalculationPeriodDates> [1]
 'The calculation period references on which settlements in non-deliverable currency are due and will then have to be converted according to the terms specified through the other parts of the nonDeliverableSettlement structure. Implemented for Brazilian-CDI swaps where it will refer to the termination date of the appropriate leg.'

End Choice

</...>

**Diagram**

**Schema Component Representation**

```

<xsd:complexType name="FxFixingDate">
  <xsd:complexContent>
    <xsd:extension base=" Offset ">
      <xsd:sequence>
        <xsd:element name="businessDayConvention" type=" BusinessDayConventionEnum "/>
        <xsd:group ref=" BusinessCentersOrReference.model " minOccurs="0"/>
        <xsd:choice>
          <xsd:element name="dateRelativeToPaymentDates" type=" DateRelativeToPaymentDates "/>
          <xsd:element name="dateRelativeToCalculationPeriodDates"
            type=" DateRelativeToCalculationPeriodDates "/>
        </xsd:choice>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
  
```

top

**Complex Type: FxLinkedNotionalAmount**

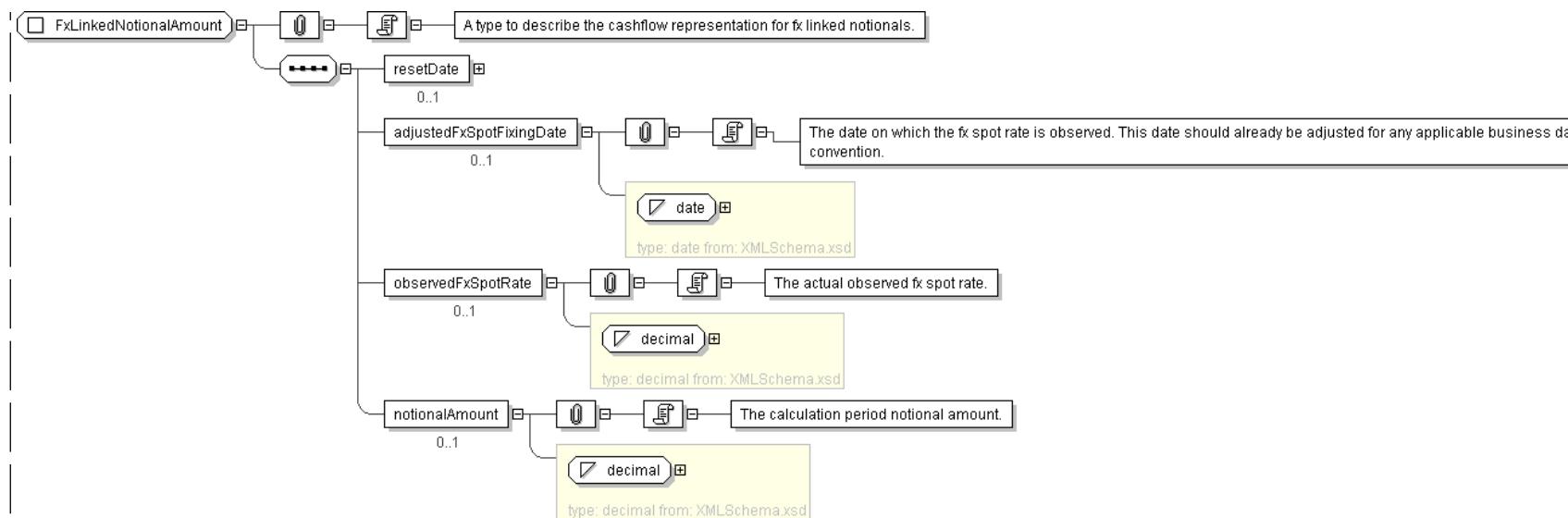
<b>Super-types:</b>	None
<b>Sub-types:</b>	None
<b>Name</b>	FxLinkedNotionalAmount
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">CalculationPeriod</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type to describe the cashflow representation for fx linked notionals.

**XML Instance Representation**

```

<...>
<resetDate> xsd:date </resetDate> [0..1]
<adjustedFxSpotFixingDate> xsd:date </adjustedFxSpotFixingDate> [0..1]
  'The date on which the fx spot rate is observed. This date should already be adjusted for
any applicable business day convention.'
<observedFxSpotRate> xsd:decimal </observedFxSpotRate> [0..1]
  'The actual observed fx spot rate.'
<notionalAmount> xsd:decimal </notionalAmount> [0..1]
  'The calculation period notional amount.'
</...>
  
```

**Diagram**

**Schema Component Representation**

```

<xsd:complexType name="FxLinkedNotionalAmount">
  <xsd:sequence>
    <xsd:element name="resetDate" type="xsd:date" minOccurs="0"/>
    <xsd:element name="adjustedFxSpotFixingDate" type="xsd:date" minOccurs="0"/>
    <xsd:element name="observedFxSpotRate" type="xsd:decimal" minOccurs="0"/>
    <xsd:element name="notionalAmount" type="xsd:decimal" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>
  
```

[top](#)**Complex Type: FxLinkedNotionalSchedule**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None
<b>Name</b>	FxLinkedNotionalSchedule
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">Calculation</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type to describe a notional 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.

**XML Instance Representation**

```

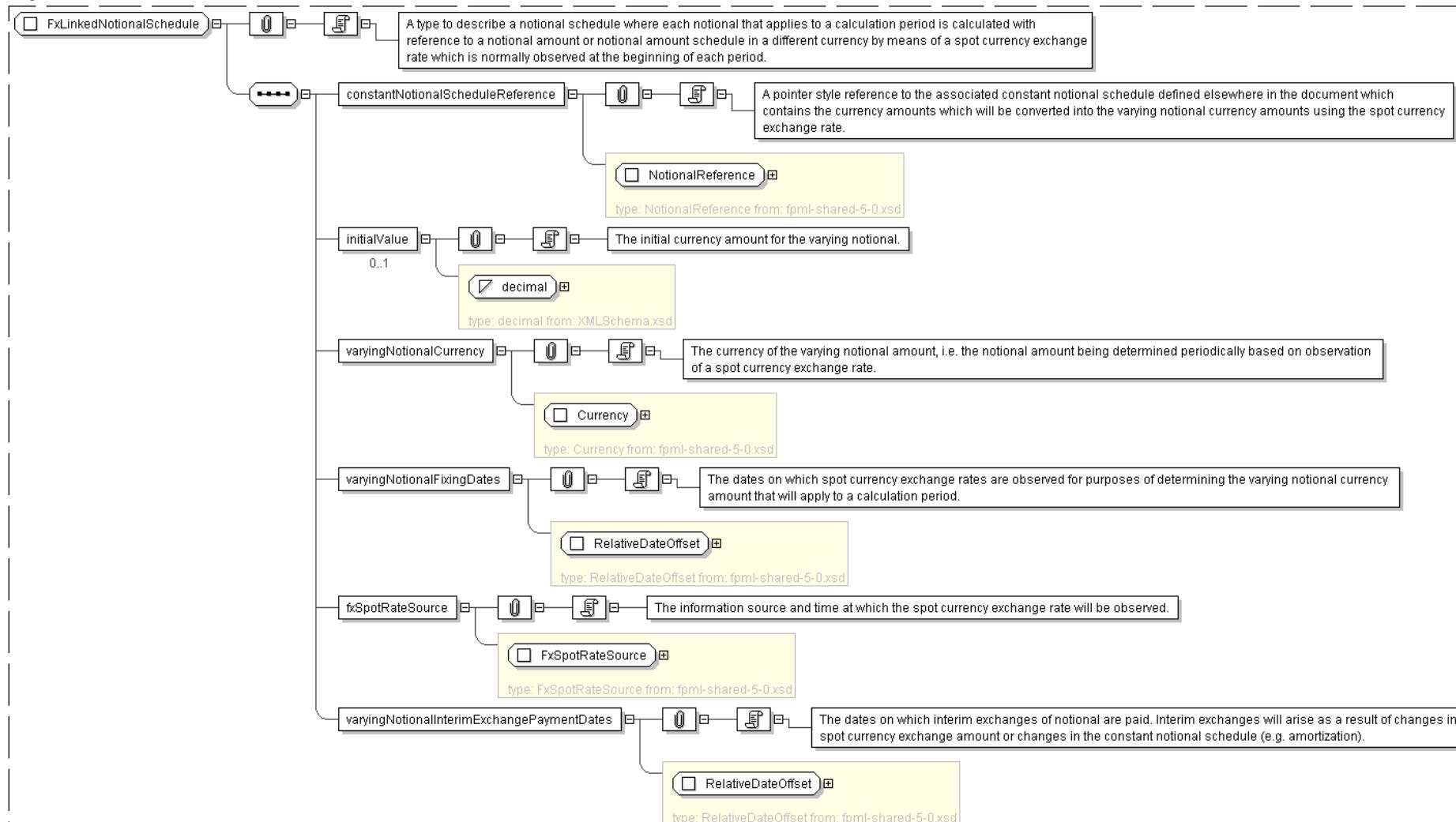
<...>
<constantNotionalScheduleReference> NotionalReference </constantNotionalScheduleReference> [1]
  'A pointer style reference to the associated constant notional schedule defined elsewhere
  in the document which contains the currency amounts which will be converted into the
  varying notional currency amounts using the spot currency exchange rate.'
<initialValue> xsd:decimal </initialValue> [0..1]
  'The initial currency amount for the varying notional.'
<varyingNotionalCurrency> Currency </varyingNotionalCurrency> [1]
  '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> RelativeDateOffset </varyingNotionalFixingDates> [1]
  'The dates on which spot currency exchange rates are observed for purposes of determining
  the varying notional amount.' 
  
```

*the varying notional currency amount that will apply to a calculation period.'*

```
<fxSpotRateSource> FxSpotRateSource </fxSpotRateSource> [1]
'The information source and time at which the spot currency exchange rate will be observed.'
```

```
<varyingNotionalInterimExchangePaymentDates> RelativeDateOffset
</varyingNotionalInterimExchangePaymentDates> [1]
'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).'
```

&lt;/...&gt;

**Diagram****Schema Component Representation**

```

<xsd:complexType name="FxLinkedNotionalSchedule">
  <xsd:sequence>
    <xsd:element name="constantNotionalScheduleReference" type=" NotionalReference " />
    <xsd:element name="initialValue" type=" xsd:decimal " minOccurs="0"/>
    <xsd:element name="varyingNotionalCurrency" type=" Currency " />
  
```

```

<xsd:element name="varyingNotionalFixingDates" type=" RelativeDateOffset " />
<xsd:element name="fxSpotRateSource" type=" FxSpotRateSource " />
<xsd:element name="varyingNotionalInterimExchangePaymentDates" type=" RelativeDateOffset " />
</xsd:sequence>
</xsd:complexType>

```

## Complex Type: InflationRateCalculation

<b>Super-types:</b>	<a href="#">FloatingRateCalculation</a> < <b>InflationRateCalculation</b> (by extension)
<b>Sub-types:</b>	None

<b>Name</b>	InflationRateCalculation
<b>Used by (from the same schema document)</b>	Element <a href="#">inflationRateCalculation</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type defining the components specifying an Inflation Rate Calculation

### XML Instance Representation

```

<...
  id=" xsd:ID [0..1]">
    <floatingRateIndex> FloatingRateIndex </floatingRateIndex> [1]
    <indexTenor> Period </indexTenor> [0..1]
      'The ISDA Designated Maturity, i.e. the tenor of the floating rate.'

    <floatingRateMultiplierSchedule> Schedule </floatingRateMultiplierSchedule> [0..1]
      'A rate multiplier or multiplier schedule to apply to the floating rate. A multiplier
      schedule is expressed as explicit multipliers and dates. In the case of a schedule, the
      step dates may be subject to adjustment in accordance with any adjustments specified in
      the calculationPeriodDatesAdjustments. The multiplier can be a positive or negative
      decimal. This element should only be included if the multiplier is not equal to 1 (one) for
      the term of the stream.'

    <spreadSchedule> SpreadSchedule </spreadSchedule> [0..*]
      '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> RateTreatmentEnum </rateTreatment> [0..1]
      '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> StrikeSchedule </capRateSchedule> [0..*]
      '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 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> StrikeSchedule </floorRateSchedule> [0..*]
      '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.'

    <initialRate> xsd:decimal </initialRate> [0..1]

```

'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> Rounding </finalRateRounding> [0..1]

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

<averagingMethod> AveragingMethodEnum </averagingMethod> [0..1]

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

<negativeInterestRateTreatment> NegativeInterestRateTreatmentEnum

</negativeInterestRateTreatment> [0..1]

'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).'

<inflationLag> Offset </inflationLag> [1]

'an offsetting period from the payment date which determines the reference period for which the inflation index is observed.'

<indexSource> RateSourcePage </indexSource> [1]

'The reference source such as Reuters or Bloomberg.'

<mainPublication> MainPublication </mainPublication> [0..1]

'The current main publication source such as relevant web site or a government body.'

<interpolationMethod> InterpolationMethod </interpolationMethod> [1]

'The method used when calculating the Inflation Index Level from multiple points - the most common is Linear.'

<initialIndexLevel> xsd:decimal </initialIndexLevel> [0..1]

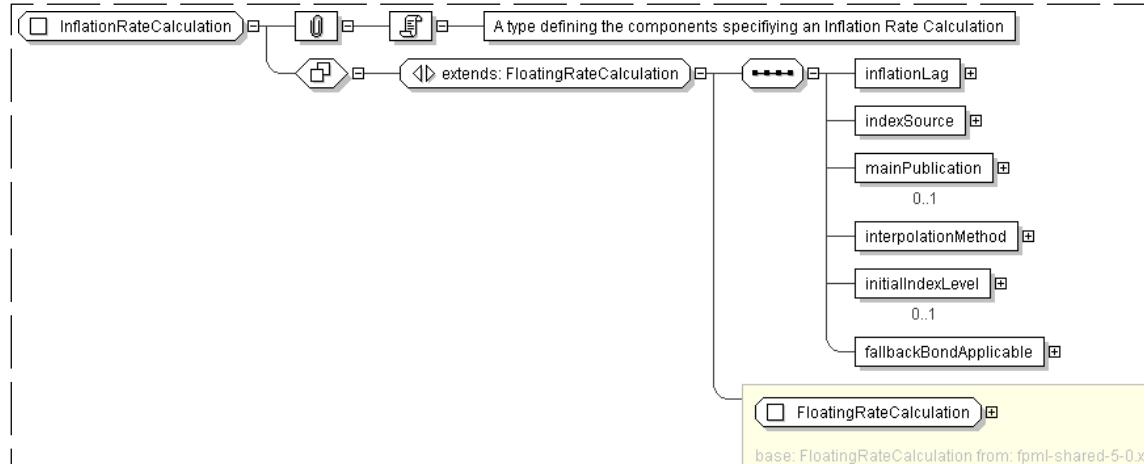
'initial known index level for the first calculation period.'

<fallbackBondApplicable> xsd:boolean </fallbackBondApplicable> [1]

'The applicability of a fallback bond as defined in the 2006 ISDA Inflation Derivatives Definitions, sections 1.3 and 1.8. Omission of this element implies a value of true.'

<...>

#### Diagram



## Schema Component Representation

```

<xsd:complexType name="InflationRateCalculation">
  <xsd:complexContent>
    <xsd:extension base=" FloatingRateCalculation ">
      <xsd:sequence>
        <xsd:element name="inflationLag" type=" Offset " />
        <xsd:element name="indexSource" type=" RateSourcePage " />
        <xsd:element name="mainPublication" type=" MainPublication " minOccurs="0"/>
        <xsd:element name="interpolationMethod" type=" InterpolationMethod " />
        <xsd:element name="initialIndexLevel" type=" xsd:decimal " minOccurs="0"/>
        <xsd:element name="fallbackBondApplicable" type=" xsd:boolean " />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

top

Complex Type: **InterestRateStream**

Super-types:	<a href="#">Leg</a> < <b>InterestRateStream</b> (by extension)
Sub-types:	None

Name	InterestRateStream
Used by (from the same schema document)	Complex Type <a href="#">CapFloor</a> , Complex Type <a href="#">Swap</a>
Abstract	no
Documentation	A type defining the components specifying an interest rate stream, including both a parametric and cashflow representation for the stream of payments.

## XML Instance Representation

```

<...>
<id=" xsd:ID [0..1]">
<payerPartyReference> PartyReference </payerPartyReference> [1]
  'A reference to the party responsible for making the payments defined by this structure.'

<payerAccountReference> AccountReference </payerAccountReference> [0..1]
  'A reference to the account responsible for making the payments defined by this structure.'

<receiverPartyReference> PartyReference </receiverPartyReference> [1]
  'A reference to the party that receives the payments corresponding to this structure.'

<receiverAccountReference> AccountReference </receiverAccountReference> [0..1]
  'A reference to the account that receives the payments corresponding to this structure.'

<calculationPeriodDates> CalculationPeriodDates </calculationPeriodDates> [1]
  'The calculation periods dates schedule.'

<paymentDates> PaymentDates </paymentDates> [1]
  'The payment dates schedule.'

<resetDates> ResetDates </resetDates> [0..1]
  'The reset dates schedule. The reset dates schedule only applies for a floating rate stream.'

<calculationPeriodAmount> CalculationPeriodAmount </calculationPeriodAmount> [1]
  'The calculation period amount parameters.'

<stubCalculationPeriodAmount> StubCalculationPeriodAmount </stubCalculationPeriodAmount> [0..1]
  '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> PrincipalExchanges </principalExchanges> [0..1]
  'The true/false flags indicating whether initial, intermediate or final exchanges of
  '

```

*principal should occur.*

<cashflows> Cashflows </cashflows> [0..1]

'The cashflows representation of the swap stream.'

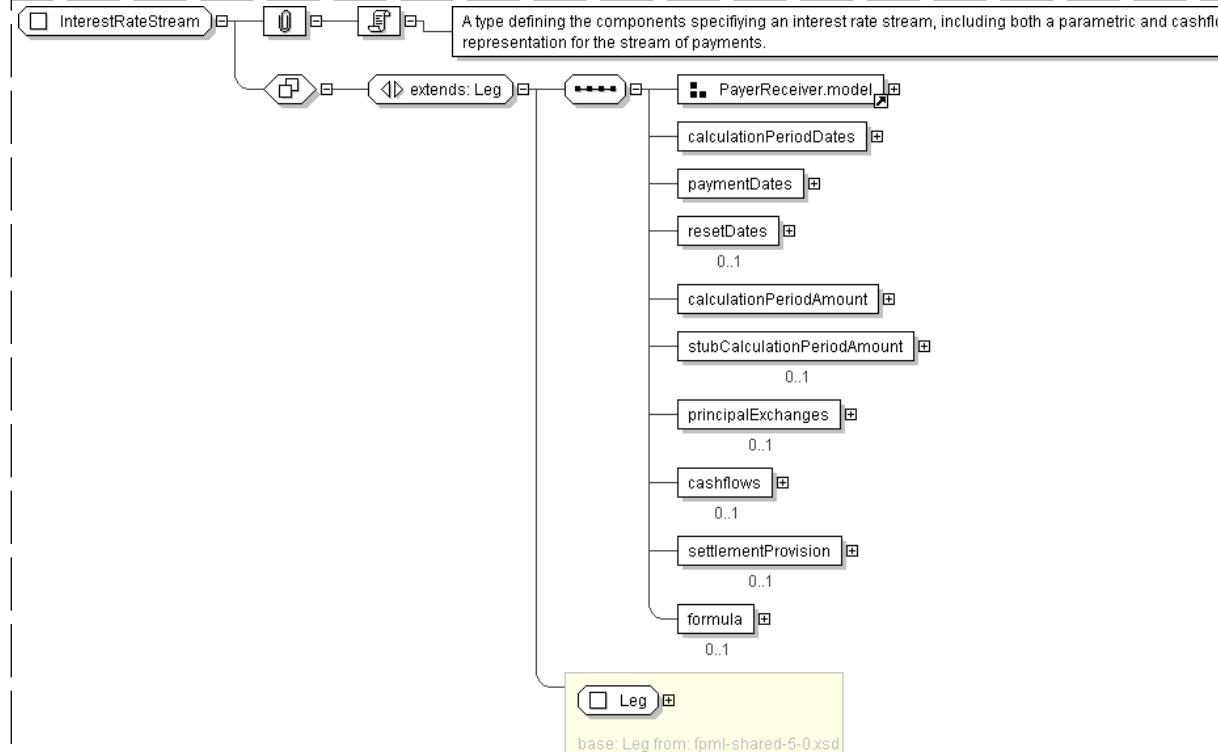
<settlementProvision> SettlementProvision </settlementProvision> [0..1]

'A provision that allows the specification of settlement terms, occurring when the settlement currency is different to the notional currency of the trade.'

<formula> Formula </formula> [0..1]

'An interest rate derivative formula.'

&lt;/...&gt;

**Diagram****Schema Component Representation**

```

<xsd:complexType name="InterestRateStream">
  <xsd:complexContent>
    <xsd:extension base=" Leg ">
      <xsd:sequence>
        <xsd:group ref=" PayerReceiver.model " />
        <xsd:element name="calculationPeriodDates" type=" CalculationPeriodDates " />
        <xsd:element name="paymentDates" type=" PaymentDates " />
        <xsd:element name="resetDates" type=" ResetDates " minOccurs="0"/>
        <xsd:element name="calculationPeriodAmount" type=" CalculationPeriodAmount " />
        <xsd:element name="stubCalculationPeriodAmount" type=" StubCalculationPeriodAmount " minOccurs="0"/>
        <xsd:element name="principalExchanges" type=" PrincipalExchanges " minOccurs="0"/>
        <xsd:element name="cashflows" type=" Cashflows " minOccurs="0"/>
        <xsd:element name="settlementProvision" type=" SettlementProvision " minOccurs="0"/>
        <xsd:element name="formula" type=" Formula " minOccurs="0"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
  
```

```
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>
```

## Complex Type: InterestRateStreamReference

Super-types:

[Reference](#) < **InterestRateStreamReference** (by extension)

Sub-types:

None

Name	InterestRateStreamReference
------	-----------------------------

Used by (from the same schema document)	Complex Type <a href="#">FinalCalculationPeriodDateAdjustment</a>
---	---

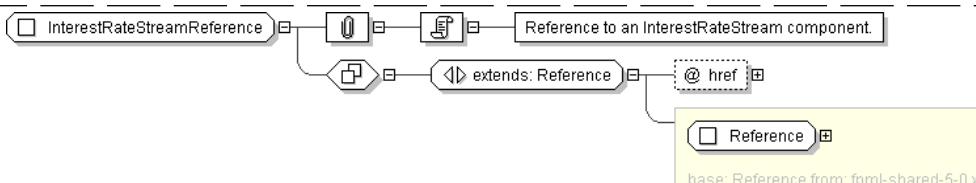
Abstract	no
----------	----

Documentation	Reference to an InterestRateStream component.
---------------	---

### XML Instance Representation

```
<...>
  href=" xsd:IDREF [1]" />
```

### Diagram



### Schema Component Representation

```
<xsd:complexType name="InterestRateStreamReference">
  <xsd:complexContent>
    <xsd:extension base=" Reference ">
      <xsd:attribute name="href" type=" xsd:IDREF " use="required" reference="InterestRateStream"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

## Complex Type: MandatoryEarlyTermination

Super-types:

None

Sub-types:

None

Name	MandatoryEarlyTermination
------	---------------------------

Used by (from the same schema document)	Model Group <a href="#">MandatoryEarlyTermination.model</a> , Model Group <a href="#">MandatoryEarlyTermination.model</a>
---	---

Abstract	no
----------	----

Documentation	A type to define an early termination provision for which exercise is mandatory.
---------------	--

### XML Instance Representation

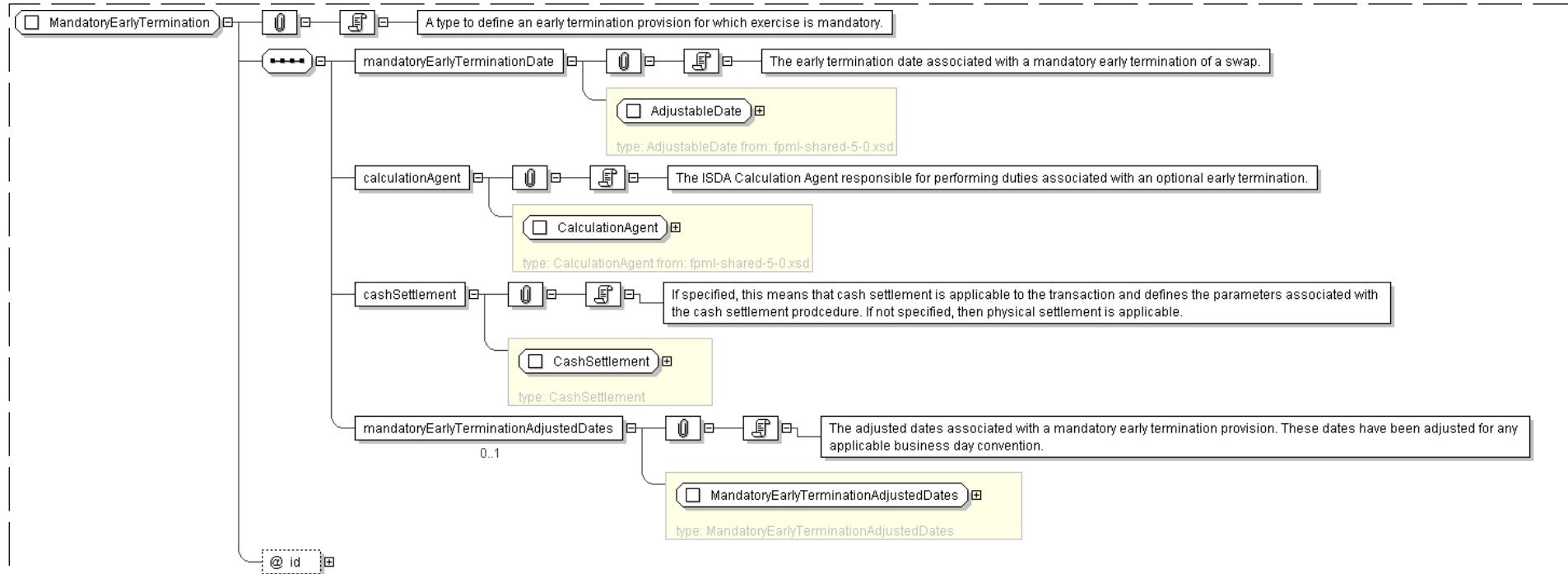
```
<...>
  id=" xsd:ID [0..1]">
  <mandatoryEarlyTerminationDate> AdjustableDate </mandatoryEarlyTerminationDate> [1]
  'The early termination date associated with a mandatory early termination of a swap.'
  <calculationAgent> CalculationAgent </calculationAgent> [1]
  'The ISDA Calculation Agent responsible for performing duties associated with an optional
  early termination.'
  <cashSettlement> CashSettlement </cashSettlement> [1]
```

'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> **MandatoryEarlyTerminationAdjustedDates**  
 </mandatoryEarlyTerminationAdjustedDates> [0..1]

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

&lt;/...&gt;

**Diagram****Schema Component Representation**

```

<xsd:complexType name="MandatoryEarlyTermination">
  <xsd:sequence>
    <xsd:element name="mandatoryEarlyTerminationDate" type=" AdjustableDate "/>
    <xsd:element name="calculationAgent" type=" CalculationAgent "/>
    <xsd:element name="cashSettlement" type=" CashSettlement "/>
    <xsd:element name="mandatoryEarlyTerminationAdjustedDates" type=" MandatoryEarlyTerminationAdjustedDates " minOccurs="0"/>
  </xsd:sequence>
  <xsd:attribute name="id" type=" xsd:ID "/>
</xsd:complexType>
  
```

top

**Complex Type: MandatoryEarlyTerminationAdjustedDates**

Super-types:	None
Sub-types:	None

Name	MandatoryEarlyTerminationAdjustedDates
Used by (from the same schema document)	Complex Type <b>MandatoryEarlyTermination</b>

<b>Abstract</b>	no
<b>Documentation</b>	A type defining the adjusted dates associated with a mandatory early termination provision.

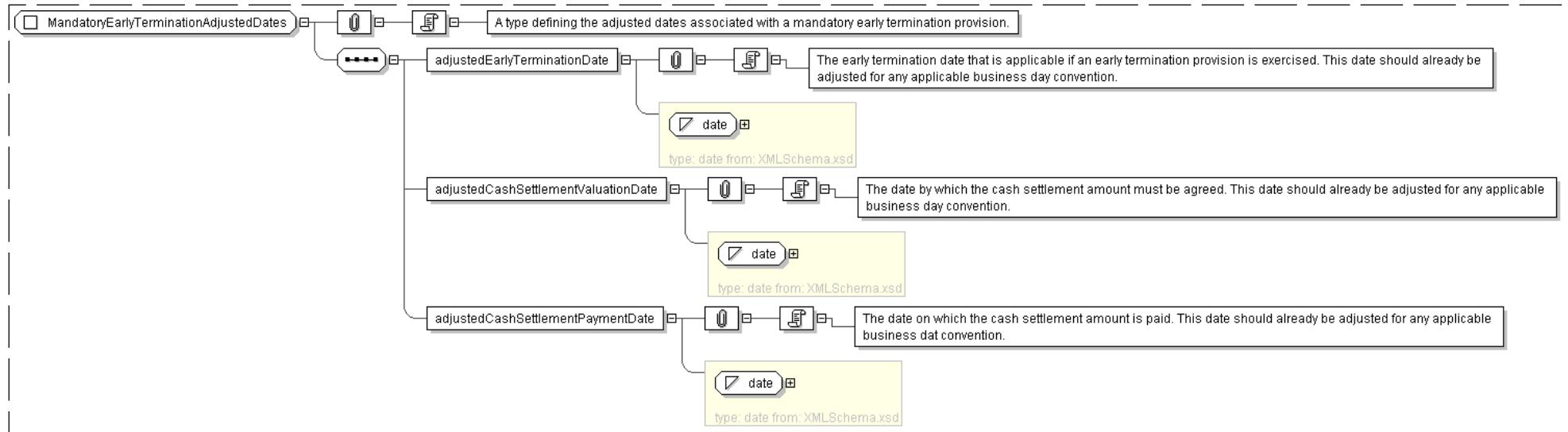
**XML Instance Representation**

```
<....>
<adjustedEarlyTerminationDate> xsd:date </adjustedEarlyTerminationDate> [1]
'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> xsd:date </adjustedCashSettlementValuationDate> [1]
'The date by which the cash settlement amount must be agreed. This date should already
be adjusted for any applicable business day convention.'

<adjustedCashSettlementPaymentDate> xsd:date </adjustedCashSettlementPaymentDate> [1]
'The date on which the cash settlement amount is paid. This date should already be adjusted
for any applicable business dat convention.'

</....>
```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="MandatoryEarlyTerminationAdjustedDates">
  <xsd:sequence>
    <xsd:element name="adjustedEarlyTerminationDate" type="xsd:date" />
    <xsd:element name="adjustedCashSettlementValuationDate" type="xsd:date" />
    <xsd:element name="adjustedCashSettlementPaymentDate" type="xsd:date" />
  </xsd:sequence>
</xsd:complexType>

```

top

**Complex Type: NonDeliverableSettlement**

Super-types:	None
Sub-types:	None

<b>Name</b>	NonDeliverableSettlement
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">SettlementProvision</a>
<b>Abstract</b>	no

**Documentation**

A type defining the parameters used when the reference currency of the swapStream is non-deliverable.

**XML Instance Representation**

```
<...>
<referenceCurrency> Currency </referenceCurrency> [1]
'The currency in which the swap stream is denominated in.'
```

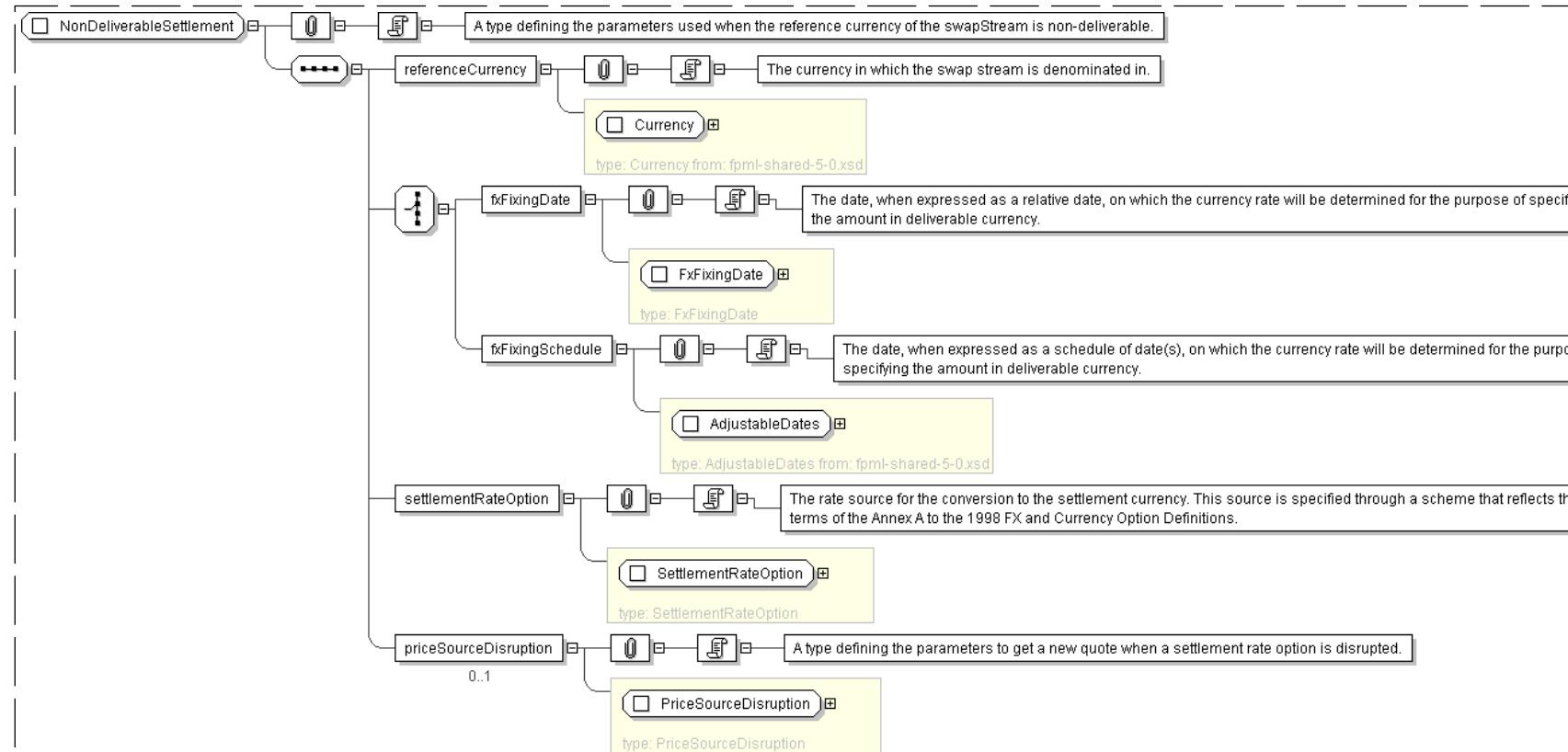
```
Start Choice [1]
<fxFixingDate> FxFixingDate </fxFixingDate> [1]
'The date, when expressed as a relative date, on which the currency rate will be determined
for the purpose of specifying the amount in deliverable currency.'
```

```
<fxFixingSchedule> AdjustableDates </fxFixingSchedule> [1]
'The date, when expressed as a schedule of date(s), on which the currency rate will
be determined for the purpose of specifying the amount in deliverable currency.'
```

```
End Choice
<settlementRateOption> SettlementRateOption </settlementRateOption> [1]
'The rate source for the conversion to the settlement currency. This source is
specified through a scheme that reflects the terms of the Annex A to the 1998 FX and
Currency Option Definitions.'
```

```
<priceSourceDisruption> PriceSourceDisruption </priceSourceDisruption> [0..1]
'A type defining the parameters to get a new quote when a settlement rate option is disrupted.'
```

```
</...>
```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="NonDeliverableSettlement">
  <xsd:sequence>
    <xsd:element name="referenceCurrency" type="Currency" />
    <xsd:choice>
      <xsd:element name="fxFixingDate" type="FxFixingDate" />
      <xsd:element name="fxFixingSchedule" type="AdjustableDates" />
    </xsd:choice>
    <xsd:element name="settlementRateOption" type="SettlementRateOption" />
    <xsd:element name="priceSourceDisruption" type="PriceSourceDisruption" minOccurs="0" />
  </xsd:sequence>
</xsd:complexType>

```

top

## Complex Type: Notional

Super-types:	None
Sub-types:	None
Name	Notional
Used by (from the same schema document)	Complex Type <a href="#">Calculation</a>
Abstract	no
Documentation	An type defining the notional amount or notional amount schedule associated with a swap stream. The notional schedule will be captured 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.

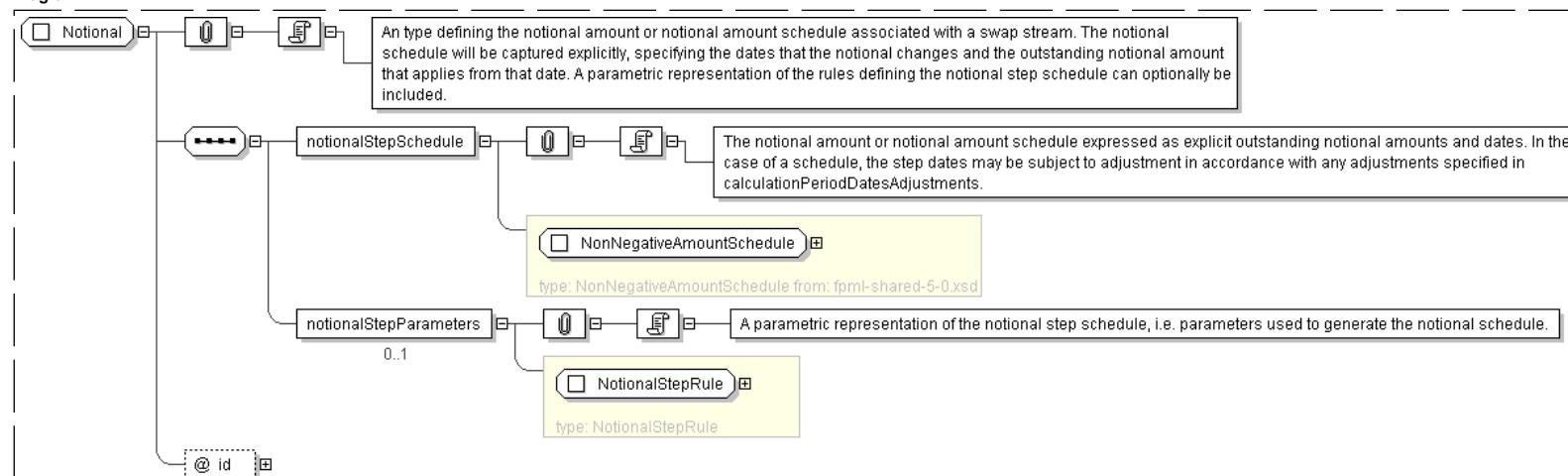
### XML Instance Representation

```

<...>
<id="xsd:ID [0..1]">
  <notionalStepSchedule> NonNegativeAmountSchedule </notionalStepSchedule> [1]
  '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> NotionalStepRule </notionalStepParameters> [0..1]
  'A parametric representation of the notional step schedule, i.e. parameters used to
  generate the notional schedule.'
</...>

```

### Diagram



### Schema Component Representation

```

<xsd:complexType name="Notional">
  <xsd:sequence>
    <xsd:element name="notionalStepSchedule" type="NonNegativeAmountSchedule" />
    <xsd:element name="notionalStepParameters" type="NotionalStepRule" minOccurs="0" />
  </xsd:sequence>
  <xsd:attribute name="id" type="xsd:ID" />
</xsd:complexType>

```

## Complex Type: NotionalStepRule

Super-types:	None
Sub-types:	None
Name	NotionalStepRule
Used by (from the same schema document)	Complex Type <a href="#">Notional</a>
Abstract	no
Documentation	A type 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.

### XML Instance Representation

```

<....>
  <calculationPeriodDatesReference> CalculationPeriodDatesReference
  </calculationPeriodDatesReference> [1]
  'A pointer style reference to the associated calculation period dates component
  defined elsewhere in the document.'

  <stepFrequency> Period </stepFrequency> [1]
  'The frequency at which the step changes occur. This frequency must be a multiple of the
  stream calculation period frequency.'

  <firstNotionalStepDate> xsd:date </firstNotionalStepDate> [1]
  'Effective date of the first change in notional (i.e. a calculation period start date).'

  <lastNotionalStepDate> xsd:date </lastNotionalStepDate> [1]
  'Effective date of the last change in notional (i.e. a calculation period start date).'

Start Choice [1]
  <notionalStepAmount> xsd:decimal </notionalStepAmount> [1]
  'The explicit amount that the notional changes on each step date. This can be a positive
  or negative amount.'

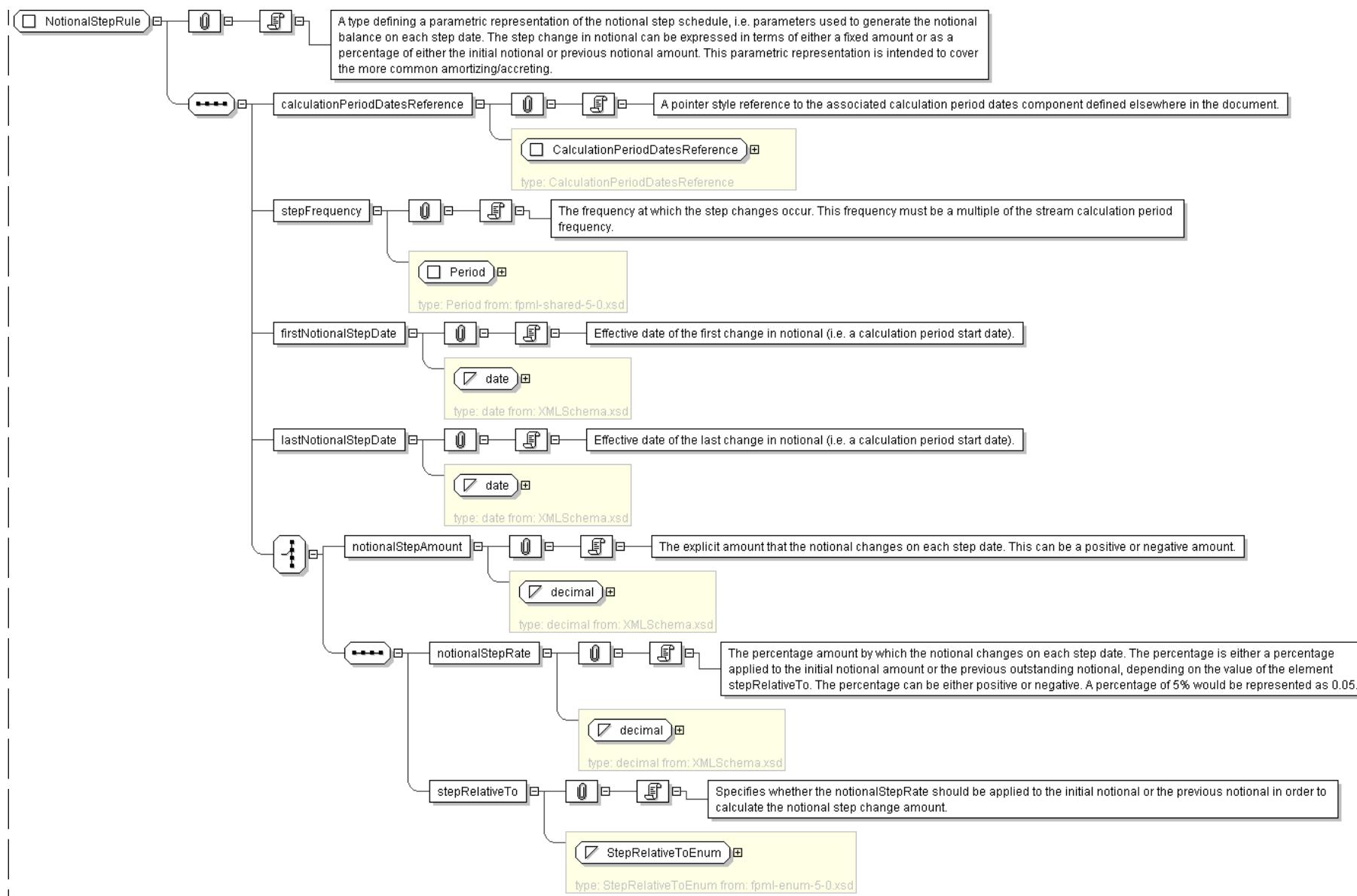
  <notionalStepRate> xsd:decimal </notionalStepRate> [1]
  'The percentage amount by which the notional changes on each step date. The percentage
  is either a percentage applied to the initial notional amount or the previous
  outstanding notional, depending on the value of the element stepRelativeTo. The percentage
  can be either positive or negative. A percentage of 5% would be represented as 0.05.'

  <stepRelativeTo> StepRelativeToEnum </stepRelativeTo> [1]
  'Specifies whether the notionalStepRate should be applied to the initial notional or
  the previous notional in order to calculate the notional step change amount.'

End Choice
</....>

```

### Diagram



#### Schema Component Representation

```

<xsd:complexType name="NotionalStepRule">
  <xsd:sequence>
    <xsd:element name="calculationPeriodDatesReference" type=" CalculationPeriodDatesReference " />
    <xsd:element name="stepFrequency" type=" Period " />
    <xsd:element name="firstNotionalStepDate" type=" xsd:date " />
    <xsd:element name="lastNotionalStepDate" type=" xsd:date " />
    <xsd:choice>
      <xsd:element name="notionalStepAmount" type=" xsd:decimal " />
      <xsd:sequence>
        <xsd:element name="notionalStepRate" type=" xsd:decimal " />
        <xsd:element name="stepRelativeTo" type=" StepRelativeToEnum " />
      </xsd:sequence>
    </xsd:choice>
  </xsd:sequence>
</xsd:complexType>

```

**Complex Type: OptionalEarlyTermination**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None

<b>Name</b>	OptionalEarlyTermination
<b>Used by (from the same schema document)</b>	Model Group <a href="#">OptionalEarlyTermination.model</a> , Model Group <a href="#">OptionalEarlyTermination.model</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type defining an early termination provision where either or both parties have the right to exercise.

**XML Instance Representation**

```
<...>
  <singlePartyOption> SinglePartyOption </singlePartyOption> [0..1]
    'If optional early termination is not available to both parties then this component
    specifies the buyer and seller of the option.'

    <exercise> ... </exercise> [1]
    <exerciseNotice> ExerciseNotice </exerciseNotice> [0..*]
      'Definition of the party to whom notice of exercise should be given.'

    <followUpConfirmation> xsd:boolean </followUpConfirmation> [0..1]
      '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.'

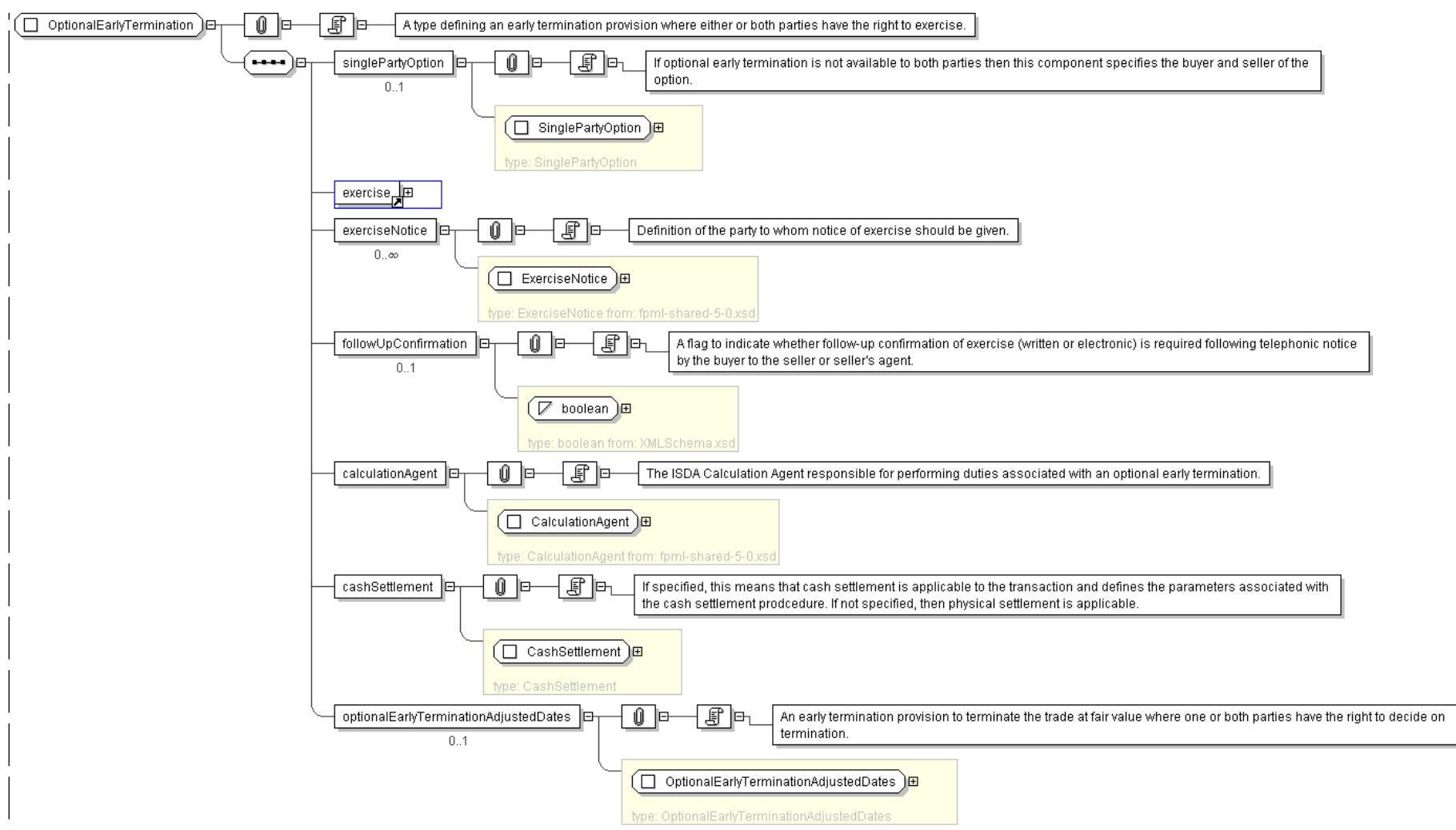
    <calculationAgent> CalculationAgent </calculationAgent> [1]
      'The ISDA Calculation Agent responsible for performing duties associated with an optional
      early termination.'

    <cashSettlement> CashSettlement </cashSettlement> [1]
      '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> OptionalEarlyTerminationAdjustedDates
    </optionalEarlyTerminationAdjustedDates> [0..1]
      'An early termination provision to terminate the trade at fair value where one or both
      parties have the right to decide on termination.'

  </...>
```

**Diagram**



#### Schema Component Representation

```

<xsd:complexType name="OptionalEarlyTermination">
  <xsd:sequence>
    <xsd;element name="singlePartyOption" type="#type-SinglePartyOption" minOccurs="0"/>
    <xsd;element ref="exercise"/>
    <xsd;element name="exerciseNotice" type="#type-ExerciseNotice" minOccurs="0" maxOccurs="unbounded"/>
    <xsd;element name="followUpConfirmation" type="xsd:boolean" minOccurs="0"/>
    <xsd;element name="calculationAgent" type="#type-CalculationAgent"/>
    <xsd;element name="cashSettlement" type="#type-CashSettlement"/>
    <xsd;element name="optionalEarlyTerminationAdjusteddates" type="#type-OptionalEarlyTerminationAdjustedDates" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>

```

top

#### Complex Type: `OptionalEarlyTerminationAdjustedDates`

Super-types:

None

Sub-types:

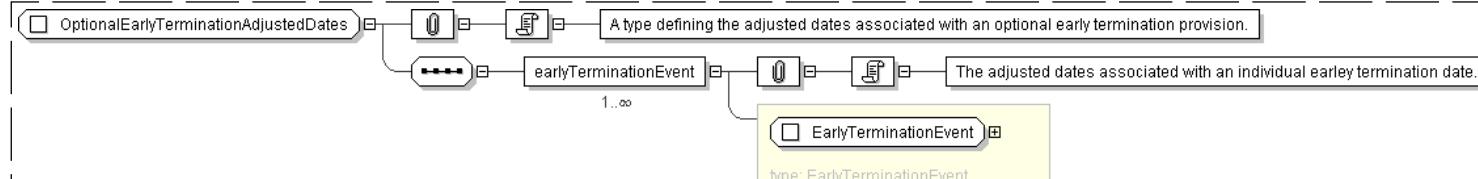
None

Name	OptionalEarlyTerminationAdjustedDates
Used by (from the same schema document)	Complex Type <a href="#">OptionalEarlyTermination</a>
Abstract	no
Documentation	A type defining the adjusted dates associated with an optional early termination provision.

## XML Instance Representation

```
<...>
<earlyTerminationEvent> EarlyTerminationEvent </earlyTerminationEvent> [1..*]
'The adjusted dates associated with an individual earley termination date.'
```

## Diagram



## Schema Component Representation

```

<xsd:complexType name="OptionalEarlyTerminationAdjustedDates">
  <xsd:sequence>
    <xsd:element name="earlyTerminationEvent" type=" EarlyTerminationEvent " maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>

```

top

Complex Type: [PaymentCalculationPeriod](#)

Super-types:

[PaymentBase](#) < **PaymentCalculationPeriod** (by extension)

Sub-types:

None

Name	PaymentCalculationPeriod
Used by (from the same schema document)	Complex Type <a href="#">Cashflows</a>
Abstract	no
Documentation	A type defining the adjusted payment date and associated calculation period parameters required to calculate the actual or projected payment amount. This type forms part of the cashflow representation of a swap stream.

## XML Instance Representation

```

<...
  id=" xsd:ID [0..1]"
  href=" xsd:IDREF [0..1]
'Attribute that can be used to reference the yield curve used to estimate the discount factor.'
```

```
">
<unadjustedPaymentDate> xsd:date </unadjustedPaymentDate> [0..1]
<adjustedPaymentDate> xsd:date </adjustedPaymentDate> [0..1]
```

'The adjusted payment date. This date should already be adjusted for any applicable business day convention. This component 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 the Cashflows type are adjusted payment dates).'

```

Start Choice [1]
<calculationPeriod> CalculationPeriod </calculationPeriod> [1..*]
```

'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 start date. An FpML document which contains an unordered list of calucarion periods is

still regarded as a conformant document.'

<fixedPaymentAmount> xsd:decimal </fixedPaymentAmount> [1]

'A known fixed payment amount.'

End Choice

<discountFactor> xsd:decimal </discountFactor> [0..1]

'A decimal value representing the discount factor used to calculate the present value of cash flow.'

<forecastPaymentAmount> Money </forecastPaymentAmount> [0..1]

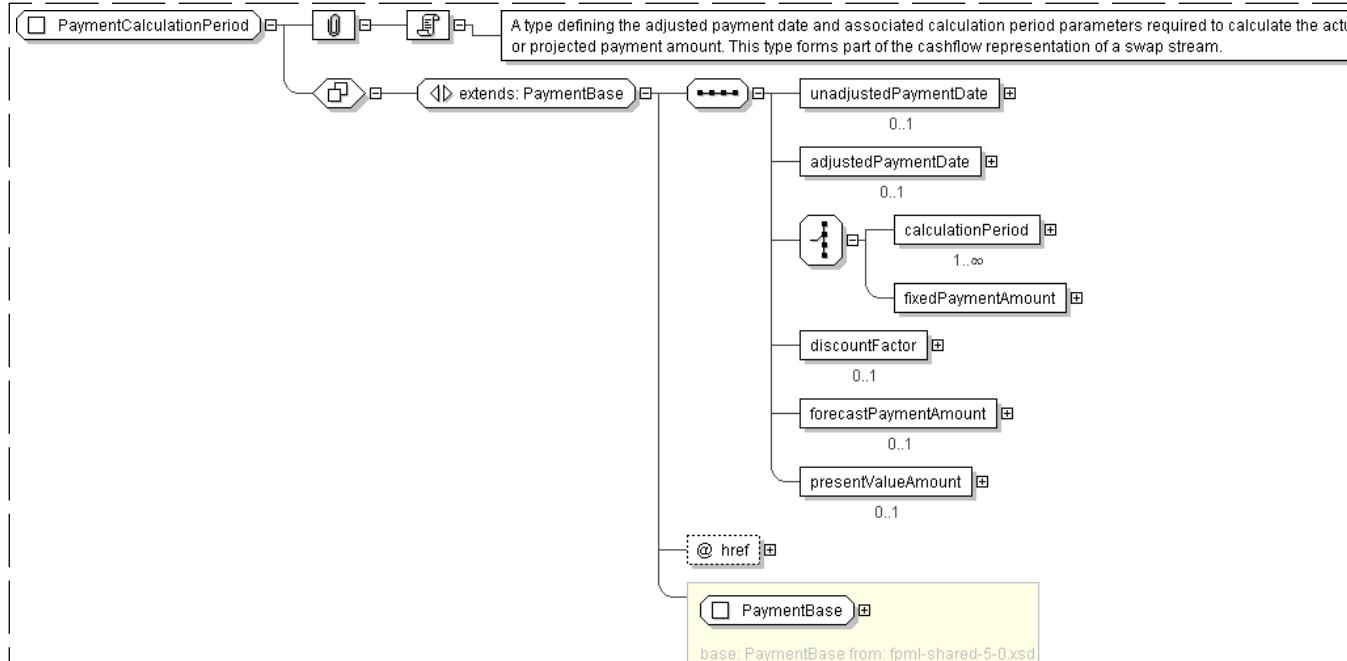
'A monetary amount representing the forecast of the future value of the payment.'

<presentValueAmount> Money </presentValueAmount> [0..1]

'A monetary amount representing the present value of the forecast payment.'

</...>

#### Diagram



#### Schema Component Representation

```

<xsd:complexType name="PaymentCalculationPeriod">
  <xsd:complexContent>
    <xsd:extension base="#PaymentBase">
      <xsd:sequence>
        <xsd:element name="unadjustedPaymentDate" type="xsd:date" minOccurs="0"/>
        <xsd:element name="adjustedPaymentDate" type="xsd:date" minOccurs="0"/>
        <xsd:choice>
          <xsd:element name="calculationPeriod" type="CalculationPeriod" maxOccurs="unbounded"/>
          <xsd:element name="fixedPaymentAmount" type="xsd:decimal"/>
        </xsd:choice>
        <xsd:element name="discountFactor" type="xsd:decimal" minOccurs="0"/>
        <xsd:element name="forecastPaymentAmount" type="Money" minOccurs="0"/>
        <xsd:element name="presentValueAmount" type="Money" minOccurs="0"/>
      </xsd:sequence>
      <xsd:attribute name="href" type="xsd:IDREF" reference="PricingStructure"/>
    </xsd:extension>
  </xsd:complexContent>

```

**Complex Type: PaymentDates**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None
<b>Name</b>	PaymentDates
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">InterestRateStream</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type defining parameters used to generate the payment dates schedule, including the specification of early or delayed payments. Payment dates are determined relative to the calculation period dates or the reset dates.

**XML Instance Representation**

```
<...>
  id=" xsd:ID [0..1]">
  Start Choice [1]
    <calculationPeriodDatesReference> CalculationPeriodDatesReference </calculationPeriodDatesReference> [1]
    'A pointer style reference to the associated calculation period dates component defined elsewhere in the document.'

    <resetDatesReference> ResetDatesReference </resetDatesReference> [1]
    'A pointer style reference to the associated reset dates component defined elsewhere in the document.'

    <valuationDatesReference> ValuationDatesReference </valuationDatesReference> [1]
    'A pointer style reference to the associated valuation dates component defined elsewhere in the document. Implemented for Brazilian-CDI Swaps where it will refer to the settlementProvision/nonDeliverableSettlement/fxFixingDate structure.'

  End Choice
  <paymentFrequency> Frequency </paymentFrequency> [1]
  '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 the 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. If the payment frequency is of value T (term), the period is defined by the swap\swapStream\calculationPeriodDates\effectiveDate and the swap\swapStream\calculationPeriodDates\terminationDate.'

  <firstPaymentDate> xsd:date </firstPaymentDate> [0..1]
  '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> xsd:date </lastRegularPaymentDate> [0..1]
  '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> PayRelativeTypeEnum </payRelativeTo> [1]
```

'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. The valuation date is applicable for Brazilian-CDI swaps.'

```
<paymentDaysOffset> Offset </paymentDaysOffset> [0..1]
```

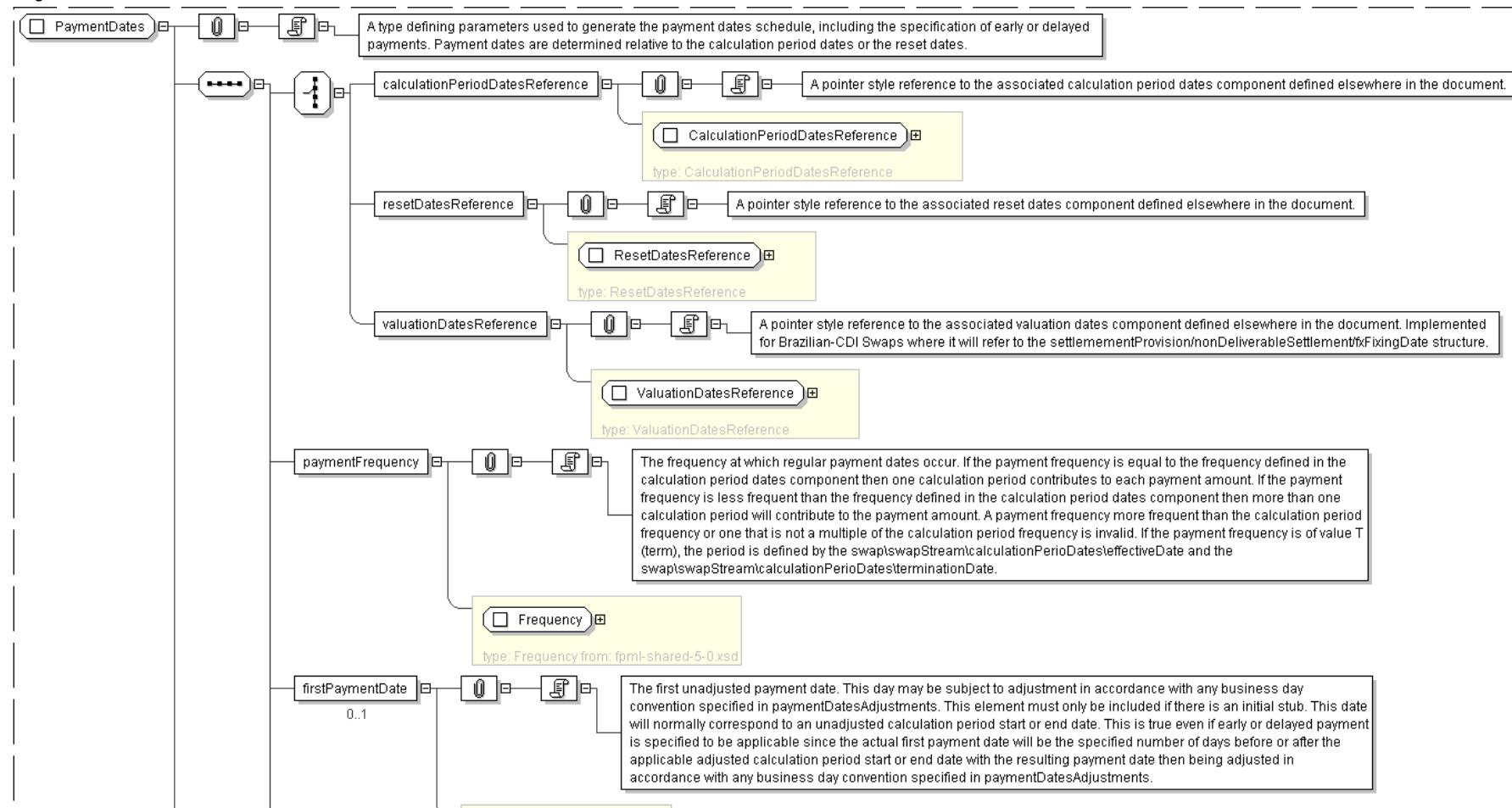
'If early payment or delayed payment is required, specifies the number of days offset that the payment occurs relative to what would otherwise be the unadjusted payment date. The offset can be specified in terms of either calendar or business days. Even in the case of a calendar days offset, the resulting 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.'

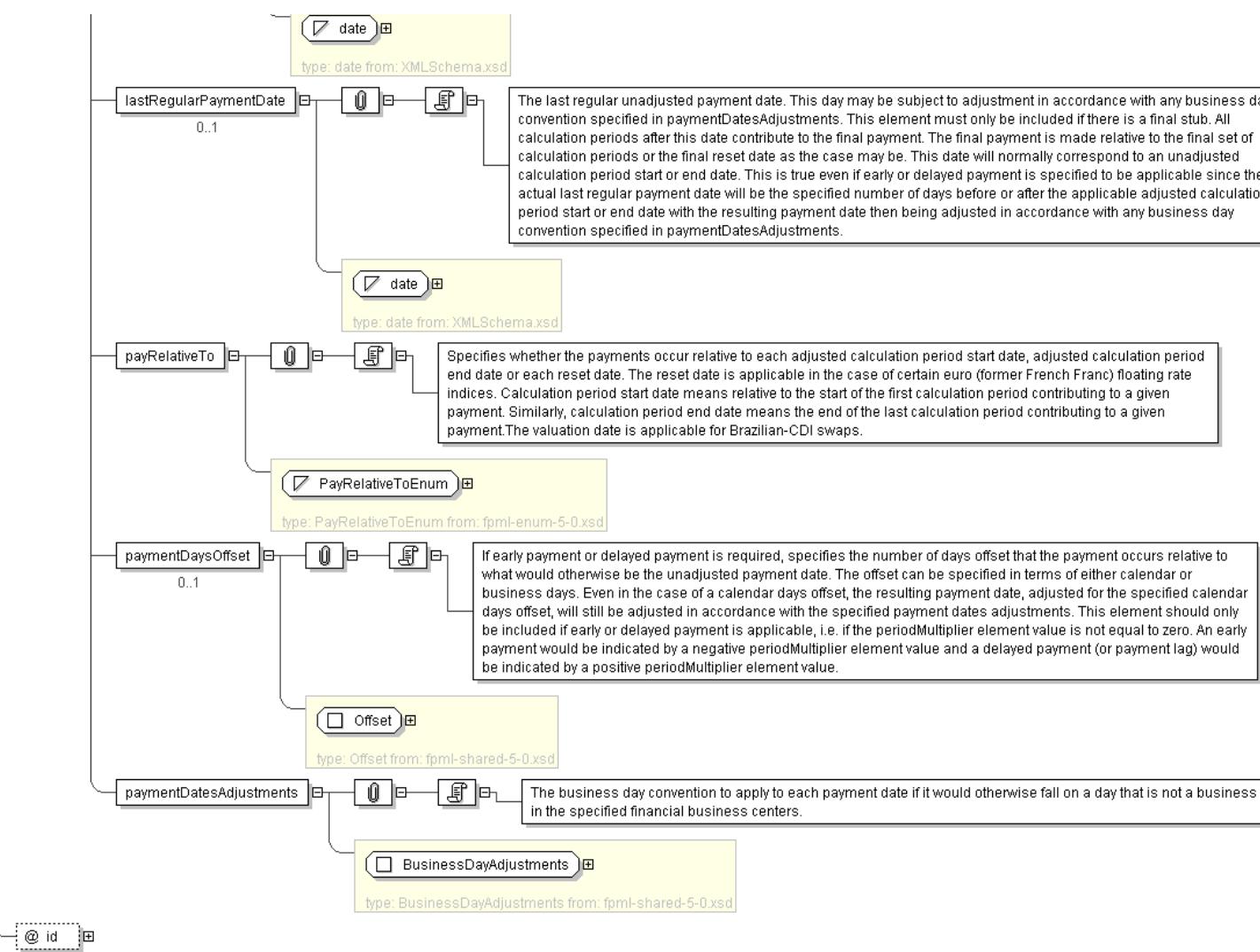
```
<paymentDatesAdjustments> BusinessDayAdjustments </paymentDatesAdjustments> [1]
```

'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.'

```
</...>
```

#### Diagram



**Schema Component Representation**

```

<xsd:complexType name="PaymentDates">
  <xsd:sequence>
    <xsd:choice>
      <xsd:element name="calculationPeriodDatesReference" type=" CalculationPeriodDatesReference " />
      <xsd:element name="resetDatesReference" type=" ResetDatesReference " />
      <xsd:element name="valuationDatesReference" type=" ValuationDatesReference " />
    </xsd:choice>
    <xsd:element name="paymentFrequency" type=" Frequency " />
    <xsd:element name="firstPaymentDate" type=" xsd:date " minOccurs="0"/>
    <xsd:element name="lastRegularPaymentDate" type=" xsd:date " minOccurs="0"/>
    <xsd:element name="payRelativeTo" type=" PayRelativeToEnum " />
    <xsd:element name="paymentDaysOffset" type=" Offset " minOccurs="0"/>
    <xsd:element name="paymentDatesAdjustments" type=" BusinessDayAdjustments " />
  </xsd:sequence>
  <xsd:attribute name="id" type=" xsd:ID " />
</xsd:complexType>

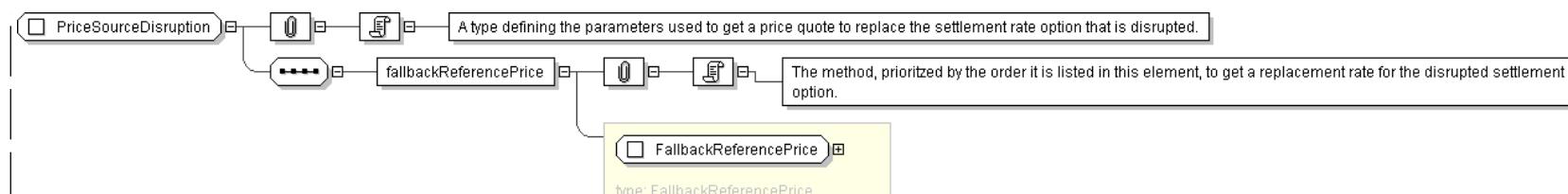
```

**Complex Type: PaymentDatesReference**

<b>Super-types:</b>	<a href="#">Reference</a> < <b>PaymentDatesReference</b> (by extension)
<b>Sub-types:</b>	None
<b>Name</b>	PaymentDatesReference
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">DateRelativeToPaymentDates</a>
<b>Abstract</b>	no
<b>Documentation</b>	Reference to a payment dates structure.
<b>XML Instance Representation</b>	
<pre>&lt;...&gt;   href="# IDREF [1]" /&gt;</pre>	
<b>Diagram</b>	
<pre> classDiagram     class PaymentDatesReference {         &lt;&lt;Reference&gt;&gt;         @ href : Reference     }     class Reference {         &lt;&lt;Reference&gt;&gt;     }     PaymentDatesReference "1" -- "1" Reference : @ href     PaymentDatesReference &lt; -- Reference   </pre> <p>The diagram illustrates the inheritance relationship between PaymentDatesReference and Reference. PaymentDatesReference is shown as a class with an attribute '@ href' of type Reference. A directed association connects PaymentDatesReference to Reference, labeled with the multiplicity '1' on both sides. The 'extends' keyword is also present in the association, indicating that PaymentDatesReference extends the functionality of Reference.</p>	
<b>Schema Component Representation</b>	
<pre> &lt;xsd:complexType name="PaymentDatesReference"&gt;   &lt;xsd:complexContent&gt;     &lt;xsd:extension base="# Reference"&gt;       &lt;xsd:attribute name="href" type="xsd:IDREF" use="required" reference="PaymentDates"/&gt;     &lt;/xsd:extension&gt;   &lt;/xsd:complexContent&gt; &lt;/xsd:complexType&gt;   </pre>	

[top](#)**Complex Type: PriceSourceDisruption**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None
<b>Name</b>	PriceSourceDisruption
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">NonDeliverableSettlement</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type defining the parameters used to get a price quote to replace the settlement rate option that is disrupted.
<b>XML Instance Representation</b>	
<pre>&lt;...&gt;   &lt;fallbackReferencePrice&gt; FallbackReferencePrice &lt;/fallbackReferencePrice&gt; [1]   'The method, prioritized by the order it is listed in this element, to get a replacement   rate for the disrupted settlement rate option.'</pre>	
<b>Diagram</b>	
<pre> classDiagram     class PriceSourceDisruption {         &lt;&lt;PriceSourceDisruption&gt;&gt;         &lt;&lt;FallbackReferencePrice&gt;&gt;         fallbackReferencePrice : FallbackReferencePrice     }     class FallbackReferencePrice {         &lt;&lt;FallbackReferencePrice&gt;&gt;     }   </pre> <p>The diagram shows PriceSourceDisruption as a class containing an attribute 'fallbackReferencePrice' of type FallbackReferencePrice.</p>	

**Schema Component Representation**

```

<xsd:complexType name="PriceSourceDisruption">
  <xsd:sequence>
    <xsd:element name="fallbackReferencePrice" type="#type:FallbackReferencePrice" />
  </xsd:sequence>
</xsd:complexType>
  
```

[top](#)**Complex Type: PrincipalExchange**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None

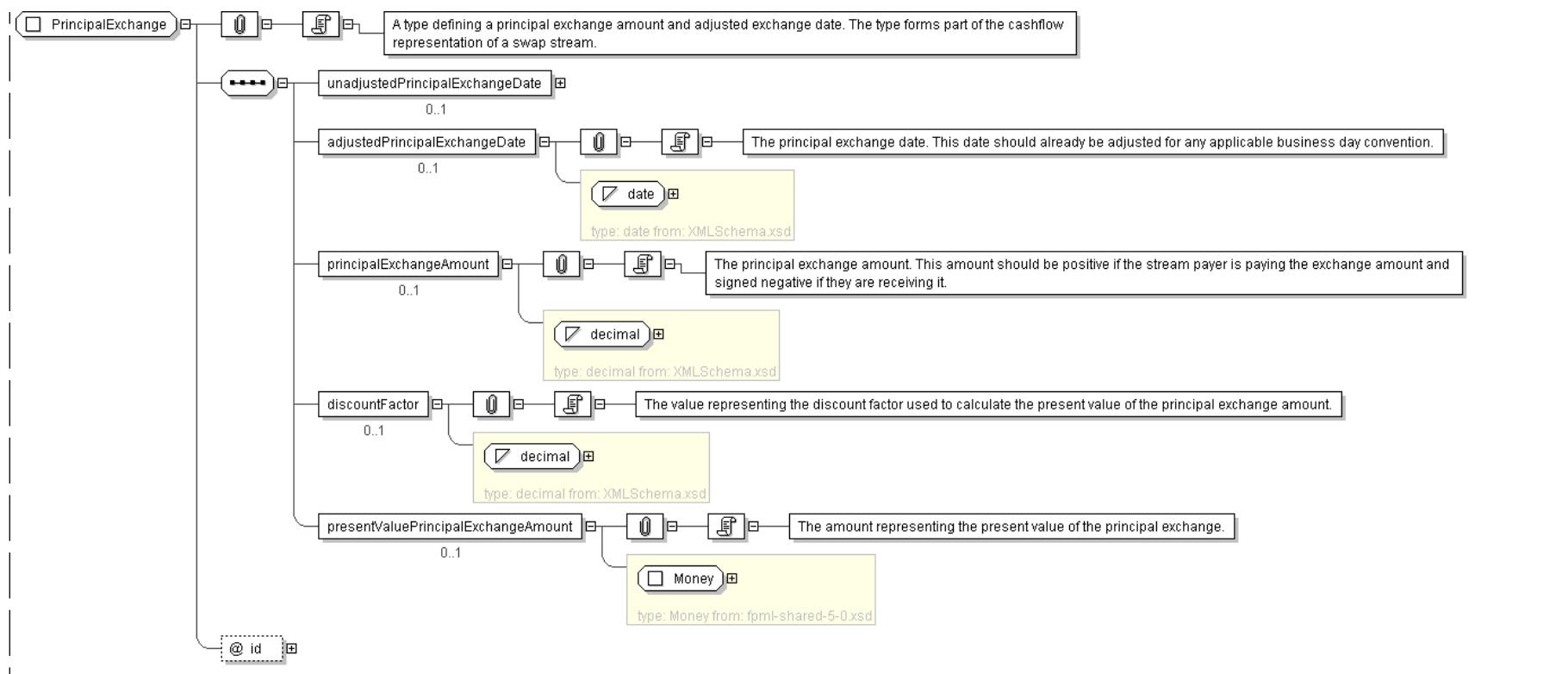
<b>Name</b>	PrincipalExchange
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">Cashflows</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type defining a principal exchange amount and adjusted exchange date. The type forms part of the cashflow representation of a swap stream.

**XML Instance Representation**

```

<...>
  id="#ID [0..1]">
    <unadjustedPrincipalExchangeDate> xsd:date </unadjustedPrincipalExchangeDate> [0..1]
    <adjustedPrincipalExchangeDate> xsd:date </adjustedPrincipalExchangeDate> [0..1]
    'The principal exchange date. This date should already be adjusted for any applicable
    business day convention.'
  <principalExchangeAmount> xsd:decimal </principalExchangeAmount> [0..1]
    '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.'
  <discountFactor> xsd:decimal </discountFactor> [0..1]
    'The value representing the discount factor used to calculate the present value of
    the principal exchange amount.'
  <presentValuePrincipalExchangeAmount> Money </presentValuePrincipalExchangeAmount> [0..1]
    'The amount representing the present value of the principal exchange.'
</...>
  
```

**Diagram**



#### Schema Component Representation

```

<xsd:complexType name="PrincipalExchange">
  <xsd:sequence>
    <xsd:element name="unadjustedPrincipalExchangeDate" type="xsd:date" minOccurs="0"/>
    <xsd:element name="adjustedPrincipalExchangeDate" type="xsd:date" minOccurs="0"/>
    <xsd:element name="principalExchangeAmount" type="xsd:decimal" minOccurs="0"/>
    <xsd:element name="discountFactor" type="xsd:decimal" minOccurs="0"/>
    <xsd:element name="presentValuePrincipalExchangeAmount" type="Money" minOccurs="0"/>
  </xsd:sequence>
  <xsd:attribute name="id" type="xsd:ID" />
</xsd:complexType>

```

top

#### Complex Type: RelevantUnderlyingDateReference

Super-types:	<a href="#">Reference</a> < RelevantUnderlyingDateReference (by extension)
Sub-types:	None
Name	RelevantUnderlyingDateReference
Used by (from the same schema document)	Complex Type <a href="#">FinalCalculationPeriodDateAdjustment</a>
Abstract	no
Documentation	Reference to relevant underlying date.

#### XML Instance Representation

```

<...
  href="xsd:IDREF [1]"/>

```

#### Diagram

**Schema Component Representation**

```

<xsd:complexType name="RelevantUnderlyingDateReference">
  <xsd:complexContent>
    <xsd:extension base=" Reference ">
      <xsd:attribute name="href" type=" xsd:IDREF " use="required"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
  
```

[top](#)**Complex Type: ResetDates**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None

<b>Name</b>	ResetDates
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">InterestRateStream</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type defining the parameters used to generate the reset dates schedule and associated fixing dates. The reset dates are determined relative to the calculation periods schedules dates.

**XML Instance Representation**

```

<...>
<id=" xsd:ID [1]">
<calculationPeriodDatesReference> CalculationPeriodDatesReference
</calculationPeriodDatesReference> [1]
'A pointer style reference to the associated calculation period dates component
defined elsewhere in the document.'

<resetRelativeTo> ResetRelativeToEnum </resetRelativeTo> [0..1]
'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.'

<initialFixingDate> RelativeDateOffset </initialFixingDate> [0..1]
<fixingDates> RelativeDateOffset </fixingDates> [1]
'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
V'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> Offset </rateCutOffDaysOffset> [0..1]
'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 reflux 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> `ResetFrequency` </resetFrequency> [1]

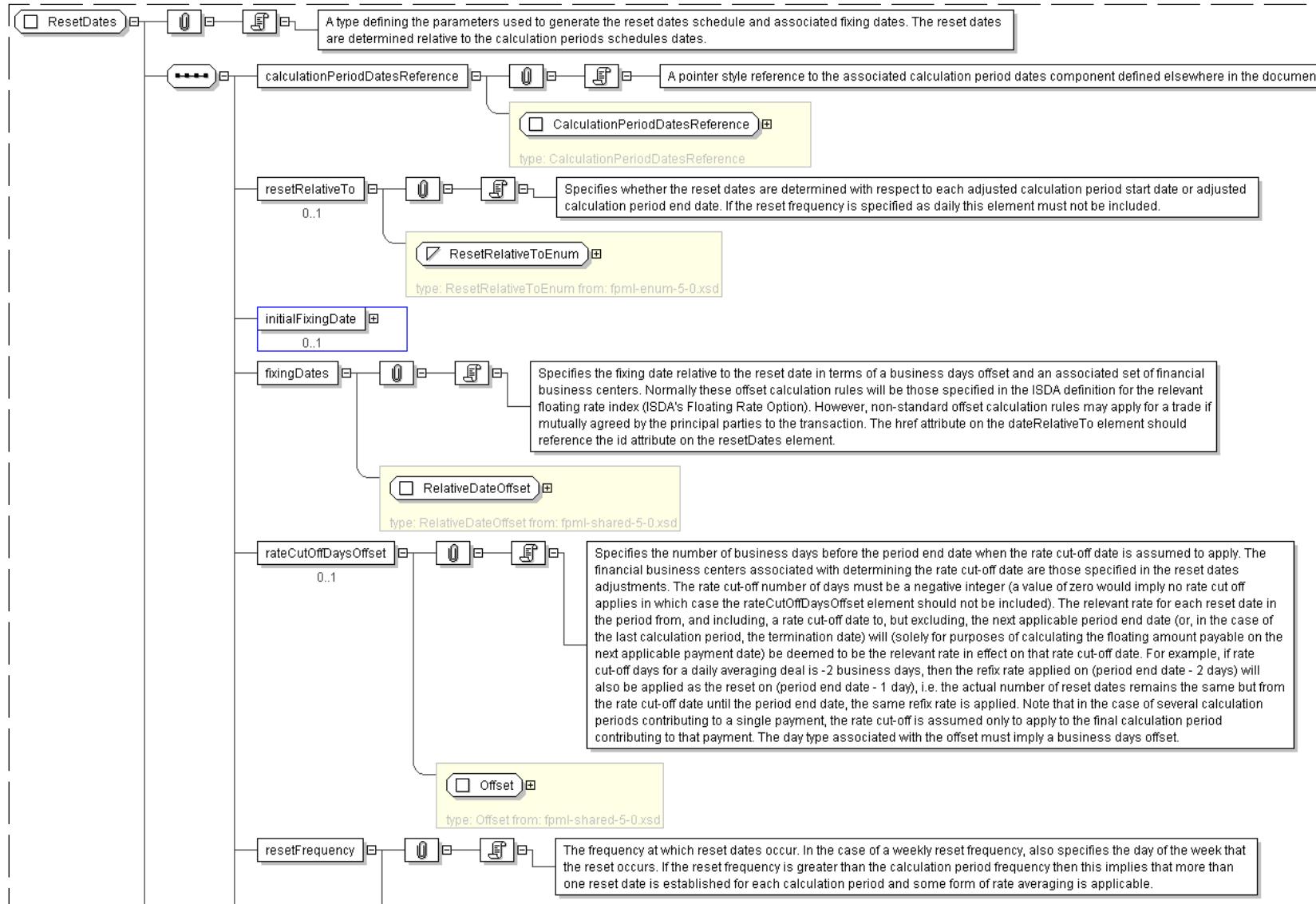
'The frequency at which reset dates occur. In the case of a weekly reset frequency, also specifies the day of the week that the reset occurs. If the reset frequency is greater than the calculation period frequency then this implies that more than one reset date is established for each calculation period and some form of rate averaging is applicable.'

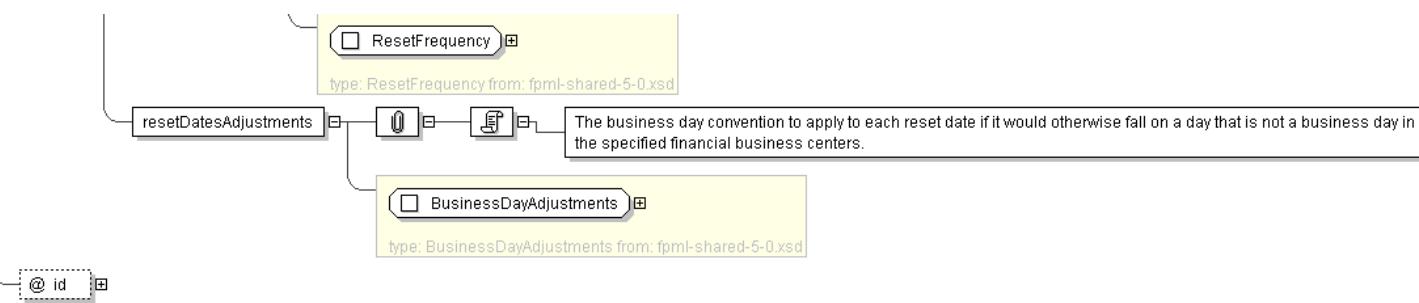
<resetDatesAdjustments> `BusinessDayAdjustments` </resetDatesAdjustments> [1]

'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.'

</...>

#### Diagram



**Schema Component Representation**

```

<xsd:complexType name="ResetDates">
  <xsd:sequence>
    <xsd:element name="calculationPeriodDatesReference" type=" CalculationPeriodDatesReference " />
    <xsd:element name="resetRelativeTo" type=" ResetRelativeTypeEnum " minOccurs="0" />
    <xsd:element name="initialFixingDate" type=" RelativeDateOffsetset " minOccurs="0" />
    <xsd:element name="fixingDates" type=" RelativeDateOffsetset " />
    <xsd:element name="rateCutOffDaysOffset" type=" Offset " minOccurs="0" />
    <xsd:element name="resetFrequency" type=" ResetFrequency " />
    <xsd:element name="resetDatesAdjustments" type=" BusinessDayAdjustments " />
  </xsd:sequence>
  <xsd:attribute name="id" type=" xsd:ID " use="required"/>
</xsd:complexType>

```

[top](#)**Complex Type: ResetDatesReference**

Super-types:	<a href="#">Reference</a> < <b>ResetDatesReference</b> (by extension)
Sub-types:	None

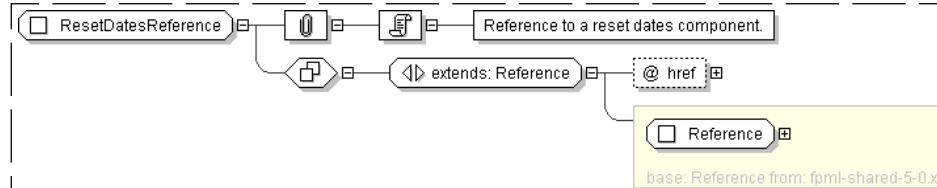
Name	ResetDatesReference
Used by (from the same schema document)	Complex Type <a href="#">PaymentDates</a>
Abstract	no
Documentation	Reference to a reset dates component.

**XML Instance Representation**

```

<...
  href=" xsd:IDREF [1]" />

```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="ResetDatesReference">
  <xsd:complexContent>
    <xsd:extension base=" Reference ">
      <xsd:attribute name="href" type=" xsd:IDREF " use="required" reference="ResetDates" />
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

## Complex Type: SettlementProvision

**Super-types:** None  
**Sub-types:** None

<b>Name</b>	SettlementProvision
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">InterestRateStream</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type defining the specification of settlement terms, occuring when the settlement currency is different to the notional currency of the trade.

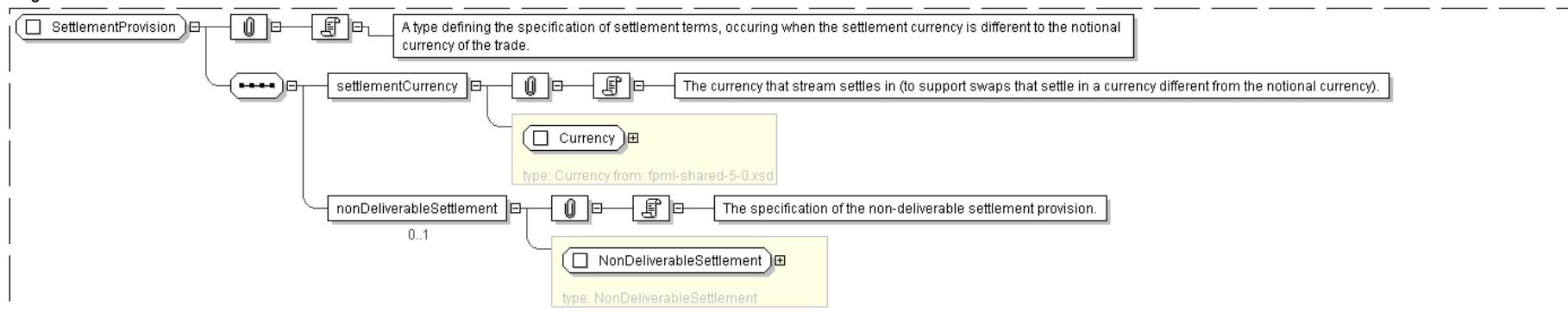
### XML Instance Representation

```
<....>
<settlementCurrency> Currency </settlementCurrency> [1]
'The currency that stream settles in (to support swaps that settle in a currency different
from the notional currency).'

<nonDeliverableSettlement> NonDeliverableSettlement </nonDeliverableSettlement> [0..1]
'The specification of the non-deliverable settlement provision.'

</....>
```

### Diagram



### Schema Component Representation

```

<xsd:complexType name="SettlementProvision">
  <xsd:sequence>
    <xsd:element name="settlementCurrency" type="Currency" />
    <xsd:element name="nonDeliverableSettlement" type="NonDeliverableSettlement" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>
  
```

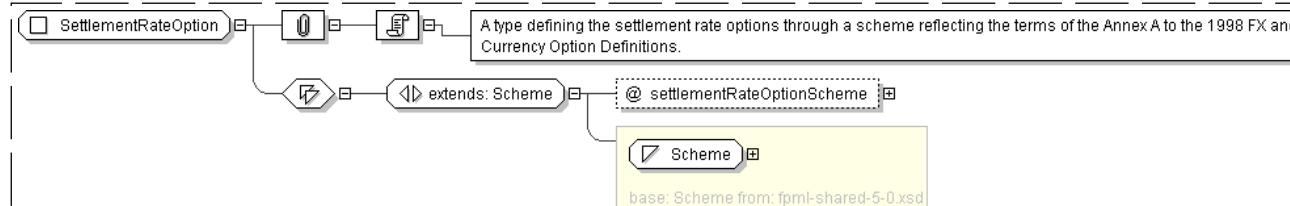
## Complex Type: SettlementRateOption

**Super-types:** [Scheme](#) < SettlementRateOption (by extension)  
**Sub-types:** None

<b>Name</b>	SettlementRateOption
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">FallbackReferencePrice</a> , Complex Type <a href="#">NonDeliverableSettlement</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type defining the settlement rate options through a scheme reflecting the terms of the Annex A to the 1998 FX and Currency Option Definitions.

### XML Instance Representation

```
'<...>
<xs:complexType name="SettlementRateOption">
  <xs:simpleContent>
    <xs:extension base="#REF_Scheme">
      <xs:attribute name="settlementRateOptionScheme" type="xs:anyURI" default="http://www.fpml.org/coding-scheme/settlement-rate-option"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
```

**Diagram****Schema Component Representation**

```
<xsd:complexType name="SettlementRateOption">
  <xsd:simpleContent>
    <xsd:extension base="#REF_Scheme">
      <xs:attribute name="settlementRateOptionScheme" type="xs:anyURI" default="http://www.fpml.org/coding-scheme/settlement-rate-option"/>
    </xs:extension>
  </xsd:simpleContent>
</xsd:complexType>
```

top

**Complex Type: SinglePartyOption**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None

<b>Name</b>	SinglePartyOption
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">OptionalEarlyTermination</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type describing the buyer and seller of an option.

**XML Instance Representation**

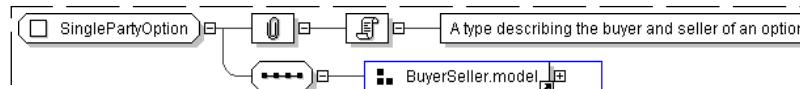
```
<...>
<buyerPartyReference> PartyReference </buyerPartyReference> [1]
'A reference to the party that buys this instrument, i.e. pays for this instrument and receives the rights defined by it. See 2000 ISDA definitions Article 11.1 (b). In the case of FRAs this is the fixed rate payer.'
```

```
<buyerAccountReference> AccountReference </buyerAccountReference> [0..1]
'A reference to the account that buys this instrument.'
```

```
<sellerPartyReference> PartyReference </sellerPartyReference> [1]
'A reference to the party that sells ("writes") this instrument, i.e. that grants the rights defined by this instrument and in return receives a payment for it. See 2000 ISDA definitions Article 11.1 (a). In the case of FRAs this is the floating rate payer.'
```

```
<sellerAccountReference> AccountReference </sellerAccountReference> [0..1]
'A reference to the account that sells this instrument.'
```

```
</...>
```

**Diagram**

**Schema Component Representation**

```
<xsd:complexType name="SinglePartyOption">
  <xsd:sequence>
    <xsd:group ref=" BuyerSeller.model " />
  </xsd:sequence>
</xsd:complexType>
```

top

**Complex Type: StubCalculationPeriodAmount**

**Super-types:** None  
**Sub-types:** None

<b>Name</b>	StubCalculationPeriodAmount
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">InterestRateStream</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type 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 used 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 component, as the case may be, must not be included.

**XML Instance Representation**

```
<....>
  <calculationPeriodDatesReference> CalculationPeriodDatesReference
  </calculationPeriodDatesReference> [1]
  'A pointer style reference to the associated calculation period dates component
  defined elsewhere in the document.'
```

Start [Choice](#) [1]

<initialStub> [StubValue](#) </initialStub> [1]

'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> [StubValue](#) </finalStub> [0..1]

'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.'

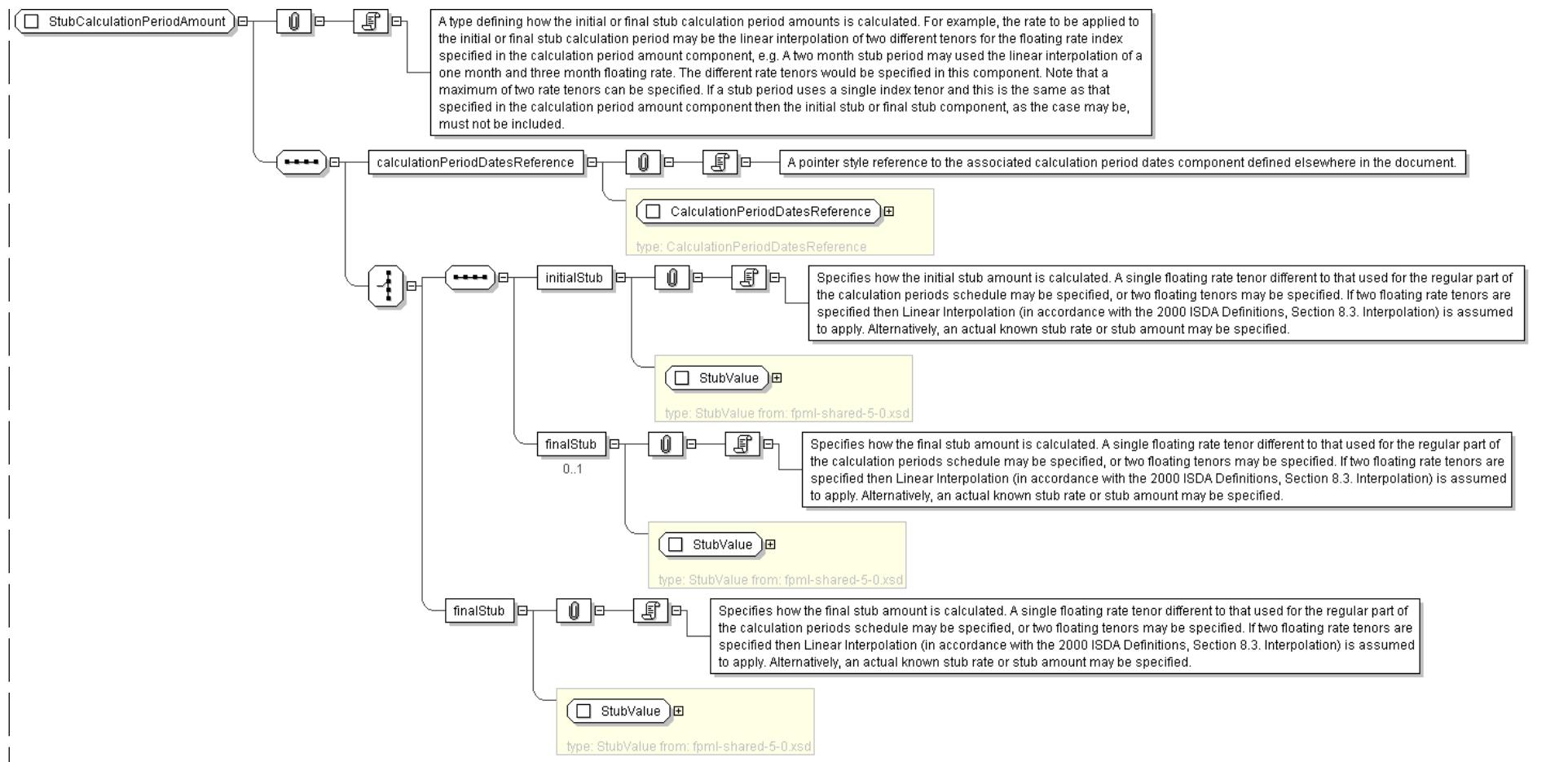
<finalStub> [StubValue](#) </finalStub> [1]

'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.'

End Choice

&lt;/....&gt;

**Diagram**



#### Schema Component Representation

```

<xsd:complexType name="StubCalculationPeriodAmount">
  <xsd:sequence>
    <xsd:element name="calculationPeriodDatesReference" type=" CalculationPeriodDatesReference " />
    <xsd:choice>
      <xsd:sequence>
        <xsd:element name="initialStub" type=" StubValue " />
        <xsd:element name="finalStub" type=" StubValue " minOccurs="0" />
      </xsd:sequence>
      <xsd:element name="finalStub" type=" StubValue " />
    </xsd:choice>
  </xsd:sequence>
</xsd:complexType>

```

top

#### Complex Type: Swap

Super-types: [Product](#) < **Swap** (by extension)

None

Name

Swap

Used by (from the same schema document)

Element [swap](#)

Abstract

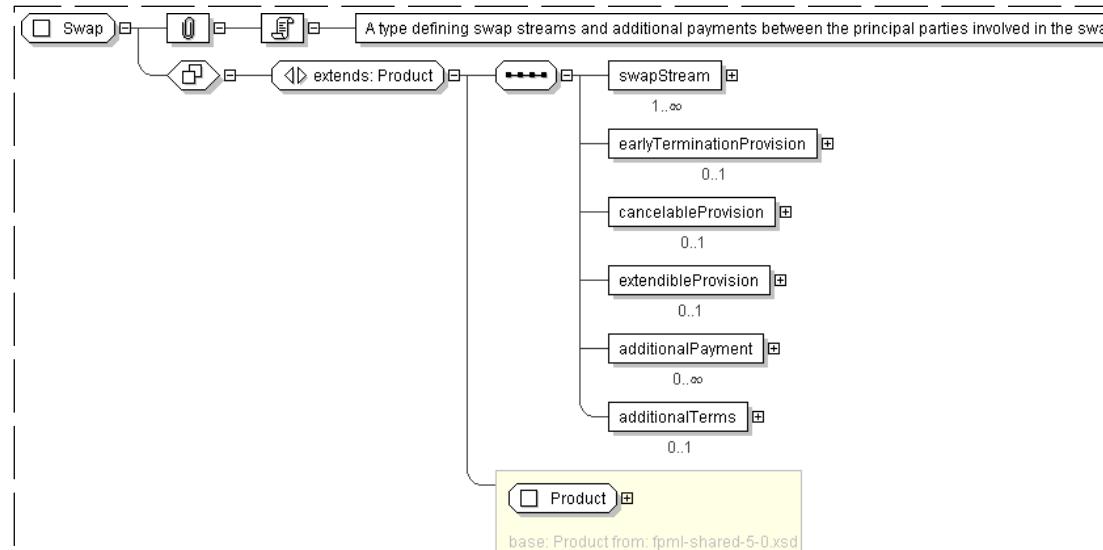
no

**Documentation**

A type defining swap streams and additional payments between the principal parties involved in the swap.

**XML Instance Representation**

```
<...>
<id=" xsd:ID [0..1]">
  <productType> ProductType </productType> [0..*]
  'A classification of the type of product. FpML defines a simple product categorization using a coding scheme.'
  <productId> ProductId </productId> [0..*]
  'A product 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.'
  <swapStream> InterestRateStream </swapStream> [1..*]
  'The swap streams.'
  <earlyTerminationProvision> EarlyTerminationProvision </earlyTerminationProvision> [0..1]
  'Parameters specifying provisions relating to the optional and mandatory early terminarion of a swap transaction.'
  <cancelableProvision> CancelableProvision </cancelableProvision> [0..1]
  '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> ExtendibleProvision </extendibleProvision> [0..1]
  'A provision that allows the specification of an embedded option with 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> Payment </additionalPayment> [0..*]
  'Additional payments between the principal parties.'
  <additionalTerms> SwapAdditionalTerms </additionalTerms> [0..1]
  'Contains any additional terms to the swap contract.'
</...>
```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="Swap">
  <xsd:complexContent>
    <xsd:extension base=" Product ">
      <xsd:sequence>
        <xsd:element name="swapStream" type=" InterestRateStream " maxOccurs="unbounded"/>
        <xsd:element name="earlyTerminationProvision" type=" EarlyTerminationProvision " minOccurs="0"/>
        <xsd:element name="cancelableProvision" type=" CancelableProvision " minOccurs="0"/>
        <xsd:element name="extendibleProvision" type=" ExtendibleProvision " minOccurs="0"/>
        <xsd:element name="additionalPayment" type=" Payment " minOccurs="0" maxOccurs="unbounded"/>
        <xsd:element name="additionalTerms" type=" SwapAdditionalTerms " minOccurs="0"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

[top](#)

## Complex Type: SwapAdditionalTerms

<b>Super-types:</b>	None
<b>Sub-types:</b>	None

<b>Name</b>	SwapAdditionalTerms
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">Swap</a>
<b>Abstract</b>	no
<b>Documentation</b>	Additional terms to a swap contract.

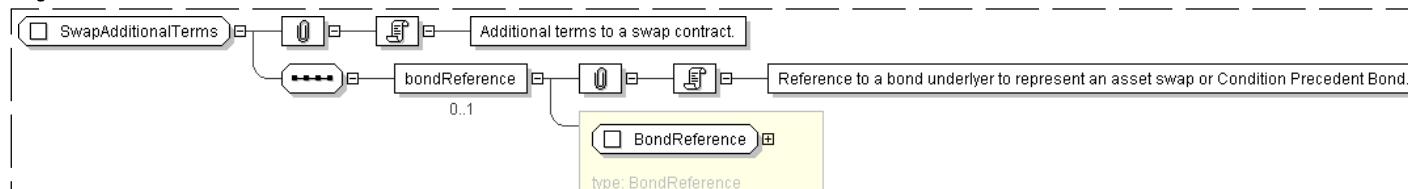
### XML Instance Representation

```

<....>
<bondReference> BondReference </bondReference> [0..1]
'Reference to a bond underlyer to represent an asset swap or Condition Precedent Bond.'
</....>

```

### Diagram



### Schema Component Representation

```

<xsd:complexType name="SwapAdditionalTerms">
  <xsd:sequence>
    <xsd:element name="bondReference" type=" BondReference " minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>

```

[top](#)

## Complex Type: Swaption

<b>Super-types:</b>	<a href="#">Product</a> < <a href="#">Swaption</a> (by extension)
<b>Sub-types:</b>	None

<b>Name</b>	Swaption
<b>Used by (from the same schema document)</b>	Element <a href="#">swaption</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type to define an option on a swap.

**XML Instance Representation**

```
<...>
  id=" xsd:ID [0..1]">
    <productType> ProductType </productType> [0..*]
      'A classification of the type of product. FpML defines a simple product categorization using a coding scheme.'
```

```
<productId> ProductId </productId> [0..*]
  'A product 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.'
```

```
<buyerPartyReference> PartyReference </buyerPartyReference> [1]
  'A reference to the party that buys this instrument, ie. pays for this instrument and receives the rights defined by it. See 2000 ISDA definitions Article 11.1 (b). In the case of FRAs this is the fixed rate payer.'
```

```
<buyerAccountReference> AccountReference </buyerAccountReference> [0..1]
  'A reference to the account that buys this instrument.'
```

```
<sellerPartyReference> PartyReference </sellerPartyReference> [1]
  'A reference to the party that sells ("writes") this instrument, i.e. that grants the rights defined by this instrument and in return receives a payment for it. See 2000 ISDA definitions Article 11.1 (a). In the case of FRAs this is the floating rate payer.'
```

```
<sellerAccountReference> AccountReference </sellerAccountReference> [0..1]
  'A reference to the account that sells this instrument.'
```

```
<premium> Payment </premium> [0..*]
  'The option premium amount payable by buyer to seller on the specified payment date.'
```

```
<exercise> ... </exercise> [1]
<exerciseProcedure> ExerciseProcedure </exerciseProcedure> [0..1]
  'A set of parameters defining procedures associated with the exercise.'
```

```
<calculationAgent> CalculationAgent </calculationAgent> [0..1]
  'The ISDA Calculation Agent responsible for performing duties associated with an optional early termination.'
```

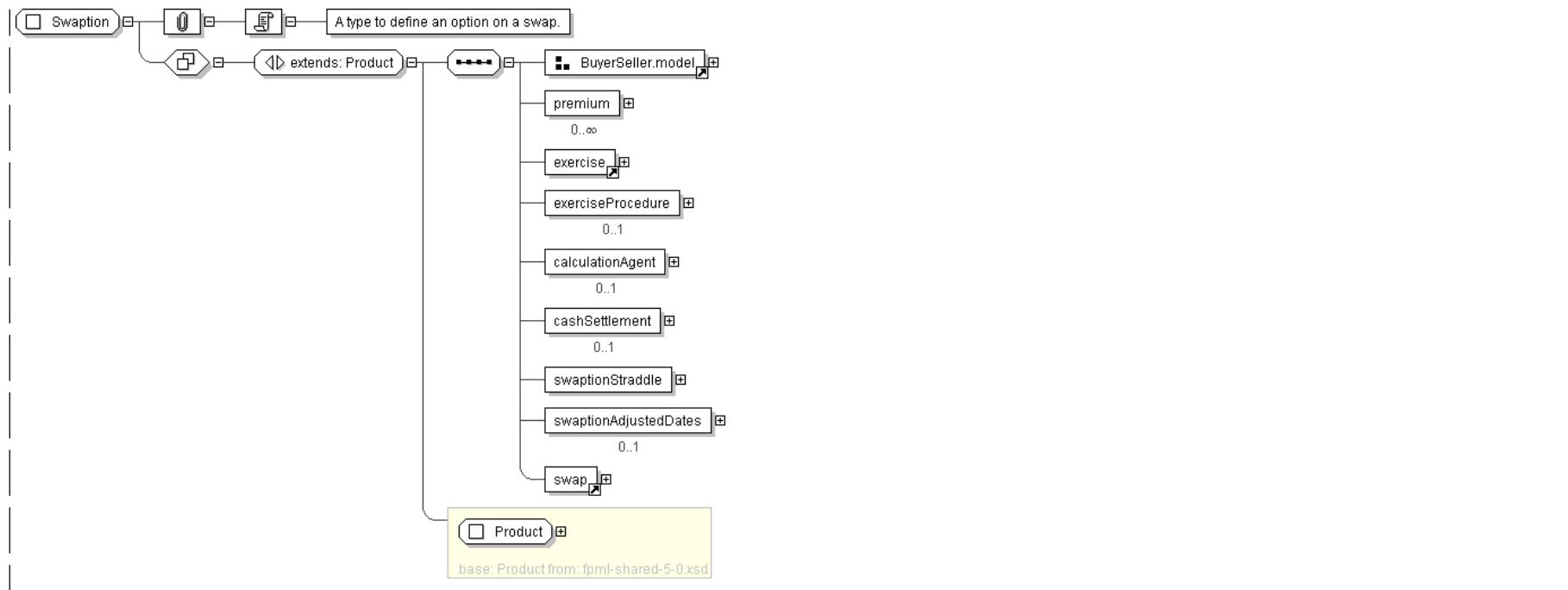
```
<cashSettlement> CashSettlement </cashSettlement> [0..1]
  'If specified, this means that cash settlement is applicable to the transaction and defines the parameters associated with the cash settlement procedure. If not specified, then physical settlement is applicable.'
```

```
<swaptionStraddle> xsd:boolean </swaptionStraddle> [1]
  'Whether the option is a swaption or a swaption straddle.'
```

```
<swaptionAdjustedDates> SwaptionAdjustedDates </swaptionAdjustedDates> [0..1]
  'The adjusted dates associated with swaption exercise. These dates have been adjusted for any applicable business day convention.'
```

```
<swap> ... </swap> [1]
</...>
```

**Diagram**

**Schema Component Representation**

```

<xsd:complexType name="Swaption">
  <xsd:complexContent>
    <xsd:extension base=" Product ">
      <xsd:sequence>
        <xsd:group ref=" BuyerSeller.model ">
          <xsd:element name="premium" type=" Payment " minOccurs="0" maxOccurs="unbounded"/>
          <xsd:element ref=" exercise ">
          <xsd:element name="exerciseProcedure" type=" ExerciseProcedure " minOccurs="0"/>
          <xsd:element name="calculationAgent" type=" CalculationAgent " minOccurs="0"/>
          <xsd:element name="cashSettlement" type=" CashSettlement " minOccurs="0"/>
          <xsd:element name="swaptionStraddle" type=" xsd:boolean " />
          <xsd:element name="swaptionAdjustedDates" type=" SwaptionAdjustedDates " minOccurs="0"/>
          <xsd:element ref=" swap "/>
        </xsd:sequence>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>

```

[top](#)**Complex Type: [SwaptionAdjustedDates](#)**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None
<b>Name</b>	<a href="#">SwaptionAdjustedDates</a>
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">Swaption</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type describing the adjusted dates associated with swaption exercise and settlement.

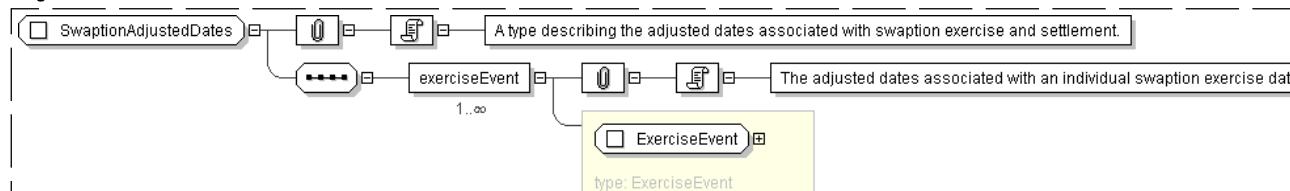
**XML Instance Representation**

```

<...>
<exerciseEvent> ExerciseEvent </exerciseEvent> [1..*]
  'The adjusted dates associated with an individual swaption exercise date.'

```

&lt;/...&gt;

**Diagram****Schema Component Representation**

```

<xsd:complexType name="SwaptionAdjustedDates">
  <xsd:sequence>
    <xsd:element name="exerciseEvent" type=" ExerciseEvent " maxOccurs="unbounded" />
  </xsd:sequence>
</xsd:complexType>
  
```

top

**Complex Type: ValuationDatesReference**

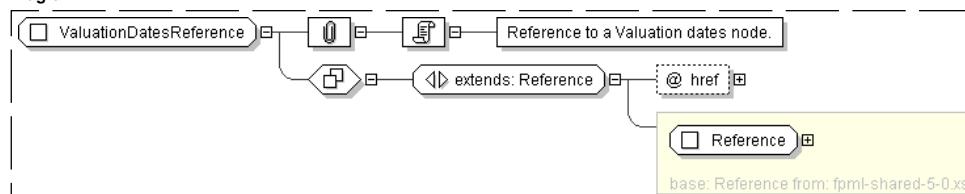
Super-types:	<a href="#">Reference</a> < <b>ValuationDatesReference</b> (by extension)
Sub-types:	None

Name	ValuationDatesReference
Used by (from the same schema document)	Complex Type <a href="#">PaymentDates</a>
Abstract	no
Documentation	Reference to a Valuation dates node.

**XML Instance Representation**

```

<...
  href=" xsd:IDREF [1]" />
  
```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="ValuationDatesReference">
  <xsd:complexContent>
    <xsd:extension base=" Reference ">
      <xsd:attribute name="href" type=" xsd:IDREF " use="required" reference="FxFixingDate" />
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
  
```

top

**Complex Type: ValuationPostponement**

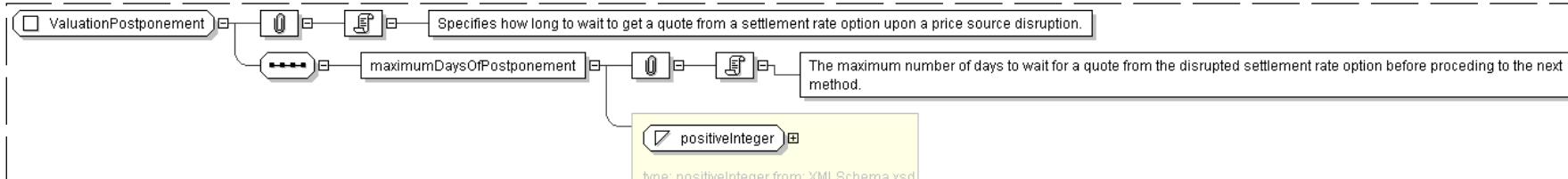
Super-types:	None
Sub-types:	None

Name	ValuationPostponement
Used by (from the same schema document)	Complex Type <a href="#">FallbackReferencePrice</a>
Abstract	no
Documentation	Specifies how long to wait to get a quote from a settlement rate option upon a price source disruption.

**XML Instance Representation**

```
<....>
<maximumDaysOfPostponement> xsd:positiveInteger </maximumDaysOfPostponement> [1]
'The maximum number of days to wait for a quote from the disrupted settlement rate
option before proceeding to the next method.'
```

&lt;/....&gt;

**Diagram****Schema Component Representation**

```
<xsd:complexType name="ValuationPostponement">
  <xsd:sequence>
    <xsd:element name="maximumDaysOfPostponement" type="xsd:positiveInteger" />
  </xsd:sequence>
</xsd:complexType>
```

top

**Complex Type: YieldCurveMethod**

Super-types:	None
Sub-types:	None

Name	YieldCurveMethod
Used by (from the same schema document)	Complex Type <a href="#">CashSettlement</a> , Complex Type <a href="#">CashSettlement</a> , Complex Type <a href="#">CashSettlement</a>
Abstract	no
Documentation	A type defining the parameters required for each of the ISDA defined yield curve methods for cash settlement.

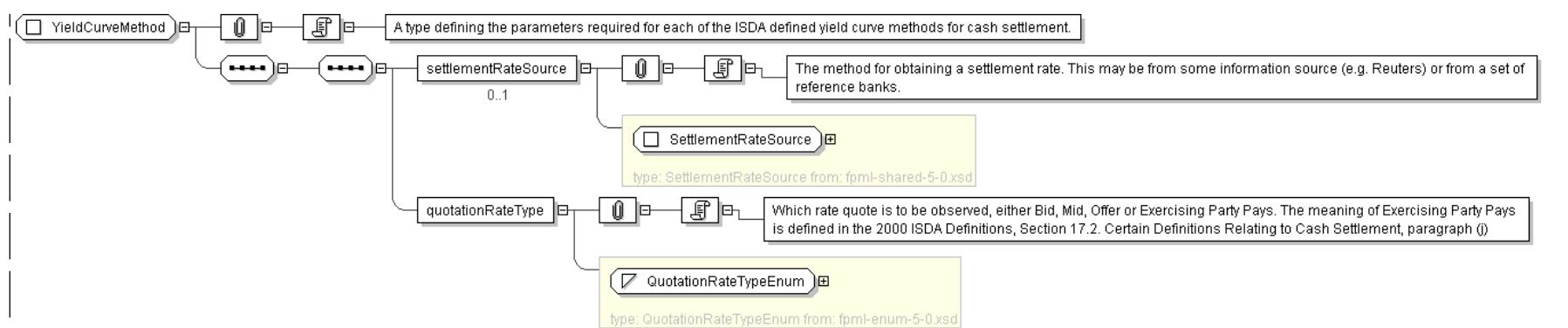
**XML Instance Representation**

```
<....>
<settlementRateSource> SettlementRateSource </settlementRateSource> [0..1]
'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> QuotationRateTypeEnum </quotationRateType> [1]
'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)'
```

&lt;/....&gt;

**Diagram**

**Schema Component Representation**

```
<xsd:complexType name="YieldCurveMethod">
  <xsd:sequence>
    <xsd:sequence>
      <xsd:element name="settlementRateSource" type="#type: SettlementRateSource" minOccurs="0"/>
      <xsd:element name="quotationRateType" type="#type: QuotationRateTypeEnum" />
    </xsd:sequence>
  </xsd:sequence>
</xsd:complexType>
```

[top](#)**Model Group: DiscountRate.model**

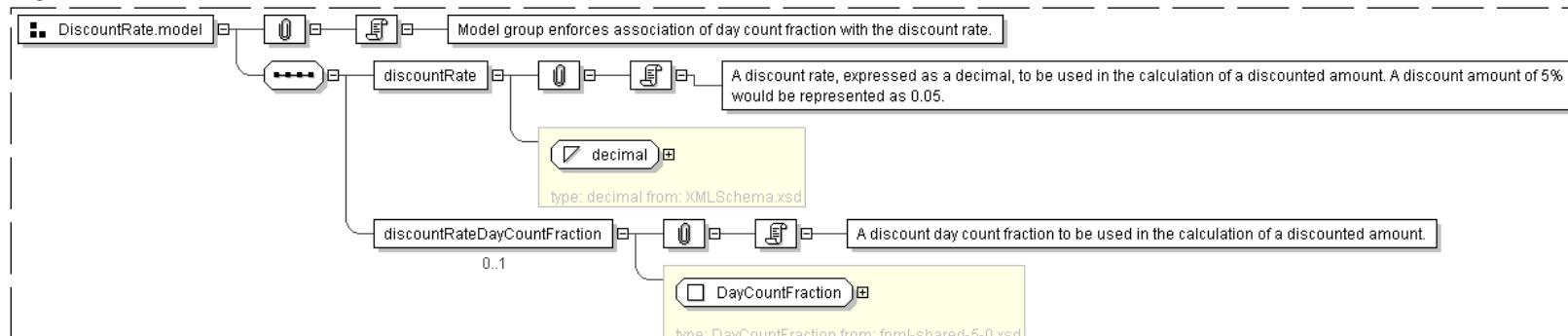
Name	DiscountRate.model
Used by (from the same schema document)	Complex Type <a href="#">Discounting</a>
Documentation	Model group enforces association of day count fraction with the discount rate.

**XML Instance Representation**

```
<discountRate> xsd:decimal </discountRate> [1]
'A discount rate, expressed as a decimal, to be used in the calculation of a discounted
amount. A discount amount of 5% would be represented as 0.05.'
```

```
<discountRateDayCountFraction> DayCountFraction </discountRateDayCountFraction> [0..1]
'A discount day count fraction to be used in the calculation of a discounted amount.'
```

**Diagram****Schema Component Representation**

```
<xsd:group name="DiscountRate.model">
  <xsd:sequence>
```

```

<xsd:element name="discountRate" type="xsd:decimal" />
<xsd:element name="discountRateDayCountFraction" type="DayCountFraction" minOccurs="0" />
</xsd:sequence>
</xsd:group>

```

[top](#)**Model Group: MandatoryEarlyTermination.model**

<b>Name</b>	MandatoryEarlyTermination.model
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">EarlyTerminationProvision</a>

**XML Instance Representation**

```

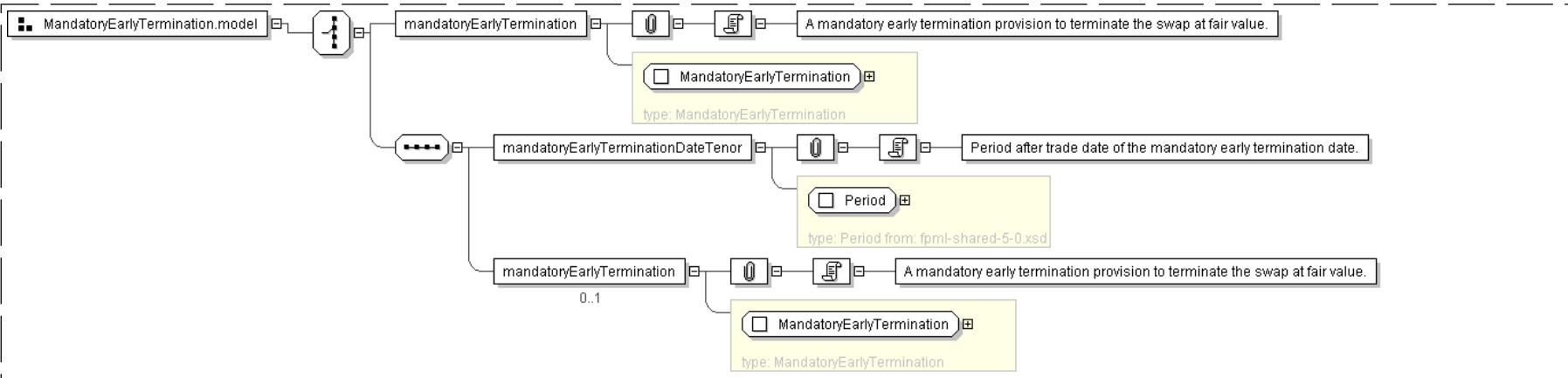
Start Choice [1]
<mandatoryEarlyTermination> MandatoryEarlyTermination </mandatoryEarlyTermination> [1]
'A mandatory early termination provision to terminate the swap at fair value.'

<mandatoryEarlyTerminationDateTenor> Period </mandatoryEarlyTerminationDateTenor> [1]
'Period after trade date of the mandatory early termination date.'

<mandatoryEarlyTermination> MandatoryEarlyTermination </mandatoryEarlyTermination> [0..1]
'A mandatory early termination provision to terminate the swap at fair value.'

```

End Choice

**Diagram****Schema Component Representation**

```

<xsd:group name="MandatoryEarlyTermination.model">
  <xsd:choice>
    <xsd:element name="mandatoryEarlyTermination" type="MandatoryEarlyTermination" />
    <xsd:sequence>
      <xsd:element name="mandatoryEarlyTerminationDateTenor" type="Period" />
      <xsd:element name="mandatoryEarlyTermination" type="MandatoryEarlyTermination" minOccurs="0" />
    </xsd:sequence>
  </xsd:choice>
</xsd:group>

```

[top](#)**Model Group: OptionalEarlyTermination.model**

<b>Name</b>	OptionalEarlyTermination.model
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">EarlyTerminationProvision</a> , Complex Type <a href="#">EarlyTerminationProvision</a>

**XML Instance Representation**

```

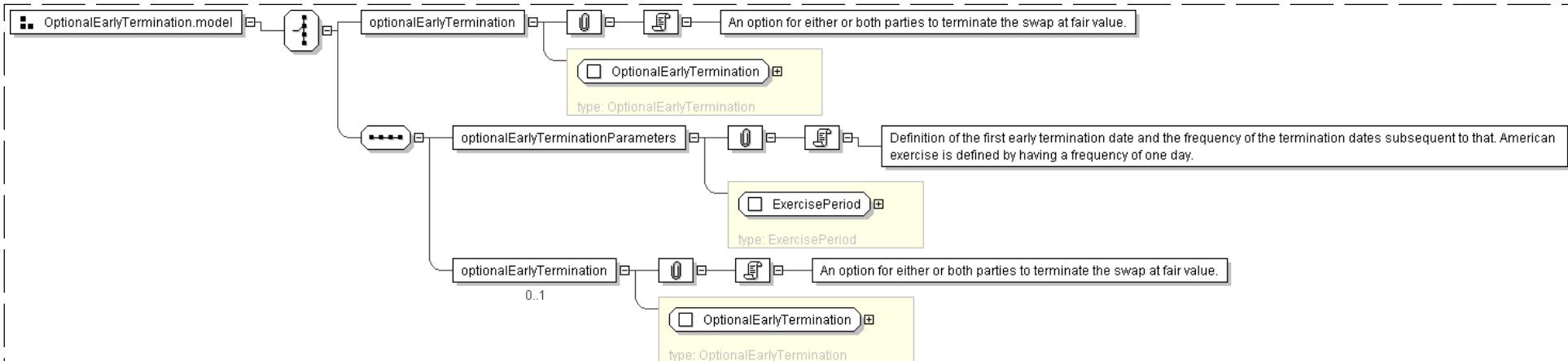
Start Choice [1]
<optionalEarlyTermination> OptionalEarlyTermination </optionalEarlyTermination> [1]
'An option for either or both parties to terminate the swap at fair value.'

<optionalEarlyTerminationParameters> ExercisePeriod </optionalEarlyTerminationParameters> [1]
'Definition of the first early termination date and the frequency of the termination
dates subsequent to that. American exercise is defined by having a frequency of one day.'

<optionalEarlyTermination> OptionalEarlyTermination </optionalEarlyTermination> [0..1]
'An option for either or both parties to terminate the swap at fair value.'

End Choice

```

**Diagram****Schema Component Representation**

```

<xsd:group name="OptionalEarlyTermination.model">
  <xsd:choice>
    <xsd:element name="optionalEarlyTermination" type="OptionalEarlyTermination" />
    <xsd:sequence>
      <xsd:element name="optionalEarlyTerminationParameters" type="ExercisePeriod" />
      <xsd:element name="optionalEarlyTermination" type="OptionalEarlyTermination" minOccurs="0" />
    </xsd:sequence>
  </xsd:choice>
</xsd:group>

```

top

**Legend****Complex Type:**

Schema Component Type

**AusAddress**

Schema Component Name

**Super-types:**[Address](#) < AusAddress (by extension)**Sub-types:**• [QLDAddress](#) (by restriction)

If this schema component is a type definition, its type hierarchy is shown in a gray-bordered box.

<b>Name</b>	AusAddress
<b>Abstract</b>	no

The table above displays the properties of this schema component.

**XML Instance Representation**

```
<... country="Australia" >
<unitNo> string </unitNo> [0..1]
<houseNo> string </houseNo> [1]
<street> string </street> [1]
Start Choice [1]
<city> string </city> [1]
<town> string </town> [1]
End Choice
<state> AusStates </state> [1]
<postcode> string <><pattern = [1-9][0-9]{3}></postcode> [1]
</...>
```

The XML Instance Representation table above shows the schema component's content as an XML instance.

- The minimum and maximum occurrence of elements and attributes are provided in square brackets, e.g. [0..1].
- Model group information are shown in gray, e.g. Start Choice ... End Choice.
- For type derivations, the elements and attributes that have been added to or changed from the base type's content are shown in **bold**.
- If an element/attribute has a fixed value, the fixed value is shown in green, e.g. country="Australia".
- Otherwise, the type of the element/attribute is displayed.
  - If the element/attribute's type is in the schema, a link is provided to it.
  - For local simple type definitions, the constraints are displayed in angle brackets, e.g. <><pattern = [1-9][0-9]{3}></>.

**Schema Component Representation**

```
<complexType name="AusAddress">
<complexContent>
<extension base=" Address ">
<sequence>
<element name="state" type=" AusStates " />
<element name="postcode">
<simpleType>
<restriction base=" string ">
<pattern value="[1-9][0-9]{3}" />
</restriction>
</simpleType>
</element>
</sequence>
<attribute name="country" type=" string " fixed="Australia"/>
</extension>
</complexContent>
</complexType>
```

The Schema Component Representation table above displays the underlying XML representation of the schema component. (Annotations are not shown.)

top

**Glossary**

**Abstract** (Applies to complex type definitions and element declarations). An abstract element or complex type cannot be used to validate an element instance. If there is a reference to an abstract element, only element declarations that can substitute the abstract element can be used to validate the instance. For references to abstract type definitions, only derived types can be used.

**All Model Group** Child elements can be provided *in any order* in instances. See: <http://www.w3.org/TR/xmlschema-1/#element-all>.

**Choice Model Group** Only one from the list of child elements and model groups can be provided in instances. See: <http://www.w3.org/TR/xmlschema-1/#element-choice>.

**Collapse Whitespace Policy** Replace tab, line feed, and carriage return characters with space character (Unicode character 32). Then, collapse contiguous sequences of space characters into single space character, and remove leading and trailing space characters.

**Disallowed Substitutions** (Applies to element declarations). If *substitution* is specified, then *substitution group* members cannot be used in place of the given element declaration to validate element instances. If *derivation methods*, e.g. extension, restriction, are specified, then the given element declaration will not validate element instances that have types derived from the element declaration's type using the specified derivation methods. Normally, element instances can override their declaration's type by specifying an *xsi:type* attribute.

**Key Constraint** Like *Uniqueness Constraint*, but additionally requires that the specified value(s) must be provided. See: [http://www.w3.org/TR/xmlschema-1/#identity-constraint\\_Definitions](http://www.w3.org/TR/xmlschema-1/#identity-constraint_Definitions).

**Key Reference Constraint** Ensures that the specified value(s) must match value(s) from a [Key Constraint](#) or [Uniqueness Constraint](#). See: [http://www.w3.org/TR/xmlschema-1/#cidentity-constraint\\_Definitions](http://www.w3.org/TR/xmlschema-1/#cidentity-constraint_Definitions).

**Model Group** Groups together element content, specifying the order in which the element content can occur and the number of times the group of element content may be repeated. See: [http://www.w3.org/TR/xmlschema-1/#Model\\_Groups](http://www.w3.org/TR/xmlschema-1/#Model_Groups).

**Nillable** (Applies to element declarations). If an element declaration is nullable, instances can use the `xsi:nil` attribute. The `xsi:nil` attribute is the boolean attribute, `nil`, from the <http://www.w3.org/2001/XMLSchema-instance> namespace. If an element instance has an `xsi:nil` attribute set to true, it can be left empty, even though its element declaration may have required content.

**Notation** A notation is used to identify the format of a piece of data. Values of elements and attributes that are of type, NOTATION, must come from the names of declared notations. See: [http://www.w3.org/TR/xmlschema-1/#cNotation\\_Declarations](http://www.w3.org/TR/xmlschema-1/#cNotation_Declarations).

**Preserve Whitespace Policy** Preserve whitespaces exactly as they appear in instances.

**Prohibited Derivations** (Applies to type definitions). Derivation methods that cannot be used to create sub-types from a given type definition.

**Prohibited Substitutions** (Applies to complex type definitions). Prevents sub-types that have been derived using the specified derivation methods from validating element instances in place of the given type definition.

**Replace Whitespace Policy** Replace tab, line feed, and carriage return characters with space character (Unicode character 32).

**Sequence Model Group** Child elements and model groups must be provided *in the specified order* in instances. See: <http://www.w3.org/TR/xmlschema-1/#element-sequence>.

**Substitution Group** Elements that are *members* of a substitution group can be used wherever the `head` element of the substitution group is referenced.

**Substitution Group Exclusions** (Applies to element declarations). Prohibits element declarations from nominating themselves as being able to substitute a given element declaration, if they have types that are derived from the original element's type using the specified derivation methods.

**Target Namespace** The target namespace identifies the namespace that components in this schema belongs to. If no target namespace is provided, then the schema components do not belong to any namespace.

**Uniqueness Constraint** Ensures uniqueness of an element/attribute value, or a combination of values, within a specified scope. See: [http://www.w3.org/TR/xmlschema-1/#cidentity-constraint\\_Definitions](http://www.w3.org/TR/xmlschema-1/#cidentity-constraint_Definitions).

top

---

Generated by <oXygen/> XML Editor using a modified version of xs3p that adds schema diagrams and chunking support.

# XML Schema Documentation

---

## Table of Contents

- [Schema Document Properties](#)
- [Global Declarations](#)
  - [Element: eventStatusException](#)
  - [Element: eventStatusResponse](#)
  - [Element: messageRejected](#)
  - [Element: requestEventStatus](#)
- [Global Definitions](#)
  - [Complex Type: Acknowledgement](#)
  - [Complex Type: AdditionalData](#)
  - [Complex Type: CorrectableRequestMessage](#)
  - [Complex Type: CorrelationId](#)
  - [Complex Type: EventIdentifier](#)
  - [Complex Type: EventStatus](#)
  - [Complex Type: EventStatusItem](#)
  - [Complex Type: EventStatusResponse](#)
  - [Complex Type: Exception](#)
  - [Complex Type: ExceptionMessageHeader](#)
  - [Complex Type: Message](#)
  - [Complex Type: MessageAddress](#)
  - [Complex Type: MessageHeader](#)
  - [Complex Type: MessageId](#)
  - [Complex Type: NonCorrectableRequestMessage](#)
  - [Complex Type: NotificationMessage](#)
  - [Complex Type: NotificationMessageHeader](#)
  - [Complex Type: OnBehalfOf](#)
  - [Complex Type: PartyMessageInformation](#)
  - [Complex Type: ProblemLocation](#)
  - [Complex Type: Reason](#)
  - [Complex Type: ReasonCode](#)
  - [Complex Type: RequestEventStatus](#)
  - [Complex Type: RequestMessage](#)
  - [Complex Type: RequestMessageHeader](#)
  - [Complex Type: ResponseMessage](#)
  - [Complex Type: ResponseMessageHeader](#)
  - [Model Group: Correlation.model](#)
  - [Model Group: Exception.model](#)
  - [Model Group: MessageHeader.model](#)
  - [Model Group: OnBehalfOf.model](#)
  - [Model Group: Sequence.model](#)
- [Legend](#)
- [Glossary](#)

[top](#)

## Schema Document Properties

<b>Target Namespace</b>	<a href="http://www.fpml.org/FpML-5/confirmation">http://www.fpml.org/FpML-5/confirmation</a>
<b>Version</b>	\$Revision: 2864 \$

<b>Element and Attribute Namespaces</b>	<ul style="list-style-type: none"> <li>Global element and attribute declarations belong to this schema's target namespace.</li> <li>By default, local element declarations belong to this schema's target namespace.</li> <li>By default, local attribute declarations have no namespace.</li> </ul>
<b>Schema Composition</b>	<ul style="list-style-type: none"> <li>This schema imports schema(s) from the following namespace(s):           <ul style="list-style-type: none"> <li><a href="http://www.w3.org/2000/09/xmldsig#">http://www.w3.org/2000/09/xmldsig#</a> (at <a href="#">xmldsig-core-schema.xsd</a>)</li> </ul> </li> <li>This schema includes components from the following schema document(s):           <ul style="list-style-type: none"> <li><a href="#">fpml-doc-5-0.xsd</a></li> </ul> </li> </ul>
<b>Documentation</b>	Event Status messages.

## Declared Namespaces

Prefix	Namespace
Default namespace	<a href="http://www.fpml.org/FpML-5/confirmation">http://www.fpml.org/FpML-5/confirmation</a>
xml	http://www.w3.org/XML/1998/namespace
dsig	http://www.w3.org/2000/09/xmldsig#
fpml-annotation	http://www.fpml.org/annotation
xsd	http://www.w3.org/2001/XMLSchema
ecore	http://www.eclipse.org/emf/2002/Ecore

### Schema Component Representation

```

<xsd:schema targetNamespace="http://www.fpml.org/FpML-5/confirmation"
documentRoot="FpML" nsPrefix="conf" package="org.fpml.confirmation" version="$Revision: 2864
$" elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xsd:import namespace="http://www.w3.org/2000/09/xmldsig#" schemaLocation="xmldsig-core-
  schema.xsd"/>
  <xsd:include schemaLocation="fpml-doc-5-0.xsd"/>
  ...
</xsd:schema>

```

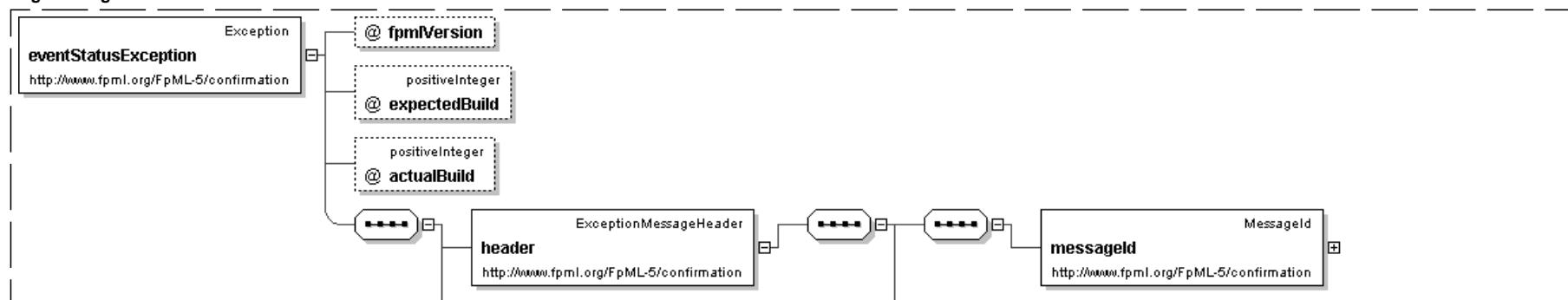
[top](#)

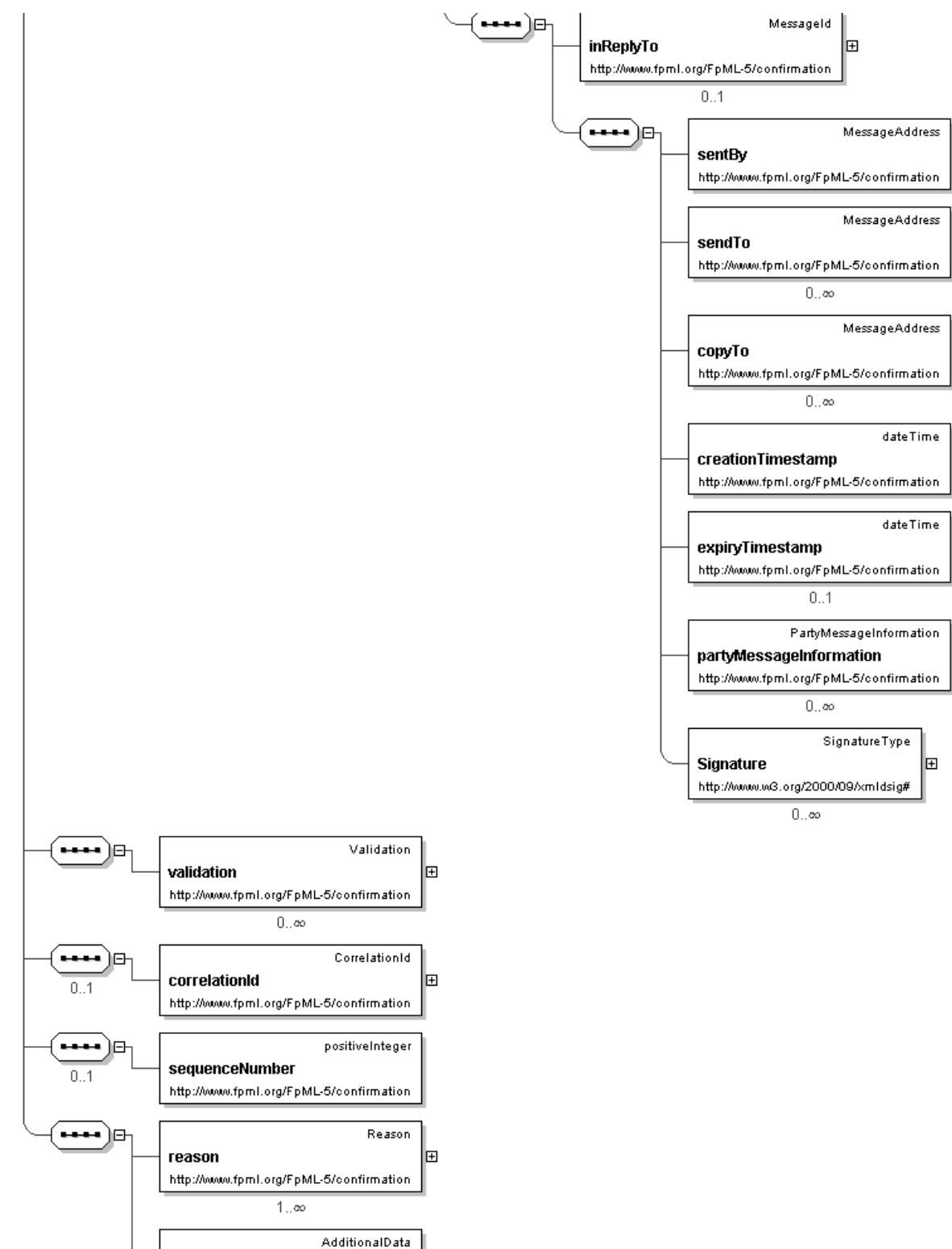
## Global Declarations

### Element: eventStatusException

<b>Name</b>	eventStatusException
<b>Type</b>	<a href="#">Exception</a>
<b>Nullable</b>	no
<b>Abstract</b>	no

### Logical Diagram





**additionalData**  
http://www.fpml.org/FpML-5/confirmation

0..1

**XML Instance Representation**

```

<eventStatusException
fpmlVersion=" xsd:token (value comes from list: {'5-0'}) [1]
'Indicate which version of the FpML Schema an FpML message adheres to.'

"
expectedBuild=" xsd:positiveInteger [0..1]
'This optional attribute can be supplied by a message creator in an FpML instance to
specify which build number of the schema was used to define the message when it was generated.'

"
actualBuild="8 [0..1]
'The specific build number of this schema version. This attribute is not included in
an instance document. Instead, it is supplied by the XML parser when the document is
validated against the FpML schema and indicates the build number of the schema file. Every
time FpML publishes a change to the schema, validation rules, or examples within a version
(e.g., version 4.2) the actual build number is incremented. If no changes have been
made between releases within a version (i.e. from Trial Recommendation to Recommendation)
the actual build number stays the same.'

">
    <header> ExceptionMessageHeader </header> [1]
    <validation> Validation </validation> [0..*]
Start Group: Correlation.model [0..1]
    <correlationId> CorrelationId </correlationId> [1]
    'A qualified identifier used to correlate between messages'

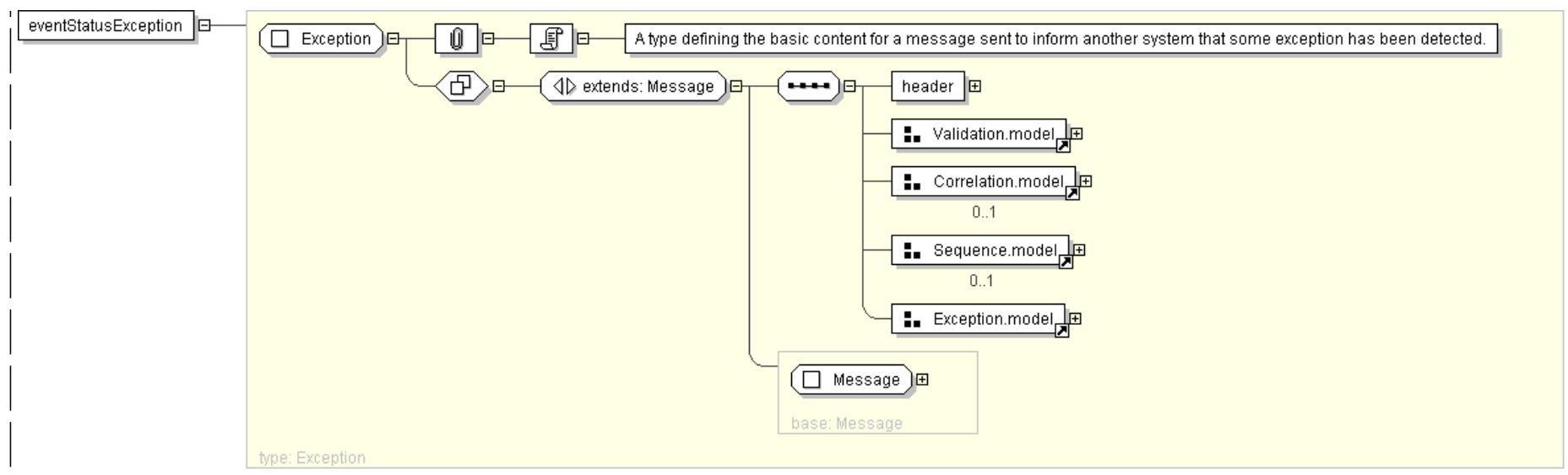
End Group: Correlation.model
Start Group: Sequence.model [0..1]
    <sequenceNumber> xsd:positiveInteger </sequenceNumber> [1]
    'A numeric value that can be used to order messages with the same correlation identifier
from the same sender.'

End Group: Sequence.model
    <reason> Reason </reason> [1..*]
    'An instance of the Reason type used to record the nature of any errors associated with
a message.'

    <additionalData> AdditionalData </additionalData> [0..1]
    'Any string of additional data that may help the message processor, for example in a
rejection message this might contain a code value or the text of the original request (within
a CDATA section).'

</eventStatusException>
```

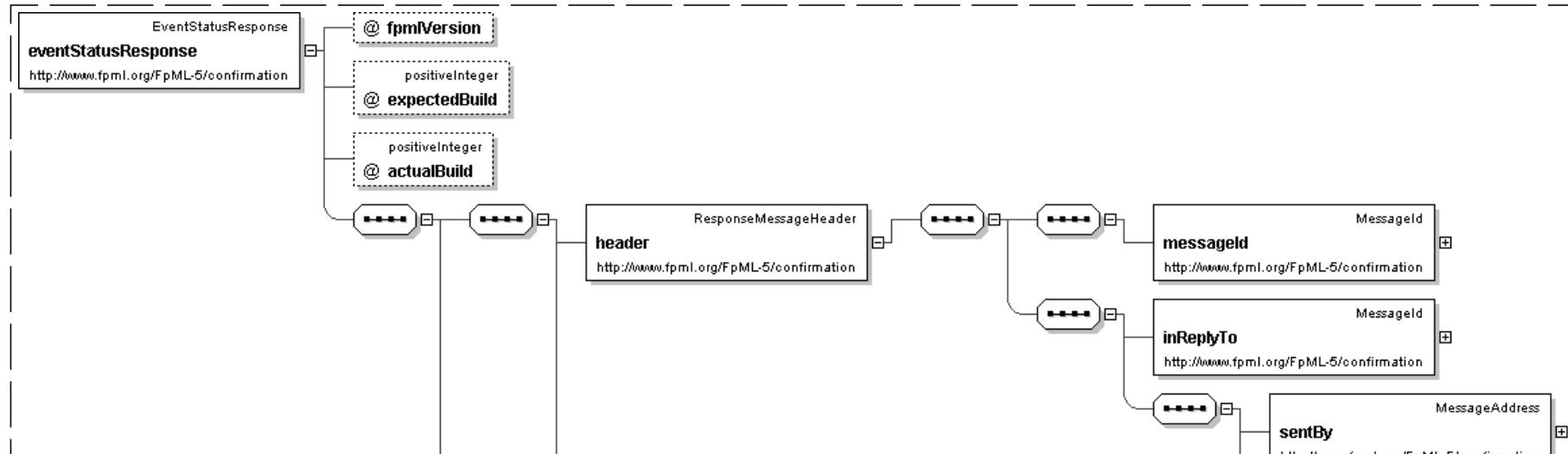
**Diagram**

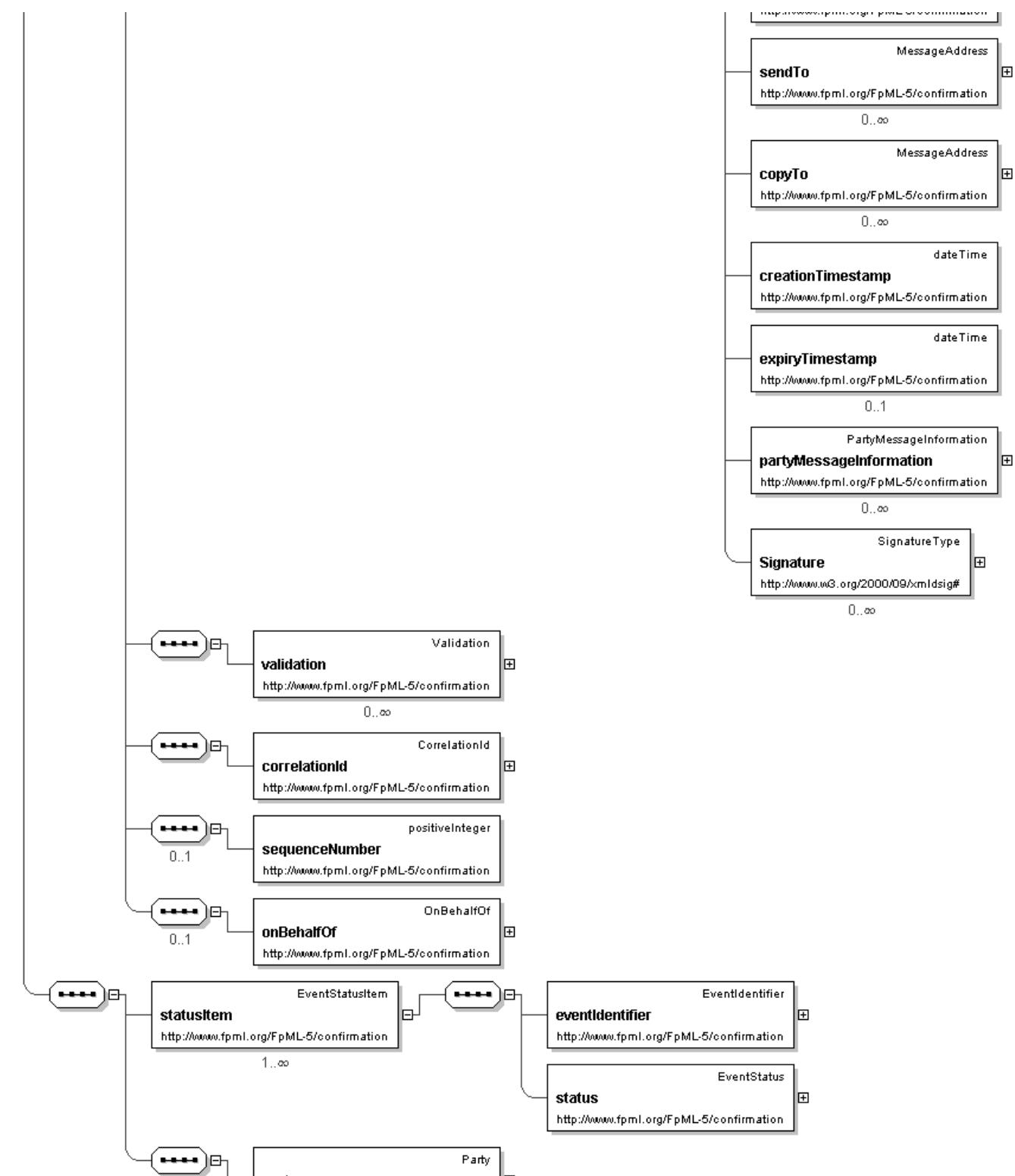
**Schema Component Representation**

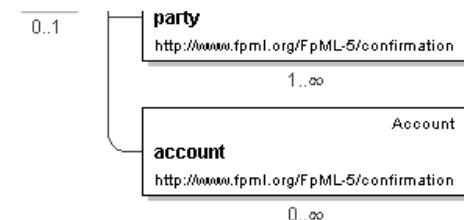
```
<xsd:element name="eventStatusException" type="Exception" />
```

[top](#)**Element: eventStatusResponse**

Name	eventStatusResponse
Type	<a href="#">EventStatusResponse</a>
Nillable	no
Abstract	no

**Logical Diagram**



**XML Instance Representation**

```

<eventStatusResponse
fpmlVersion=" xsd:token (value comes from list: {'5-0'}) [1]
'Indicate which version of the FpML Schema an FpML message adheres to.'

"
expectedBuild=" xsd:positiveInteger [0..1]
'This optional attribute can be supplied by a message creator in an FpML instance to
specify which build number of the schema was used to define the message when it was generated.'

"
actualBuild="8 [0..1]
'The specific build number of this schema version. This attribute is not included in
an instance document. Instead, it is supplied by the XML parser when the document is
validated against the FpML schema and indicates the build number of the schema file. Every
time FpML publishes a change to the schema, validation rules, or examples within a version
(e.g., version 4.2) the actual build number is incremented. If no changes have been
made between releases within a version (i.e. from Trial Recommendation to Recommendation)
the actual build number stays the same.'

">
<header> ResponseMessageHeader </header> [1]
<validation> Validation </validation> [0..*]
<correlationId> CorrelationId </correlationId> [1]
'A qualified identifier used to correlate between messages'

Start Group: Sequence.model [0..1]
<sequenceNumber> xsd:positiveInteger </sequenceNumber> [1]
'A numeric value that can be used to order messages with the same correlation identifier
from the same sender.'

End Group: Sequence.model
Start Group: OnBehalfOf.model [0..1]
<onBehalfOf> OnBehalfOf </onBehalfOf> [1]
'Indicates which party (and accounts) a trade is being processed for.'

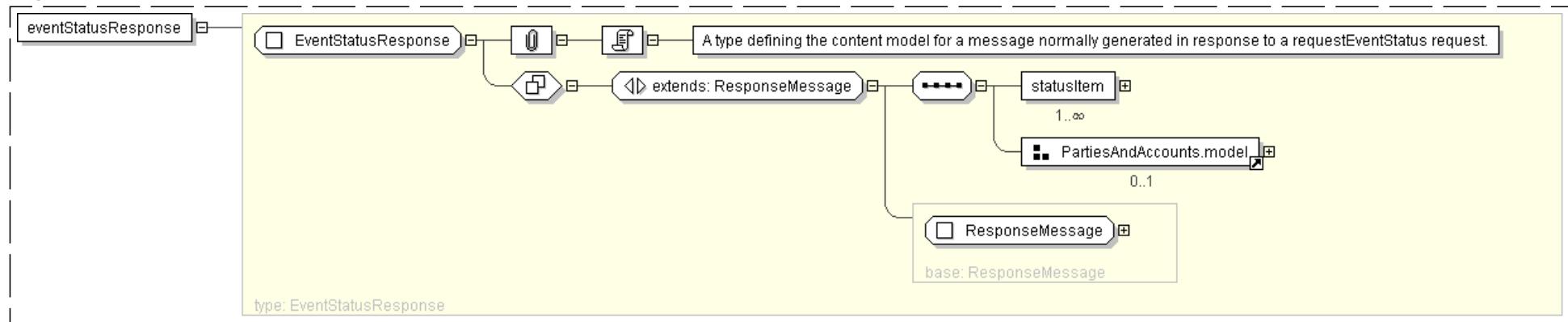
End Group: OnBehalfOf.model
<statusItem> EventStatusItem </statusItem> [1..*]
Start Group: PartiesAndAccounts.model [0..1]
<party> Party </party> [1..*]
'A legal entity or a subdivision of a legal entity.', 'Parties can perform multiple roles in
a trade lifecycle. For example, the principal parties obligated to make payments from time
to time during the term of the trade, but may include other parties involved in, or
incidental to, the trade, such as parties acting in the role of novation transferor/
transferee, broker, calculation agent, etc. In FpML roles are defined in multiple places
within a document.'

<account> Account </account> [0..*]

```

'Optional account information used to precisely define the origination and destination of financial instruments.'

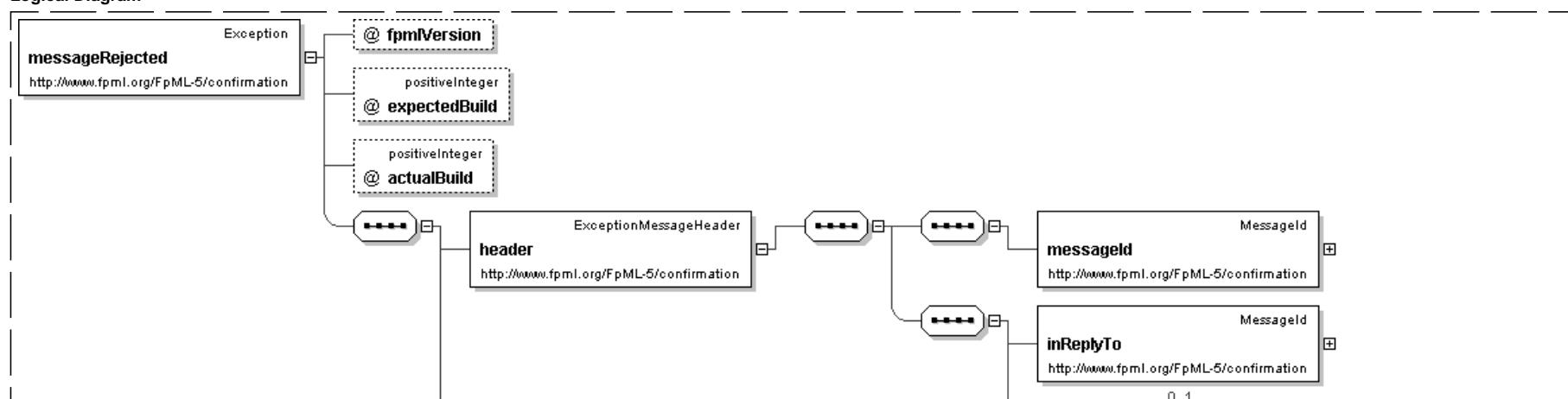
End Group: [PartiesAndAccounts.model](#)  
 </eventStatusResponse>

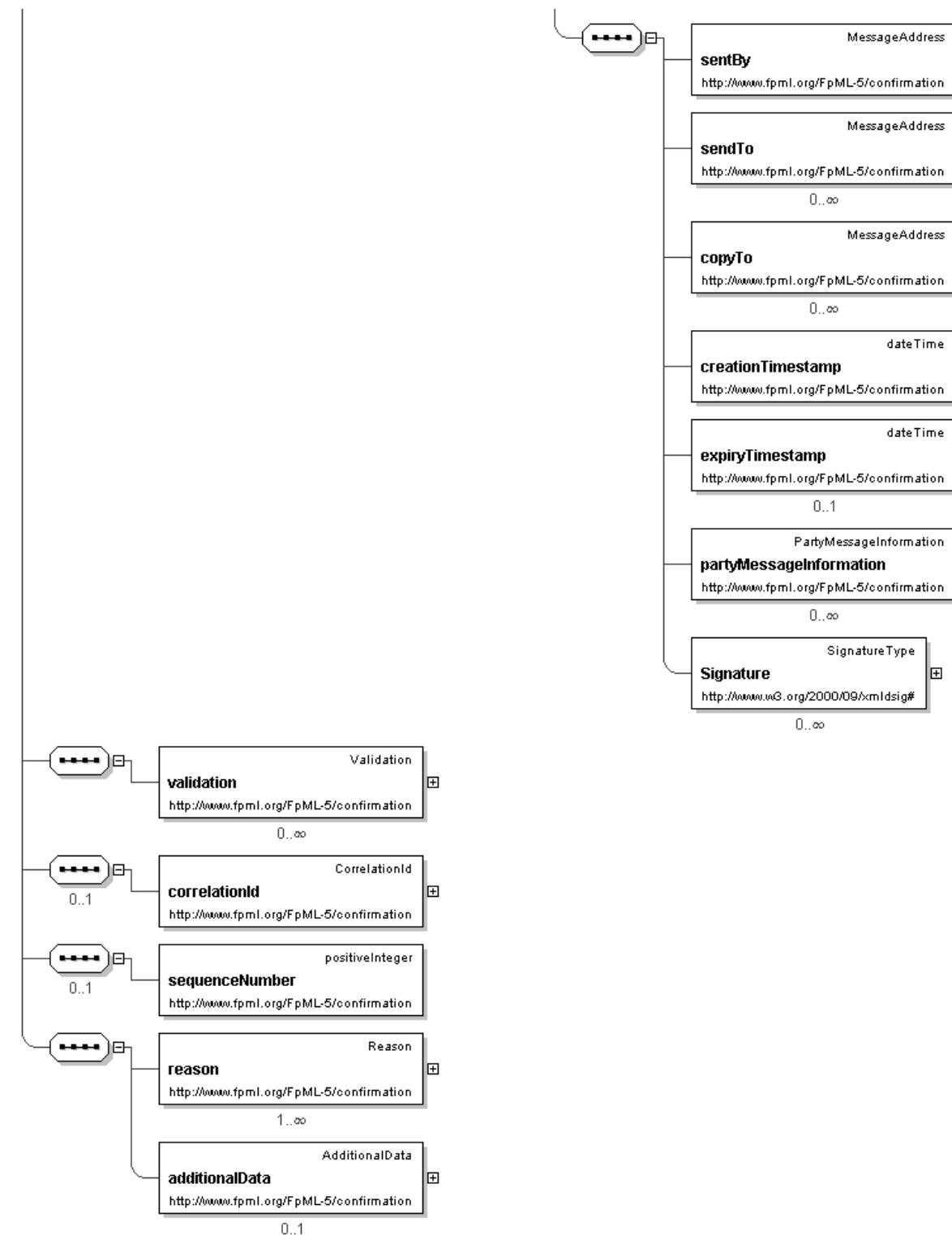
**Diagram****Schema Component Representation**

```
<xsd:element name="eventStatusResponse" type=" EventStatusResponse " />
```

[top](#)**Element: messageRejected**

Name	messageRejected
Type	<a href="#">Exception</a>
Nillable	no
Abstract	no
Documentation	The root element used for rejected message exceptions

**Logical Diagram**



**XML Instance Representation**

```

<messageRejected
  fpmVersion=" xsd:token (value comes from list: {'5-0'}) [1]
  'Indicate which version of the FpML Schema an FpML message adheres to.'

  "
  expectedBuild=" xsd:positiveInteger [0..1]
  'This optional attribute can be supplied by a message creator in an FpML instance to
  specify which build number of the schema was used to define the message when it was generated.'

  "
  actualBuild="8 [0..1]
  'The specific build number of this schema version. This attribute is not included in
  an instance document. Instead, it is supplied by the XML parser when the document is
  validated against the FpML schema and indicates the build number of the schema file. Every
  time FpML publishes a change to the schema, validation rules, or examples within a version
  (e.g., version 4.2) the actual build number is incremented. If no changes have been
  made between releases within a version (i.e. from Trial Recommendation to Recommendation)
  the actual build number stays the same.'


  ">
    <header> ExceptionMessageHeader </header> [1]
    <validation> Validation </validation> [0..*]
    Start Group: Correlation.model [0..1]
      <correlationId> CorrelationId </correlationId> [1]
      'A qualified identifier used to correlate between messages'

    End Group: Correlation.model
    Start Group: Sequence.model [0..1]
      <sequenceNumber> xsd:positiveInteger </sequenceNumber> [1]
      'A numeric value that can be used to order messages with the same correlation identifier
      from the same sender.'

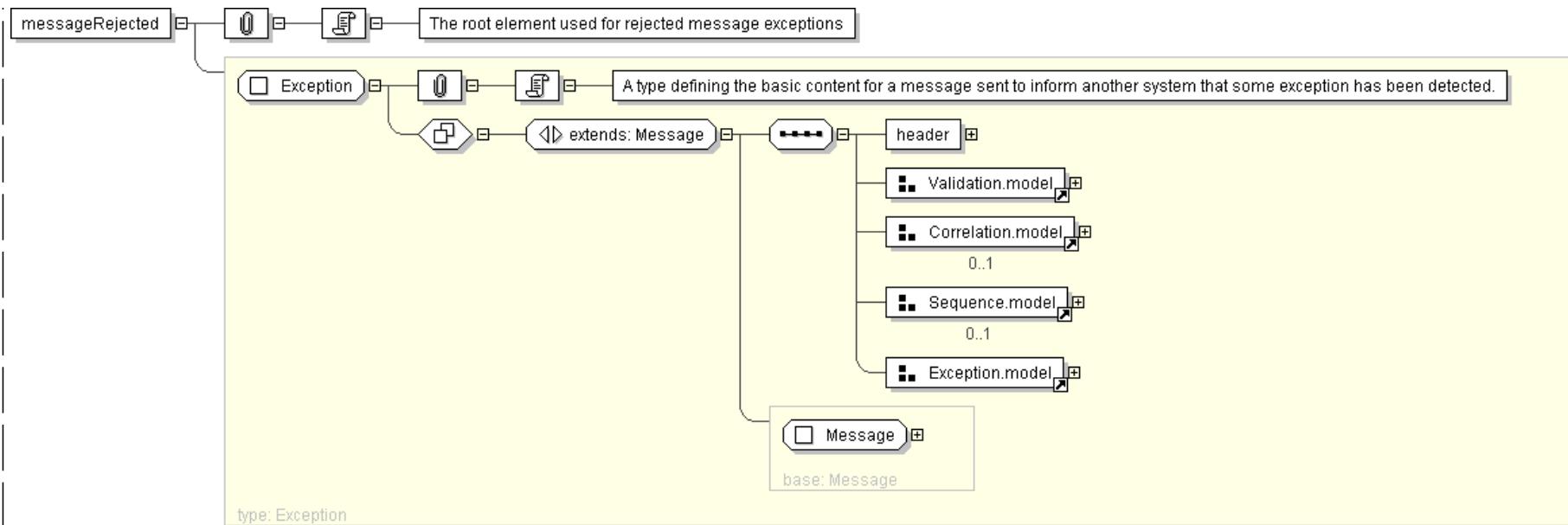
    End Group: Sequence.model
      <reason> Reason </reason> [1..*]
      'An instance of the Reason type used to record the nature of any errors associated with
      a message.'

      <additionalData> AdditionalData </additionalData> [0..1]
      'Any string of additional data that may help the message processor, for example in a
      rejection message this might contain a code value or the text of the original request (within
      a CDATA section).'

    </messageRejected>

```

**Diagram**



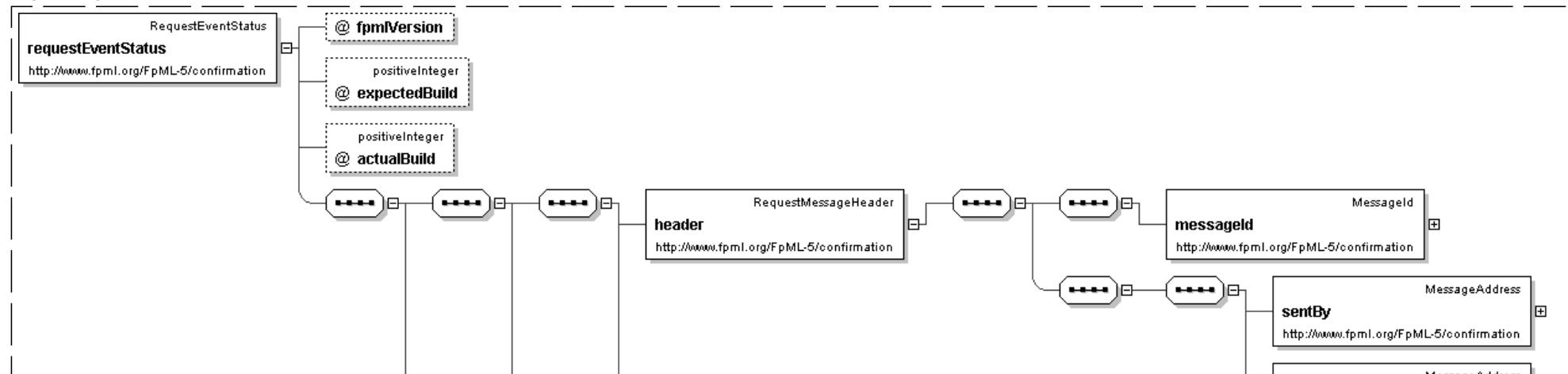
type: Exception

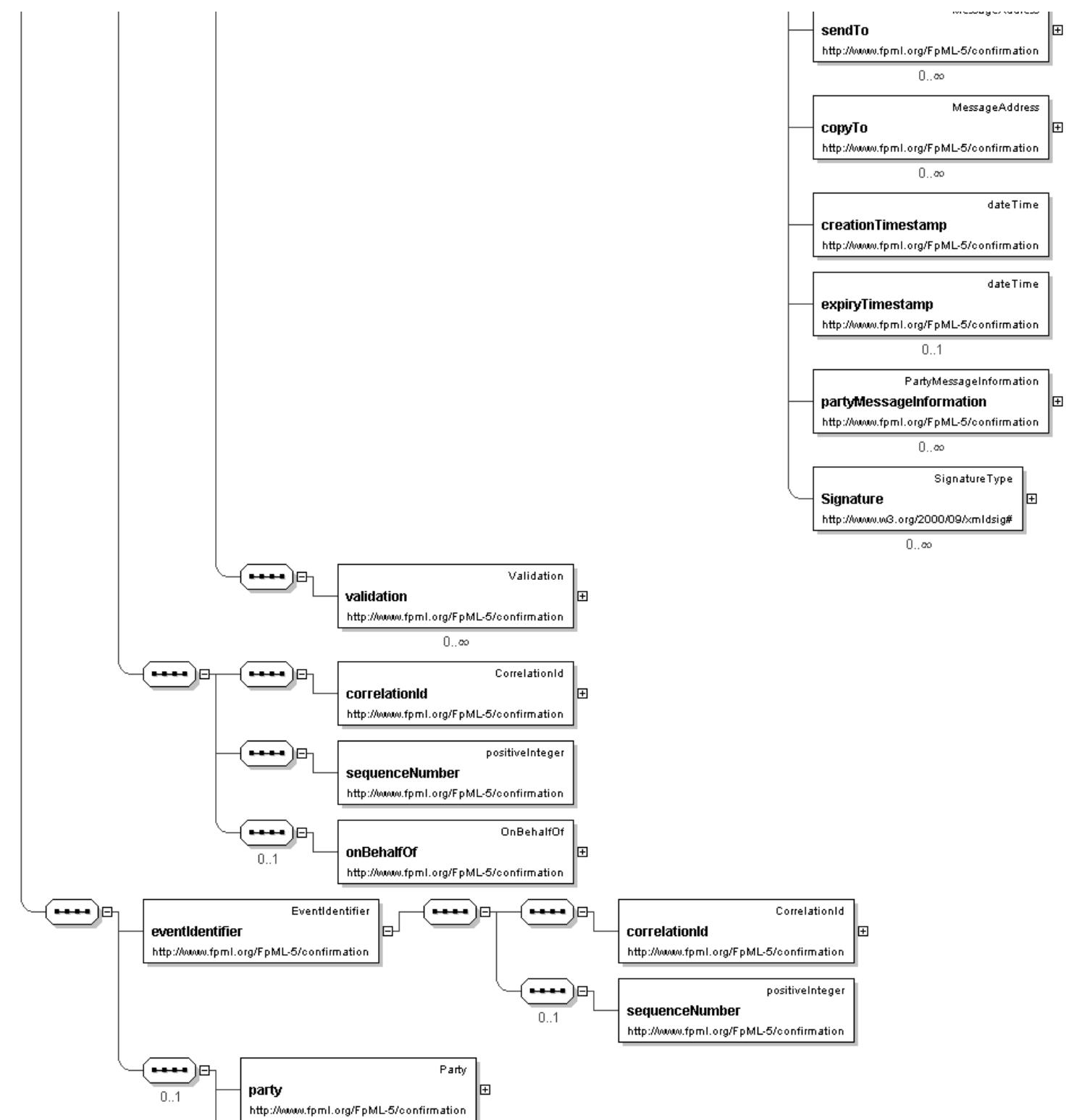
**Schema Component Representation**

```
<xsd:element name="messageRejected" type="Exception" />
```

[top](#)**Element: requestEventStatus**

Name	requestEventStatus
Type	<a href="#">RequestEventStatus</a>
Nillable	no
Abstract	no

**Logical Diagram**



**XML Instance Representation**

```

<requestEventStatus
  fpmlVersion=" xsd:token (value comes from list: {'5-0'}) [1]
  'Indicate which version of the FpML Schema an FpML message adheres to.'>

  "
  expectedBuild=" xsd:positiveInteger [0..1]
  'This optional attribute can be supplied by a message creator in an FpML instance to
  specify which build number of the schema was used to define the message when it was generated.'

  "
  actualBuild="8 [0..1]
  'The specific build number of this schema version. This attribute is not included in
  an instance document. Instead, it is supplied by the XML parser when the document is
  validated against the FpML schema and indicates the build number of the schema file. Every
  time FpML publishes a change to the schema, validation rules, or examples within a version
  (e.g., version 4.2) the actual build number is incremented. If no changes have been
  made between releases within a version (i.e. from Trial Recommendation to Recommendation)
  the actual build number stays the same.'

  ">
  <header> RequestMessageHeader </header> [1]
  <validation> Validation </validation> [0..*]
  <correlationId> CorrelationId </correlationId> [1]
  'A qualified identifier used to correlate between messages'

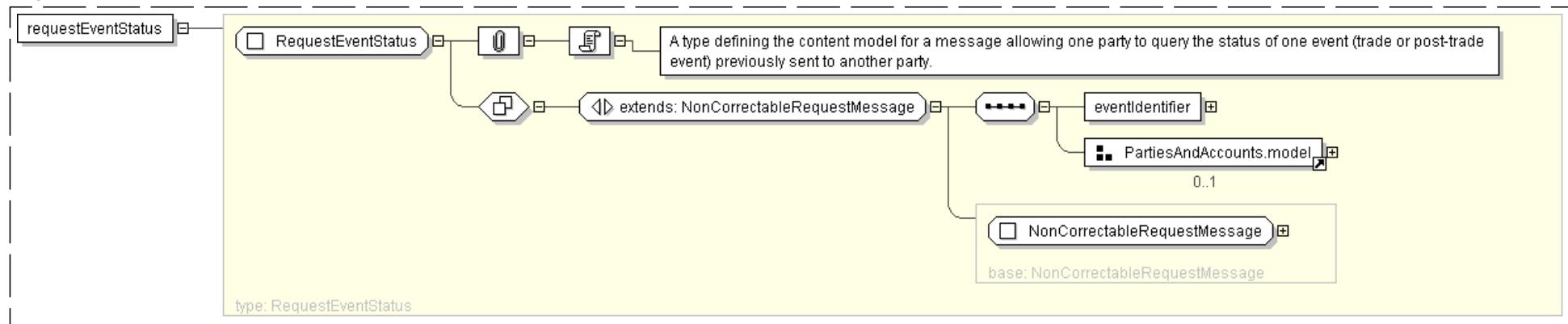
  <sequenceNumber> xsd:positiveInteger </sequenceNumber> [1]
  'A numeric value that can be used to order messages with the same correlation identifier
  from the same sender.'

Start Group: OnBehalfOf.model [0..1]
  <onBehalfOf> OnBehalfOf </onBehalfOf> [1]
  'Indicates which party (and accounts) a trade is being processed for.'

End Group: OnBehalfOf.model
  <eventIdentifier> EventIdentifier </eventIdentifier> [1]
Start Group: PartiesAndAccounts.model [0..1]
  <party> Party </party> [1..*]
  'A legal entity or a subdivision of a legal entity.', 'Parties can perform multiple roles in
  a trade lifecycle. For example, the principal parties obligated to make payments from time
  to time during the term of the trade, but may include other parties involved in, or
  incidental to, the trade, such as parties acting in the role of novation transferor/
  transferee, broker, calculation agent, etc. In FpML roles are defined in multiple places
  within a document.'

  <account> Account </account> [0..*]
  'Optional account information used to precisely define the origination and destination
  of financial instruments.'
  
```

```
| End Group: PartiesAndAccounts.model
</requestEventStatus>
```

**Diagram****Schema Component Representation**

```
<xsd:element name="requestEventStatus" type=" RequestEventStatus " />
```

[top](#)**Global Definitions****Complex Type: Acknowledgement**

<b>Super-types:</b>	<a href="#">Document</a> < <a href="#">Message</a> (by extension) < <a href="#">ResponseMessage</a> (by extension) < <b>Acknowledgement</b> (by extension)
---------------------	--

<b>Sub-types:</b>	None
-------------------	------

<b>Name</b>	Acknowledgement
-------------	-----------------

<b>Abstract</b>	no
-----------------	----

**XML Instance Representation**

```

<...
fpmVersion=" xsd:token (value comes from list: {'5-0'}) [1]
'Indicate which version of the FpML Schema an FpML message adheres to.'

"
expectedBuild=" xsd:positiveInteger [0..1]
'This optional attribute can be supplied by a message creator in an FpML instance to
specify which build number of the schema was used to define the message when it was generated.'

"
actualBuild="8 [0..1]
```

'The specific build number of this schema version. This attribute is not included in an instance document. Instead, it is supplied by the XML parser when the document is validated against the FpML schema and indicates the build number of the schema file. Every time FpML publishes a change to the schema, validation rules, or examples within a version (e.g., version 4.2) the actual build number is incremented. If no changes have been made between releases within a version (i.e. from Trial Recommendation to Recommendation) the actual build number stays the same.'

```

">>
<header> ResponseMessageHeader </header> [1]
<validation> Validation </validation> [0..*]
<correlationId> CorrelationId </correlationId> [1]
'A qualified identifier used to correlate between messages'

Start Group: Sequence.model [0..1]
<sequenceNumber> xsd:positiveInteger </sequenceNumber> [1]
'A numeric value that can be used to order messages with the same correlation identifier
from the same sender.'

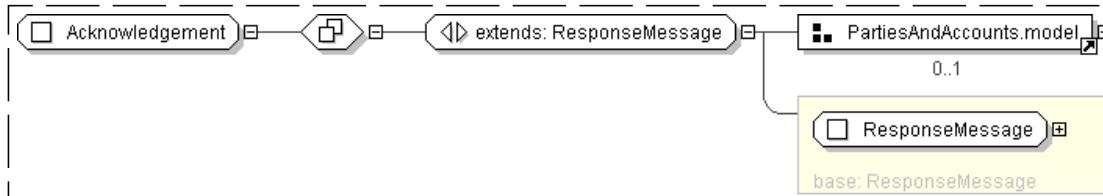
End Group: Sequence.model
Start Group: OnBehalfOf.model [0..1]
<onBehalfOf> OnBehalfOf </onBehalfOf> [1]
'Indicates which party (and accounts) a trade is being processed for.'

End Group: OnBehalfOf.model
Start Group: PartiesAndAccounts.model [0..1]
<party> Party </party> [1..*]
'A legal entity or a subdivision of a legal entity.', 'Parties can perform multiple roles in
a trade lifecycle. For example, the principal parties obligated to make payments from time
to time during the term of the trade, but may include other parties involved in, or
incidental to, the trade, such as parties acting in the role of novation transferor/
transferee, broker, calculation agent, etc. In FpML roles are defined in multiple places
within a document.'

<account> Account </account> [0..*]
'Optional account information used to precisely define the origination and destination
of financial instruments.'

End Group: PartiesAndAccounts.model
</...>

```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="Acknowledgement">
  <xsd:complexContent>
    <xsd:extension base="ResponseMessage">
      <xsd:group ref="PartiesAndAccounts.model" minOccurs="0"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

**Complex Type: AdditionalData**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None

<b>Name</b>	AdditionalData
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">Reason</a> , Model Group <a href="#">Exception.model</a>
<b>Abstract</b>	no
<b>Documentation</b>	Provides extra information not represented in the model that may be useful in processing the message i.e. diagnosing the reason for failure.

**XML Instance Representation**

```
<...>
<mimeType> MimeType </mimeType> [1]
'Indicates the type of media used to provide the extra information. mimeType is used to determine the software product(s) that can read the content. MIME Types are described in RFC 2046.'

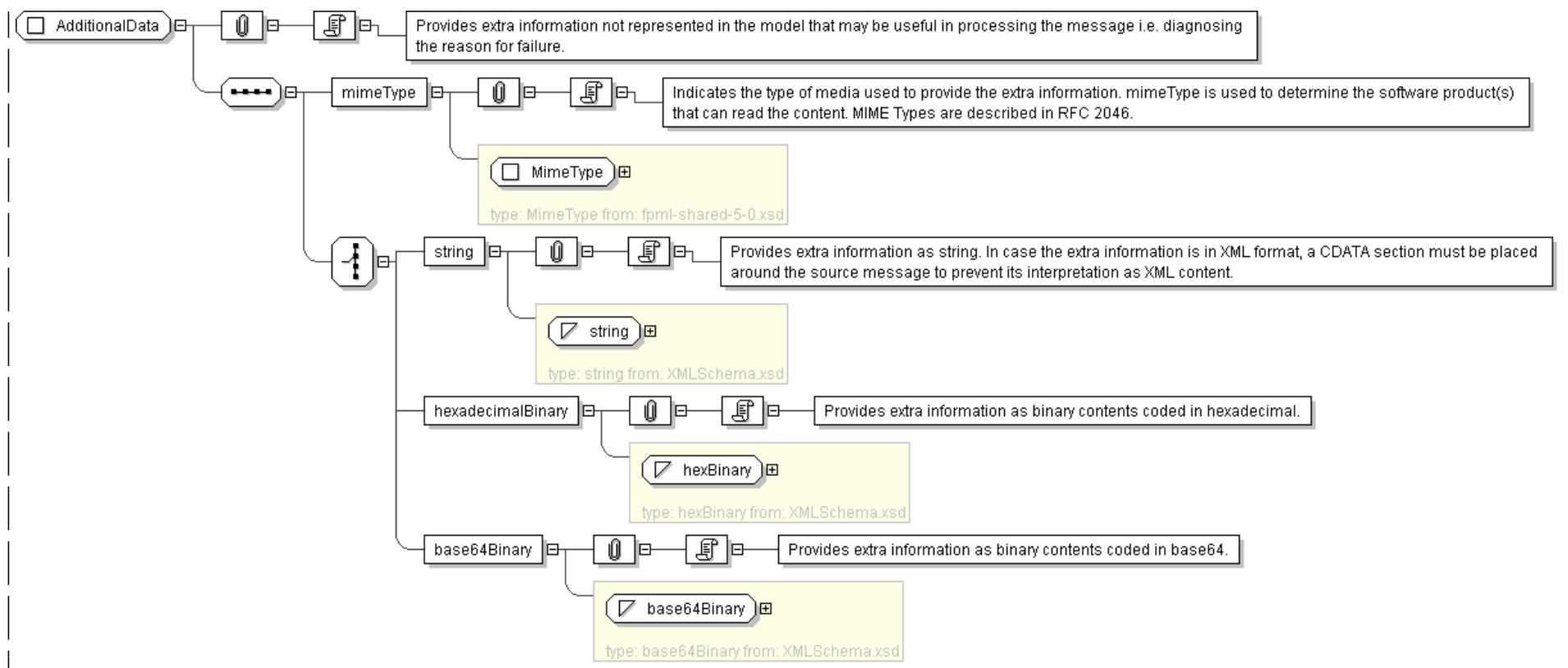
Start Choice [1]
<string> xsd:string </string> [1]
'Provides extra information as string. In case the extra information is in XML format, a CDATA section must be placed around the source message to prevent its interpretation as XML content.'

<hexadecimalBinary> xsd:hexBinary </hexadecimalBinary> [1]
'Provides extra information as binary contents coded in hexadecimal.'

<base64Binary> xsd:base64Binary </base64Binary> [1]
'Provides extra information as binary contents coded in base64.'

End Choice
</...>
```

**Diagram**



#### Schema Component Representation

```

<xsd:complexType name="AdditionalData">
  <xsd:sequence>
    <xsd:element name="MimeType" type="#MimeType" />
    <xsd:choice>
      <xsd:element name="string" type="xsd:string" />
      <xsd:element name="hexadecimalBinary" type="xsd:hexBinary" />
      <xsd:element name="base64Binary" type="xsd:base64Binary" />
    </xsd:choice>
  </xsd:sequence>
</xsd:complexType>

```

top

#### Complex Type: `CorrectableRequestMessage`

Super-types:	<a href="#">Document</a> < <a href="#">Message</a> (by extension) < <a href="#">RequestMessage</a> (by extension) < <b>CorrectableRequestMessage</b> (by extension)
Sub-types:	None

Name	CorrectableRequestMessage
Abstract	no
Documentation	A type defining the content model for a request message that can be subsequently corrected or retracted.

#### XML Instance Representation

```

<...
  fpmlVersion=" xsd:token (value comes from list: {'5-0'}) [1]
  'Indicate which version of the FpML Schema an FpML message adheres to.'

  "
  expectedBuild=" xsd:positiveInteger [0..1]
  'This optional attribute can be supplied by a message creator in an FpML instance to
  specify which build number of the schema was used to define the message when it was generated.'

  "
  actualBuild="8 [0..1]
  'The specific build number of this schema version. This attribute is not included in
  an instance document. Instead, it is supplied by the XML parser when the document is
  validated against the FpML schema and indicates the build number of the schema file. Every
  time FpML publishes a change to the schema, validation rules, or examples within a version
  (e.g., version 4.2) the actual build number is incremented. If no changes have been
  made between releases within a version (i.e. from Trial Recommendation to Recommendation)
  the actual build number stays the same.'


">
  <header> RequestMessageHeader </header> [1]
  <validation> Validation </validation> [0..*]
  <isCorrection> xsd:boolean </isCorrection> [1]
  'Indicates if this message corrects an earlier request.'

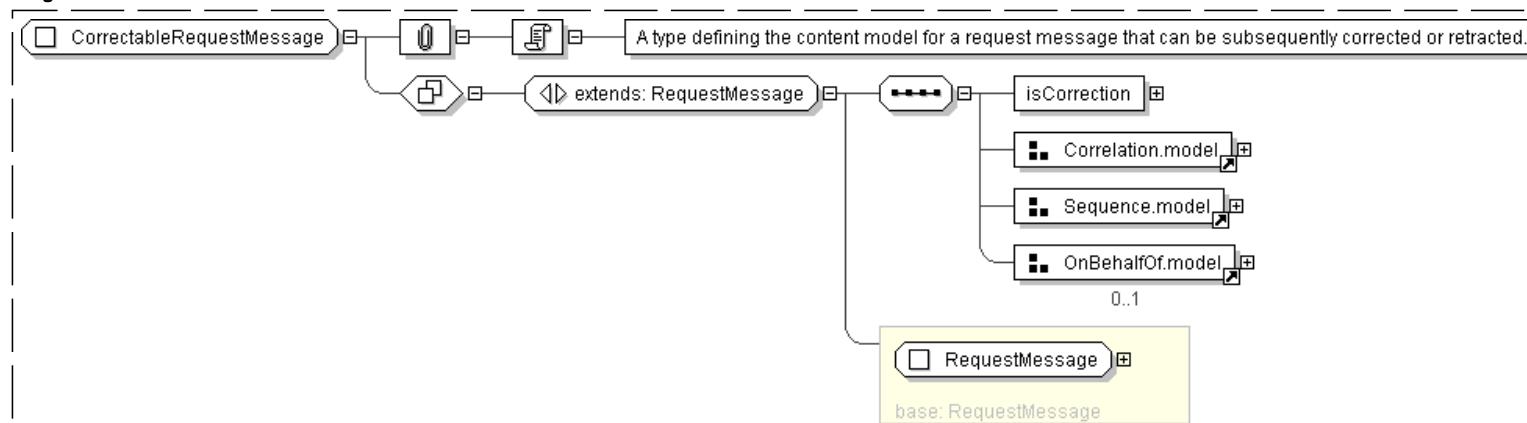

  <correlationId> CorrelationId </correlationId> [1]
  'A qualified identifier used to correlate between messages'


  <sequenceNumber> xsd:positiveInteger </sequenceNumber> [1]
  'A numeric value that can be used to order messages with the same correlation identifier
  from the same sender.'


Start Group: OnBehalfOf.model [0..1]
  <onBehalfOf> OnBehalfOf </onBehalfOf> [1]
  'Indicates which party (and accounts) a trade is being processed for.'


End Group: OnBehalfOf.model
</...>

```

**Diagram**

**Schema Component Representation**

```
<xsd:complexType name="CorrectableRequestMessage">
  <xsd:complexContent>
    <xsd:extension base=" RequestMessage ">
      <xsd:sequence>
        <xsd:element name="isCorrection" type=" xsd:boolean " />
        <xsd:group ref=" Correlation.model "/>
        <xsd:group ref=" Sequence.model "/>
        <xsd:group ref=" OnBehalfOf.model " minOccurs="0" />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

[top](#)**Complex Type: CorrelationId**

**Super-types:** [xsd:normalizedString](#) < **CorrelationId** (by extension)

**Sub-types:** None

**Name** CorrelationId

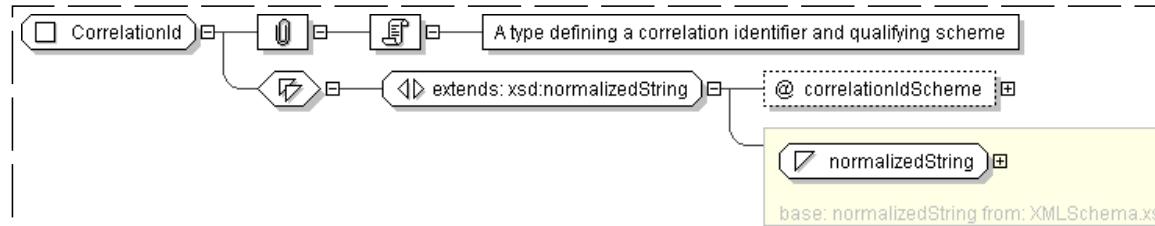
**Used by (from the same schema document)** Model Group [Correlation.model](#)

**Abstract** no

**Documentation** A type defining a correlation identifier and qualifying scheme

**XML Instance Representation**

```
<...>
<correlationIdscheme=" xsd:anyURI [1]">
  xsd:normalizedString
</...>
```

**Diagram****Schema Component Representation**

```
<xsd:complexType name="CorrelationId">
  <xsd:simpleContent>
    <xsd:extension base=" xsd:normalizedString ">
      <xsd:attribute name="correlationIdScheme" type=" xsd:anyURI " use="required"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
```

[top](#)

**Complex Type: EventIdentifier**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None

<b>Name</b>	EventIdentifier
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">EventStatusItem</a> , Complex Type <a href="#">RequestEventStatus</a>
<b>Abstract</b>	no
<b>Documentation</b>	Identification of a business event through its correlation id.

**XML Instance Representation**

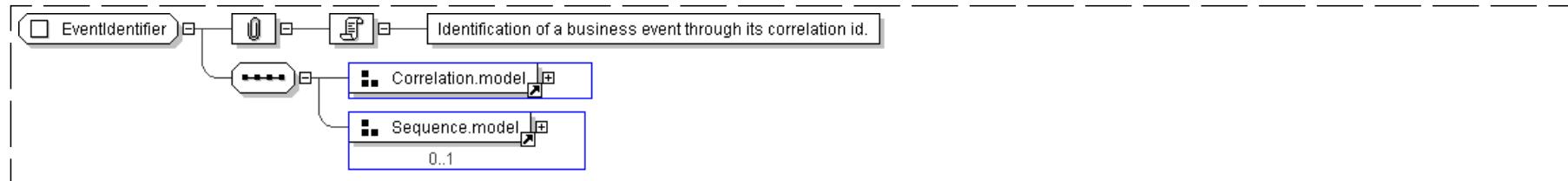
```

<...>
  <correlationId> CorrelationId </correlationId> [1]
    'A qualified identifier used to correlate between messages'

  Start Group: Sequence.model [0..1]
    <sequenceNumber> xsd:positiveInteger </sequenceNumber> [1]
      'A numeric value that can be used to order messages with the same correlation identifier
      from the same sender.'

  End Group: Sequence.model
</...>

```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="EventIdentifier">
  <xsd:sequence>
    <xsd:group ref="#" Correlation.model "/>
    <xsd:group ref="#" Sequence.model " minOccurs="0" />
  </xsd:sequence>
</xsd:complexType>

```

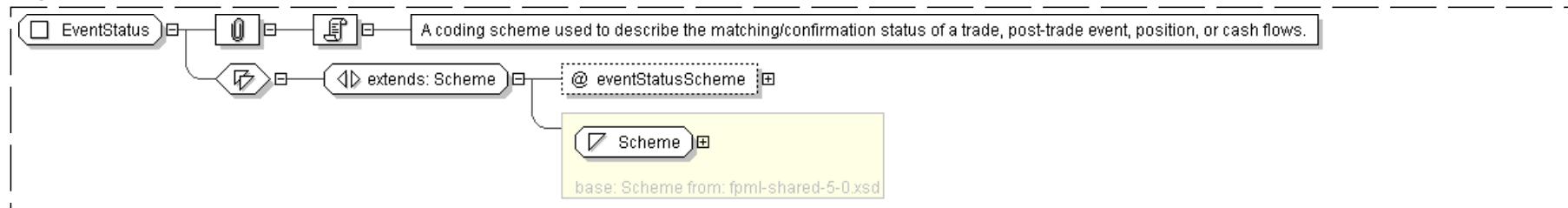
[top](#)**Complex Type: EventStatus**

<b>Super-types:</b>	<a href="#">Scheme</a> < <b>EventStatus</b> (by extension)
<b>Sub-types:</b>	None

<b>Name</b>	EventStatus
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">EventStatusItem</a>
<b>Abstract</b>	no
<b>Documentation</b>	A coding scheme used to describe the matching/confirmation status of a trade, post-trade event, position, or cash flows.

**XML Instance Representation**

```
<...>
<eventStatusScheme=" xsd:anyURI [0..1]">
  Scheme
</...>
```

**Diagram****Schema Component Representation**

```
<xsd:complexType name="EventStatus">
  <xsd:simpleContent>
    <xsd:extension base=" Scheme ">
      <xsd:attribute name="eventStatusScheme" type=" xsd:anyURI " default="http://www.fpml.org/
        coding-scheme/event-status" />
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
```

[top](#)**Complex Type: EventStatusItem**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None

<b>Name</b>	EventStatusItem
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">EventStatusResponse</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type used in event status enquiry messages which relates an event identifier to its current status value.

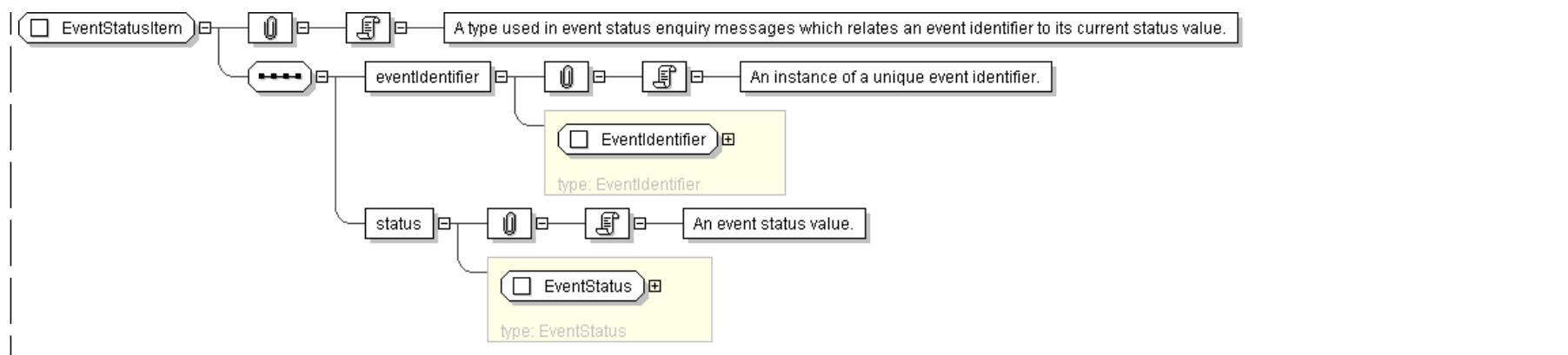
**XML Instance Representation**

```
<...>
  <eventIdentifier> EventIdentifier </eventIdentifier> [1]
  'An instance of a unique event identifier.'

  <status> EventStatus </status> [1]
  'An event status value.'

</...>
```

**Diagram**



### Schema Component Representation

```

<xsd:complexType name="EventStatusItem">
  <xsd:sequence>
    <xsd:element name="eventIdentifier" type="EventIdentifier" />
    <xsd:element name="status" type="EventStatus" />
  </xsd:sequence>
</xsd:complexType>
  
```

top

## Complex Type: `EventStatusResponse`

<b>Super-types:</b>	<a href="#">Document</a> < <a href="#">Message</a> (by extension) < <a href="#">ResponseMessage</a> (by extension) < <b>EventStatusResponse</b> (by extension)
<b>Sub-types:</b>	None

<b>Name</b>	EventStatusResponse
<b>Used by (from the same schema document)</b>	Element <a href="#">eventStatusResponse</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type defining the content model for a message normally generated in response to a requestEventStatus request.

### XML Instance Representation

```

<...
fpmVersion=" xsd:token (value comes from list: {'5-0'}) [1]
'Indicate which version of the FpML Schema an FpML message adheres to.'

"
expectedBuild=" xsd:positiveInteger [0..1]
'This optional attribute can be supplied by a message creator in an FpML instance to
specify which build number of the schema was used to define the message when it was generated.'

"
actualBuild="8 [0..1]
'The specific build number of this schema version. This attribute is not included in
an instance document. Instead, it is supplied by the XML parser when the document is
validated against the FpML schema and indicates the build number of the schema file. Every
time FpML publishes a change to the schema, validation rules, or examples within a version
(e.g., version 4.2) the actual build number is incremented. If no changes have been
made between releases within a version (i.e. from Trial Recommendation to Recommendation)
the actual build number stays the same.'
  
```

```
">
  <header> ResponseMessageHeader </header> [1]
  <validation> Validation </validation> [0..*]
  <correlationId> CorrelationId </correlationId> [1]
    'A qualified identifier used to correlate between messages'

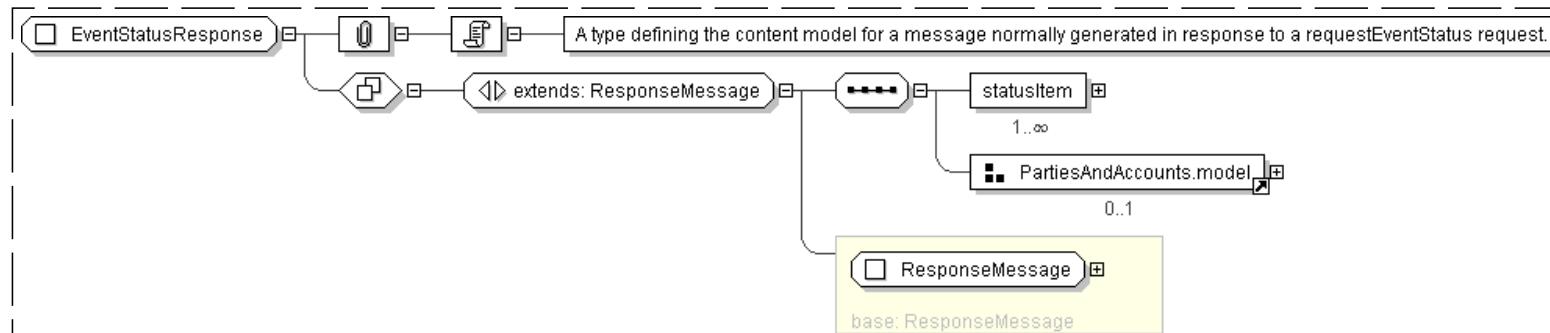
Start Group: Sequence.model [0..1]
  <sequenceNumber> xsd:positiveInteger </sequenceNumber> [1]
    'A numeric value that can be used to order messages with the same correlation identifier
    from the same sender.'

End Group: Sequence.model
Start Group: OnBehalfOf.model [0..1]
  <onBehalfOf> OnBehalfOf </onBehalfOf> [1]
    'Indicates which party (and accounts) a trade is being processed for.'

End Group: OnBehalfOf.model
  <statusItem> EventStatusItem </statusItem> [1..*]
Start Group: PartiesAndAccounts.model [0..1]
  <party> Party </party> [1..*]
    'A legal entity or a subdivision of a legal entity.', 'Parties can perform multiple roles in
    a trade lifecycle. For example, the principal parties obligated to make payments from time
    to time during the term of the trade, but may include other parties involved in, or
    incidental to, the trade, such as parties acting in the role of novation transferor/
    transferee, broker, calculation agent, etc. In FpML roles are defined in multiple places
    within a document.'

  <account> Account </account> [0..*]
    'Optional account information used to precisely define the origination and destination
    of financial instruments.'

End Group: PartiesAndAccounts.model
</...>
```

**Diagram****Schema Component Representation**

```
<xsd:complexType name="EventStatusResponse">
  <xsd:complexContent>
    <xsd:extension base=" ResponseMessage ">
      <xsd:sequence>
        <xsd:element name=" statusItem " type=" EventStatusItem " maxOccurs="unbounded" />
        <xsd:group ref=" PartiesAndAccounts.model " minOccurs="0" />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

```

</xsd:sequence>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>

```

## Complex Type: **Exception**

**Super-types:** [Document](#) < [Message](#) (by extension) < **Exception** (by extension)

**Sub-types:** None

<b>Name</b>	Exception
<b>Used by (from the same schema document)</b>	Element <a href="#">eventStatusException</a> , Element <a href="#">messageRejected</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type defining the basic content for a message sent to inform another system that some exception has been detected.

### XML Instance Representation

```

<...
fpmVersion=" xsd:token (value comes from list: {'5-0'}) [1]
'Indicate which version of the FpML Schema an FpML message adheres to.'

"
expectedBuild=" xsd:positiveInteger [0..1]
'This optional attribute can be supplied by a message creator in an FpML instance to
specify which build number of the schema was used to define the message when it was generated.'

"
actualBuild="8 [0..1]
'The specific build number of this schema version. This attribute is not included in
an instance document. Instead, it is supplied by the XML parser when the document is
validated against the FpML schema and indicates the build number of the schema file. Every
time FpML publishes a change to the schema, validation rules, or examples within a version
(e.g., version 4.2) the actual build number is incremented. If no changes have been
made between releases within a version (i.e. from Trial Recommendation to Recommendation)
the actual build number stays the same.'

">
<header> ExceptionMessageHeader </header> [1]
<validation> Validation </validation> [0..*]
Start Group: Correlation.model [0..1]
  <correlationId> CorrelationId </correlationId> [1]
    'A qualified identifier used to correlate between messages'

End Group: Correlation.model
Start Group: Sequence.model [0..1]
  <sequenceNumber> xsd:positiveInteger </sequenceNumber> [1]
    'A numeric value that can be used to order messages with the same correlation identifier
    from the same sender.'

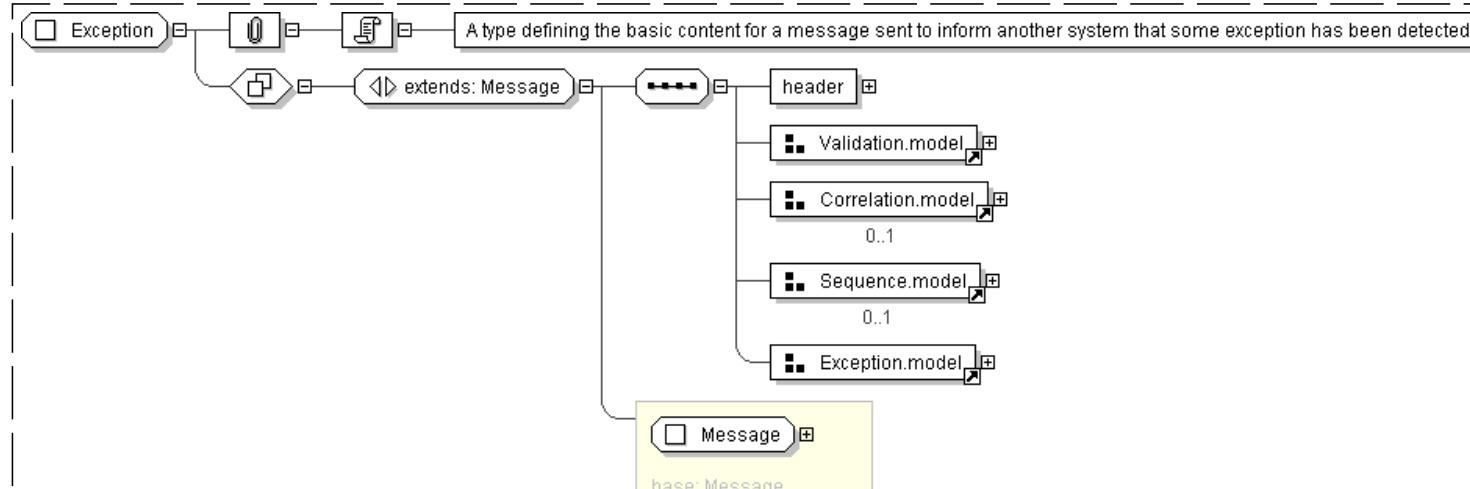
End Group: Sequence.model
<reason> Reason </reason> [1..*]
  'An instance of the Reason type used to record the nature of any errors associated with
  a message.'

```

```
<additionalData> AdditionalData </additionalData> [0..1]
```

'Any string of additional data that may help the message processor, for example in a rejection message this might contain a code value or the text of the original request (within a CDATA section).'

&lt;/...&gt;

**Diagram****Schema Component Representation**

```

<xsd:complexType name="Exception">
  <xsd:complexContent>
    <xsd:extension base="Message">
      <xsd:sequence>
        <xsd:element name="header" type="ExceptionMessageHeader" />
        <xsd:group ref="Validation.model" />
        <xsd:group ref="Correlation.model" minOccurs="0" />
        <xsd:group ref="Sequence.model" minOccurs="0" />
        <xsd:group ref="Exception.model" />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

[top](#)**Complex Type: ExceptionMessageHeader**

<b>Super-types:</b>	<a href="#">MessageHeader</a> < <b>ExceptionMessageHeader</b> (by extension)
<b>Sub-types:</b>	None

<b>Name</b>	ExceptionMessageHeader
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">Exception</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type defining the content model for an exception message header.

**XML Instance Representation**

```
<...>
< messageId> MessageId </messageId> [1]
'A unique identifier (within its coding scheme) assigned to the message by its creating party.'

<inReplyTo> MessageId </inReplyTo> [0..1]
'A copy of the unique message identifier (within its own coding scheme) to which this message is responding.'

<sentBy> MessageAddress </sentBy> [1]
'The unique identifier (within its coding scheme) for the originator of a message instance.'

<sendTo> MessageAddress </sendTo> [0..*]
'A unique identifier (within its coding scheme) indicating an intended recipient of a message.'

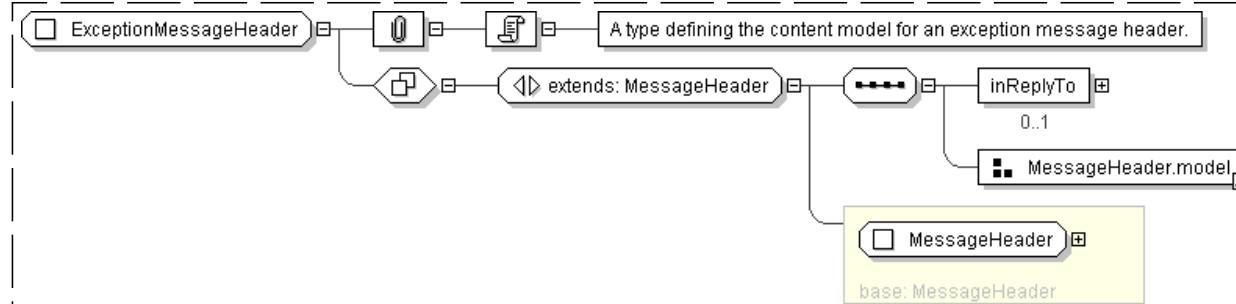
<copyTo> MessageAddress </copyTo> [0..*]
'A unique identifier (within the specified coding scheme) giving the details of some party to whom a copy of this message will be sent for reference.'

<creationTimestamp> xsd:dateTime </creationTimestamp> [1]
'The date and time (on the source system) when this message instance was created.'

<expiryTimestamp> xsd:dateTime </expiryTimestamp> [0..1]
'The date and time (on the source system) when this message instance will be considered expired.'

<partyMessageInformation> PartyMessageInformation </partyMessageInformation> [0..*]
'Additional message information that may be provided by each involved party.'

<dsig:Signature> ... </dsig:Signature> [0..*]
</...>
```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="ExceptionMessageHeader">
  <xsd:complexContent>
    <xsd:extension base=" MessageHeader ">
      <xsd:sequence>
        <xsd:element name="inReplyTo" type=" MessageId " minOccurs="0" />
        <xsd:group ref=" MessageHeader.model "/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>

```

&lt;/xsd:complexType&gt;

[top](#)

## Complex Type: Message

**Super-types:**
[Document](#) < **Message** (by extension)

**Sub-types:**

- [Exception](#) (by extension)
- [NotificationMessage](#) (by extension)
- [RequestMessage](#) (by extension)
  - [CorrectableRequestMessage](#) (by extension)
  - [NonCorrectableRequestMessage](#) (by extension)
    - [RequestEventStatus](#) (by extension)
- [ResponseMessage](#) (by extension)
  - [Acknowledgement](#) (by extension)
  - [EventStatusResponse](#) (by extension)

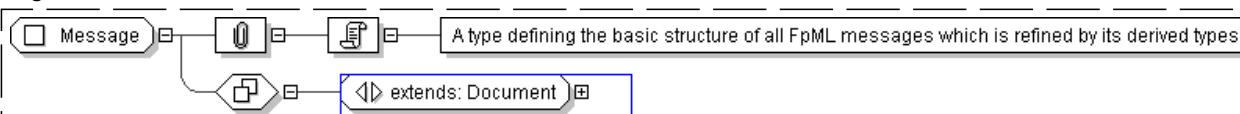
<b>Name</b>	Message
<b>Abstract</b>	yes
<b>Documentation</b>	A type defining the basic structure of all FpML messages which is refined by its derived types.

**XML Instance Representation**

```
<...>
  fpmlVersion=" xsd:token (value comes from list: {'5-0'}) [1]
  'Indicate which version of the FpML Schema an FpML message adheres to.'

  "
  expectedBuild=" xsd:positiveInteger [0..1]
  'This optional attribute can be supplied by a message creator in an FpML instance to
  specify which build number of the schema was used to define the message when it was generated.'

  "
  actualBuild="8 [0..1]
  'The specific build number of this schema version. This attribute is not included in
  an instance document. Instead, it is supplied by the XML parser when the document is
  validated against the FpML schema and indicates the build number of the schema file. Every
  time FpML publishes a change to the schema, validation rules, or examples within a version
  (e.g., version 4.2) the actual build number is incremented. If no changes have been
  made between releases within a version (i.e. from Trial Recommendation to Recommendation)
  the actual build number stays the same.'>
  "/>
```

**Diagram**

**Schema Component Representation**

```
<xsd:complexType name="Message" abstract="true">
  <xsd:complexContent>
    <xsd:extension base=" Document ">
  </xsd:complexContent>
```

## Complex Type: MessageAddress

**Super-types:** [Scheme](#) < **MessageAddress** (by extension)

**Sub-types:** None

**Name** MessageAddress

**Used by (from the same schema document)** Model Group [MessageHeader.model](#), Model Group [MessageHeader.model](#), Model Group [MessageHeader.model](#)

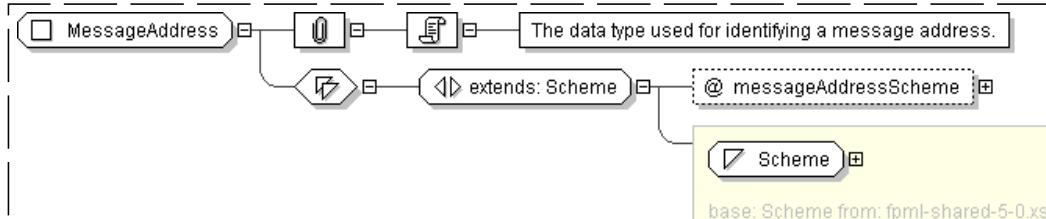
**Abstract** no

**Documentation** The data type used for identifying a message address.

### XML Instance Representation

```
<...>
<messageAddressScheme=" xsd:anyURI [ 0..1 ]>
<a href="#">Scheme
</...>
```

### Diagram



### Schema Component Representation

```

<xsd:complexType name="MessageAddress">
  <xsd:simpleContent>
    <xsd:extension base=" Scheme ">
      <xsd:attribute name="messageAddressScheme" type=" xsd:anyURI " />
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
  
```

## Complex Type: MessageHeader

**Super-types:** None

**Sub-types:**

- [ExceptionMessageHeader](#) (by extension)
- [NotificationMessageHeader](#) (by extension)
- [RequestMessageHeader](#) (by extension)
- [ResponseMessageHeader](#) (by extension)

**Name** MessageHeader

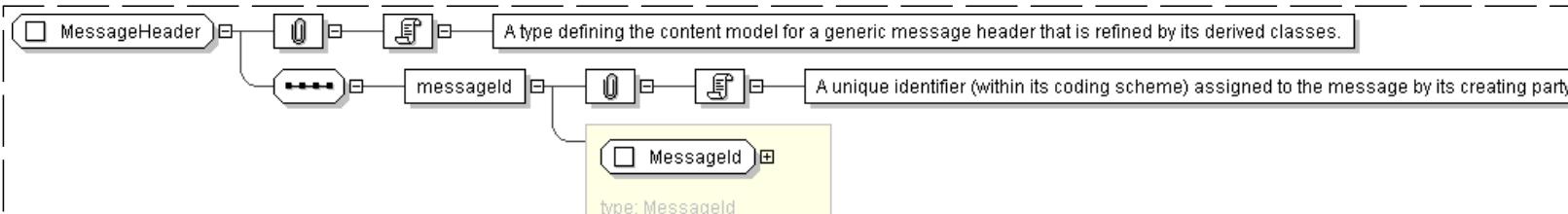
**Abstract** yes

**Documentation**

A type defining the content model for a generic message header that is refined by its derived classes.

**XML Instance Representation**

```
<...>
<messageId> MessageId </messageId> [1]
  'A unique identifier (within its coding scheme) assigned to the message by its creating party.'
</...>
```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="MessageHeader" abstract="true">
  <xsd:sequence>
    <xsd:element name="messageId" type=" MessageId " />
  </xsd:sequence>
</xsd:complexType>
  
```

[top](#)**Complex Type: [Messageld](#)**

<b>Super-types:</b>	<a href="#">Scheme</a> < <b>Messageld</b> (by extension)
---------------------	--

<b>Sub-types:</b>	None
-------------------	------

<b>Name</b>	Messageld
-------------	-----------

<b>Used by (from the same schema document)</b>	Complex Type <a href="#">ExceptionMessageHeader</a> , Complex Type <a href="#">MessageHeader</a> , Complex Type <a href="#">NotificationMessageHeader</a> , Complex Type <a href="#">ResponseMessageHeader</a>
--	--

<b>Abstract</b>	no
-----------------	----

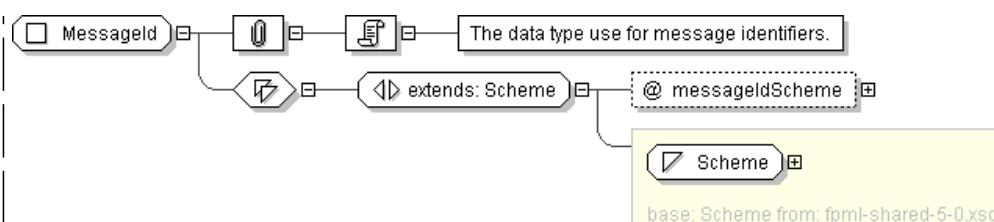
<b>Documentation</b>	The data type use for message identifiers.
----------------------	--

**XML Instance Representation**

```

<...
<messageIdScheme=" xsd:anyURI [1]">
  Scheme
</...>
  
```

**Diagram**

**Schema Component Representation**

```

<xsd:complexType name="MessageId">
  <xsd:simpleContent>
    <xsd:extension base=" Scheme ">
      <xsd:attribute name="messageIdScheme" type=" xsd:anyURI " use="required"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>

```

[top](#)**Complex Type: NonCorrectableRequestMessage**

<b>Super-types:</b>	<a href="#">Document</a> < <a href="#">Message</a> (by extension) < <a href="#">RequestMessage</a> (by extension) < <b>NonCorrectableRequestMessage</b> (by extension)
<b>Sub-types:</b>	• <a href="#">RequestEventStatus</a> (by extension)

<b>Name</b>	NonCorrectableRequestMessage
<b>Abstract</b>	no
<b>Documentation</b>	A type defining the content model for a request message that cannot be subsequently corrected or retracted.

**XML Instance Representation**

```

<...
  fpmiVersion=" xsd:token (value comes from list: {'5-0'}) [1]
  'Indicate which version of the FpML Schema an FpML message adheres to.'

  "
  expectedBuild=" xsd:positiveInteger [0..1]
  'This optional attribute can be supplied by a message creator in an FpML instance to
  specify which build number of the schema was used to define the message when it was generated.'

  "
  actualBuild="8 [0..1]
  'The specific build number of this schema version. This attribute is not included in
  an instance document. Instead, it is supplied by the XML parser when the document is
  validated against the FpML schema and indicates the build number of the schema file. Every
  time FpML publishes a change to the schema, validation rules, or examples within a version
  (e.g., version 4.2) the actual build number is incremented. If no changes have been
  made between releases within a version (i.e. from Trial Recommendation to Recommendation)
  the actual build number stays the same.'

  ">
  <header> RequestMessageHeader </header> [1]
  <validation> Validation </validation> [0..*]
  <correlationId> CorrelationId </correlationId> [1]
  'A qualified identifier used to correlate between messages'

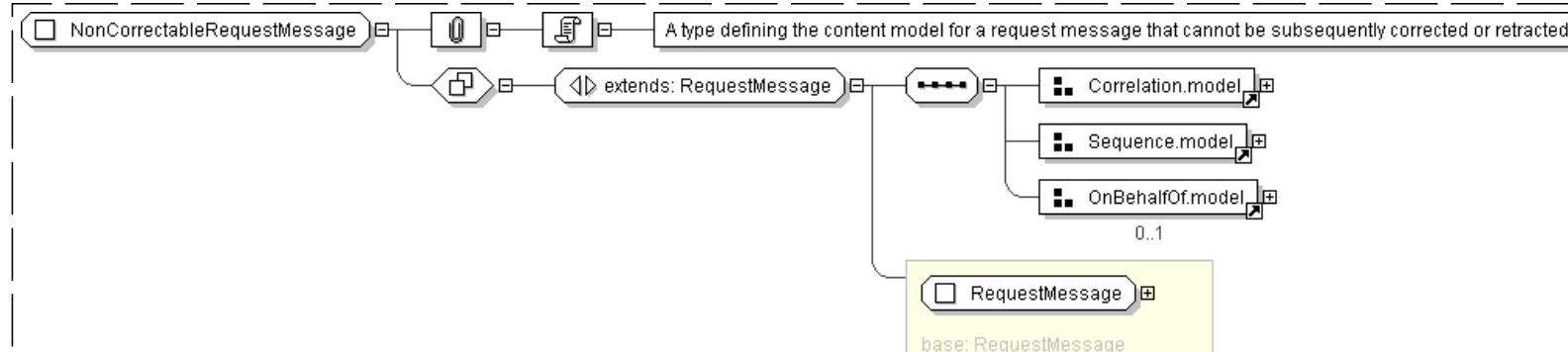
```

```
<sequenceNumber> xsd:positiveInteger </sequenceNumber> [1]
'A numeric value that can be used to order messages with the same correlation identifier
from the same sender.'
```

Start Group: OnBehalfOf.model [0..1]

```
<onBehalfOf> OnBehalfOf </onBehalfOf> [1]
'Indicates which party (and accounts) a trade is being processed for.'
```

End Group: OnBehalfOf.model  
</...>

**Diagram****Schema Component Representation**

```
<xsd:complexType name="NonCorrectableRequestMessage">
  <xsd:complexContent>
    <xsd:extension base=" RequestMessage ">
      <xsd:sequence>
        <xsd:group ref=" Correlation.model ">
        <xsd:group ref=" Sequence.model ">
        <xsd:group ref=" OnBehalfOf.model " minOccurs="0" />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

[top](#)**Complex Type: NotificationMessage**

<b>Super-types:</b>	<a href="#">Document</a> < <a href="#">Message</a> (by extension) < <b>NotificationMessage</b> (by extension)
<b>Sub-types:</b>	None
<b>Name</b>	NotificationMessage
<b>Abstract</b>	yes
<b>Documentation</b>	A type defining the basic content for a message sent to inform another system that some 'business event' has occurred. Notifications are not expected to be replied to.

**XML Instance Representation**

```
<...
|
```

```
fpmVersion=" xsd:token (value comes from list: {'5-0'}) [1]
'Indicate which version of the FpML Schema an FpML message adheres to.'
```

"

```
expectedBuild=" xsd:positiveInteger [0..1]
'This optional attribute can be supplied by a message creator in an FpML instance to
specify which build number of the schema was used to define the message when it was generated.'
```

"

```
actualBuild="8 [0..1]
'The specific build number of this schema version. This attribute is not included in
an instance document. Instead, it is supplied by the XML parser when the document is
validated against the FpML schema and indicates the build number of the schema file. Every
time FpML publishes a change to the schema, validation rules, or examples within a version
(e.g., version 4.2) the actual build number is incremented. If no changes have been
made between releases within a version (i.e. from Trial Recommendation to Recommendation)
the actual build number stays the same.'
```

">

```
<header> NotificationMessageHeader </header> [1]
<validation> Validation </validation> [0..*]
```

Start Group: Correlation.model [0..1]

```
<correlationId> CorrelationID </correlationId> [1]
'A qualified identifier used to correlate between messages'
```

End Group: Correlation.model

Start Group: Sequence.model [0..1]

```
<sequenceNumber> xsd:positiveInteger </sequenceNumber> [1]
'A numeric value that can be used to order messages with the same correlation identifier
from the same sender.'
```

End Group: Sequence.model

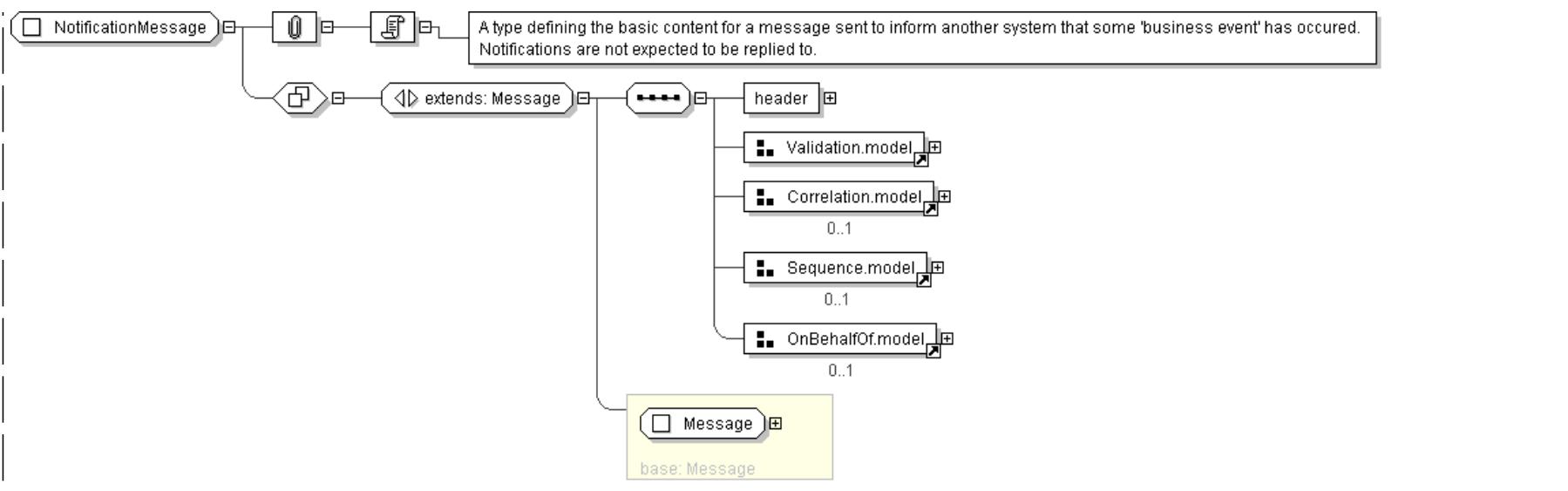
Start Group: OnBehalfOf.model [0..1]

```
<onBehalfOf> OnBehalfOf </onBehalfOf> [1]
'Indicates which party (and accounts) a trade is being processed for.'
```

End Group: OnBehalfOf.model

</...>

**Diagram**



### Schema Component Representation

```

<xsd:complexType name="NotificationMessage" abstract="true">
  <xsd:complexContent>
    <xsd:extension base=" Message ">
      <xsd:sequence>
        <xsd:element name="header" type=" NotificationMessageHeader " />
        <xsd:group ref=" Validation.model " />
        <xsd:group ref=" Correlation.model " minOccurs="0" />
        <xsd:group ref=" Sequence.model " minOccurs="0" />
        <xsd:group ref=" OnBehalfOf.model " minOccurs="0" />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

[top](#)

### Complex Type: [NotificationMessageHeader](#)

Super-types:	<a href="#">MessageHeader</a> < <b>NotificationMessageHeader</b> (by extension)
Sub-types:	None

Name	NotificationMessageHeader
------	---------------------------

Used by (from the same schema document)	Complex Type <a href="#">NotificationMessage</a>
---	--

Abstract	no
----------	----

Documentation	A type that refines the generic message header to match the requirements of a NotificationMessage.
---------------	--

### XML Instance Representation

```

<...>
<messageId> MessageId </messageId> [1]
'A unique identifier (within its coding scheme) assigned to the message by its creating party.'

<inReplyTo> MessageId </inReplyTo> [0..1]

```

'A copy of the unique message identifier (within its own coding scheme) to which this message is responding.'

```
<sentBy> MessageAddress </sentBy> [1]
```

'The unique identifier (within its coding scheme) for the originator of a message instance.'

```
<sendTo> MessageAddress </sendTo> [0..*]
```

'A unique identifier (within its coding scheme) indicating an intended recipient of a message.'

```
<copyTo> MessageAddress </copyTo> [0..*]
```

'A unique identifier (within the specified coding scheme) giving the details of some party to whom a copy of this message will be sent for reference.'

```
<creationTimestamp> xsd:dateTime </creationTimestamp> [1]
```

'The date and time (on the source system) when this message instance was created.'

```
<expiryTimestamp> xsd:dateTime </expiryTimestamp> [0..1]
```

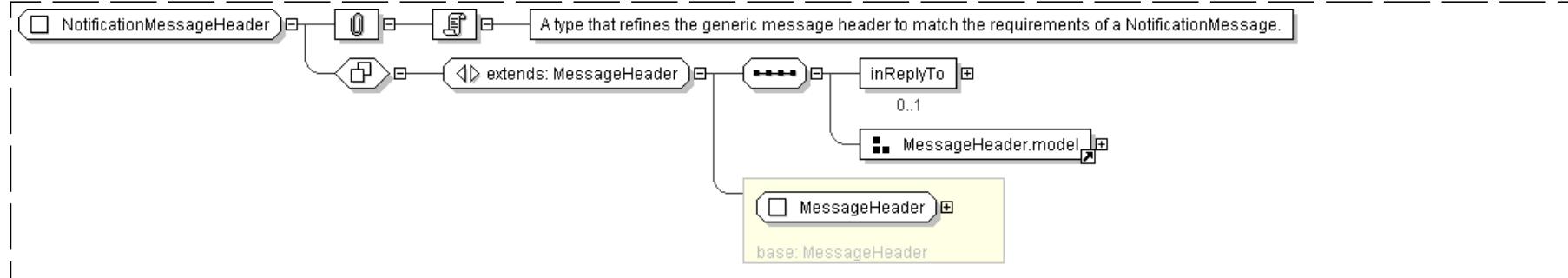
'The date and time (on the source system) when this message instance will be considered expired.'

```
<partyMessageInformation> PartyMessageInformation </partyMessageInformation> [0..*]
```

'Additional message information that may be provided by each involved party.'

```
<dsig:Signature> ... </dsig:Signature> [0..*]
```

```
</...>
```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="NotificationMessageHeader">
  <xsd:complexContent>
    <xsd:extension base=" MessageHeader ">
      <xsd:sequence>
        <xsd:element name="inReplyTo" type=" MessageId " minOccurs="0"/>
        <xsd:group ref=" MessageHeader.model "/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
  
```

<b>Super-types:</b>	None
<b>Sub-types:</b>	None

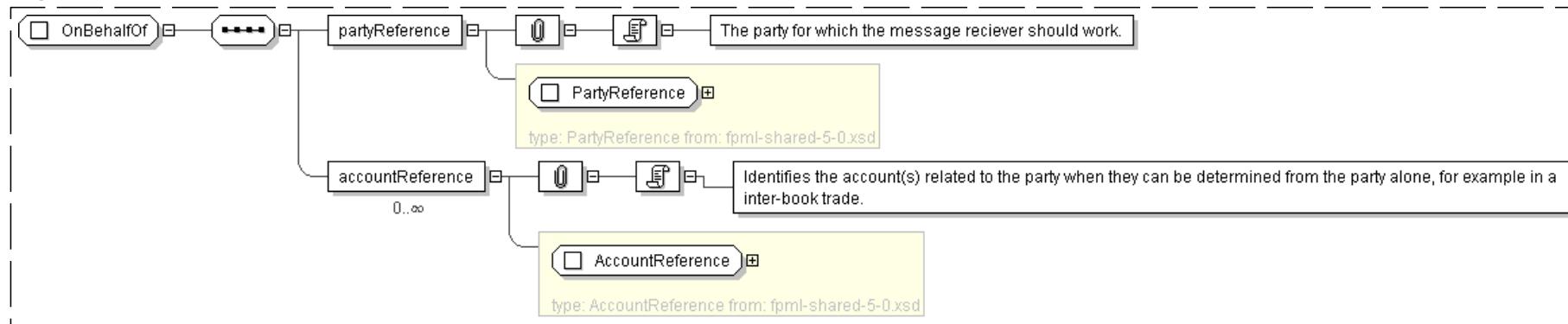
<b>Name</b>	OnBehalfOf
<b>Used by (from the same schema document)</b>	Model Group <a href="#">OnBehalfOf.model</a>
<b>Abstract</b>	no

**XML Instance Representation**

```
<...>
<partyReference> PartyReference </partyReference> [1]
'The party for which the message receiver should work.'

<accountReference> AccountReference </accountReference> [0..*]
'Identifies the account(s) related to the party when they can be determined from the
party alone, for example in a inter-book trade.'

</...>
```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="OnBehalfOf">
  <xsd:sequence>
    <xsd:element name="partyReference" type="PartyReference" />
    <xsd:element name="accountReference" type="AccountReference" %
      minOccurs="0" maxOccurs="unbounded" />
  </xsd:sequence>
</xsd:complexType>
  
```

[top](#)**Complex Type: [PartyMessageInformation](#)**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None

<b>Name</b>	PartyMessageInformation
<b>Used by (from the same schema document)</b>	Model Group <a href="#">MessageHeader.model</a>

**Abstract**

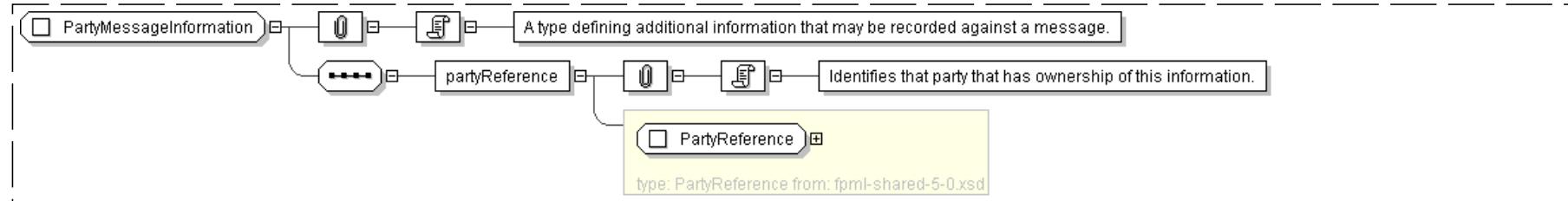
no

**Documentation**

A type defining additional information that may be recorded against a message.

**XML Instance Representation**

```
<...>
  <partyReference> PartyReference </partyReference> [1]
    'Identifies that party that has ownership of this information.'
<...>
```

**Diagram****Schema Component Representation**

```
<xsd:complexType name="PartyMessageInformation">
  <xsd:sequence>
    <xsd:element name="partyReference" type="PartyReference" />
  </xsd:sequence>
</xsd:complexType>
```

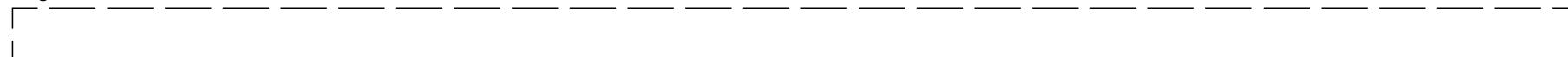
[top](#)**Complex Type: ProblemLocation**

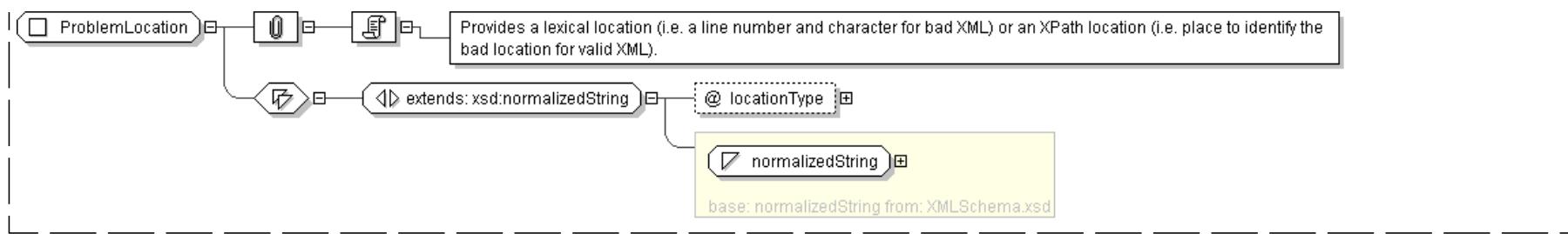
<b>Super-types:</b>	xsd:normalizedString < <b>ProblemLocation</b> (by extension)
<b>Sub-types:</b>	None

<b>Name</b>	ProblemLocation
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">Reason</a>
<b>Abstract</b>	no
<b>Documentation</b>	Provides a lexical location (i.e. a line number and character for bad XML) or an XPath location (i.e. place to identify the bad location for valid XML).

**XML Instance Representation**

```
<...
  locationType=" xsd:token [0..1]
    'The value of the locationType attribute defines which type of location has been given. It
    may take the values \'lexical\' or \'xpath\'.'>
  xsd:normalizedString
<...>
```

**Diagram**



## Schema Component Representation

```
<xsd:complexType name="ProblemLocation">
    <xsd:simpleContent>
        <xsd:extension base=" xsd:normalizedString ">
            <xsd:attribute name="locationType" type=" xsd:token " />
        </xsd:extension>
    </xsd:simpleContent>
</xsd:complexType>
```

top

## Complex Type: Reason

<i>Super-types:</i>	None
<i>Sub-types:</i>	None
<b>Name</b>	Reason
<b>Used by (from the same schema document)</b>	Model Group <a href="#">Exception.model</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type defining a content model for describing the nature and possible location of a error within a previous message.

## XML Instance Representation

```
<...>
<reasonCode> ReasonCode </reasonCode> [1]
'A machine interpretable error code.'

<location> ProblemLocation </location> [0..1]
'A value indicating the location of the problem within the subject message.'

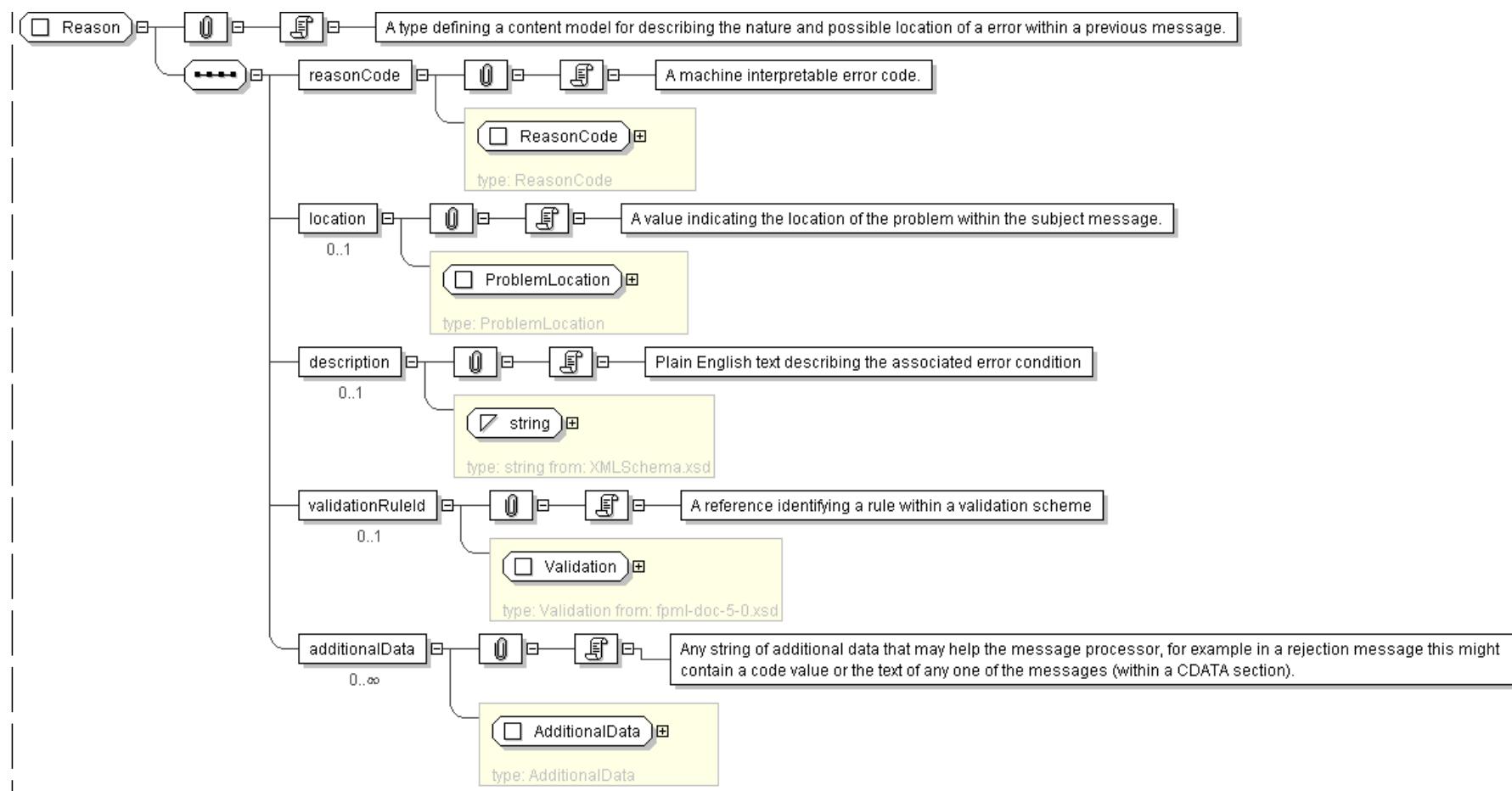
<description> xsd:string </description> [0..1]
'Plain English text describing the associated error condition'

<validationRuleId> Validation </validationRuleId> [0..1]
'A reference identifying a rule within a validation scheme'

<additionalData> AdditionalData </additionalData> [0..*]
'Any string of additional data that may help the message processor, for example in a rejection message this might contain a code value or the text of any one of the messages (within a CDATA section).'

</...>
```

## Diagram



#### Schema Component Representation

```

<xsd:complexType name="Reason">
  <xsd:sequence>
    <xsd:element name="reasonCode" type=" ReasonCode " />
    <xsd:element name="location" type=" ProblemLocation " minOccurs="0"/>
    <xsd:element name="description" type=" xsd:string " minOccurs="0"/>
    <xsd:element name="validationRuleId" type=" Validation " minOccurs="0"/>
    <xsd:element name="additionalData" type=" AdditionalData " minOccurs="0" maxOccurs="unbounded" />
  </xsd:sequence>
</xsd:complexType>

```

[top](#)

#### Complex Type: **ReasonCode**

Super-types:	<a href="#">Scheme</a> < <b>ReasonCode</b> (by extension)
Sub-types:	None
Name	ReasonCode

**Used by (from the same schema document)**Complex Type [Reason](#)**Abstract**

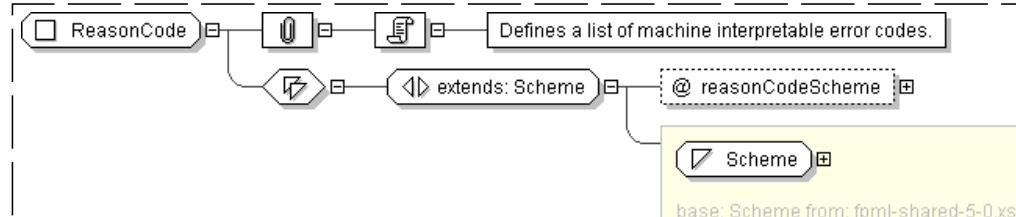
no

**Documentation**

Defines a list of machine interpretable error codes.

**XML Instance Representation**

```
<...>
<reasonCodeScheme=" xsd:anyURI [0..1]">
  Scheme
</...>
```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="ReasonCode">
  <xsd:simpleContent>
    <xsd:extension base=" Scheme ">
      <xsd:attribute name="reasonCodeScheme" type=" xsd:anyURI " default="http://www.fpml.org/
        coding-scheme/reason-code" />
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
```

[top](#)**Complex Type: RequestEventStatus****Super-types:**[Document](#) < [Message](#) (by extension) < [RequestMessage](#) (by extension) < [NonCorrectableRequestMessage](#) (by extension) < **RequestEventStatus**  
(by extension)**Sub-types:**

None

**Name**

RequestEventStatus

**Used by (from the same schema document)**Element [requestEventStatus](#)**Abstract**

no

**Documentation**

A type defining the content model for a message allowing one party to query the status of one event (trade or post-trade event) previously sent to another party.

**XML Instance Representation**

```
<...>
  fpmlVersion=" xsd:token (value comes from list: {'5-0'}) [1]
  'Indicate which version of the FpML Schema an FpML message adheres to.'
  "
  expectedBuild=" xsd:positiveInteger [0..1]
  'This optional attribute can be supplied by a message creator in an FpML instance to
  specify which build number of the schema was used to define the message when it was generated.'
```

```

"
actualBuild="8 [0..1]
'The specific build number of this schema version. This attribute is not included in
an instance document. Instead, it is supplied by the XML parser when the document is
validated against the FpML schema and indicates the build number of the schema file. Every
time FpML publishes a change to the schema, validation rules, or examples within a version
(e.g., version 4.2) the actual build number is incremented. If no changes have been
made between releases within a version (i.e. from Trial Recommendation to Recommendation)
the actual build number stays the same.'

">
<header> RequestMessageHeader </header> [1]
<validation> Validation </validation> [0..*]
<correlationId> CorrelationId </correlationId> [1]
'A qualified identifier used to correlate between messages'

<sequenceNumber> xsd:positiveInteger </sequenceNumber> [1]
'A numeric value that can be used to order messages with the same correlation identifier
from the same sender.'

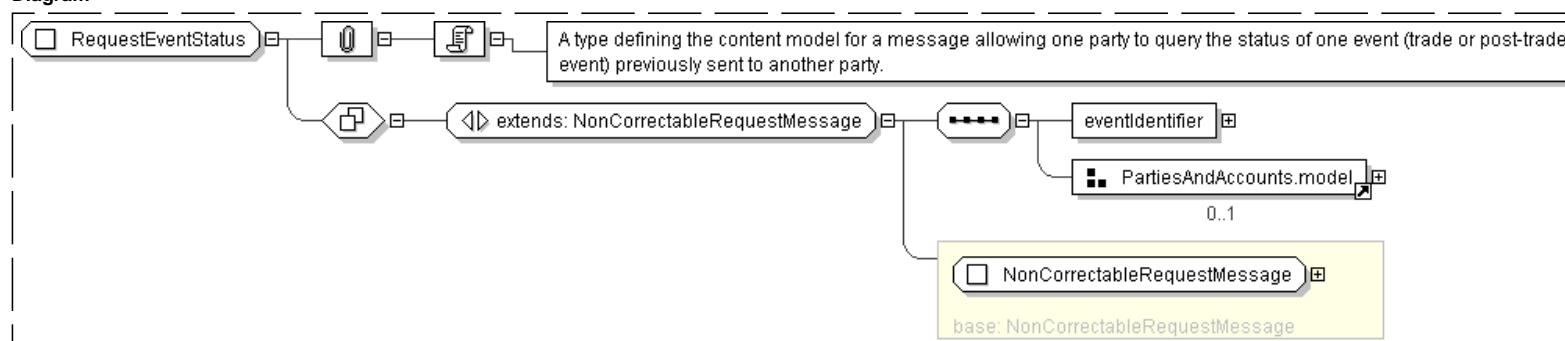
Start Group: OnBehalfOf.model [0..1]
<onBehalfOf> OnBehalfOf </onBehalfOf> [1]
'Indicates which party (and accounts) a trade is being processed for.'

End Group: OnBehalfOf.model
<eventIdentifier> EventIdentifier </eventIdentifier> [1]
Start Group: PartiesAndAccounts.model [0..1]
<party> Party </party> [1..*]
'A legal entity or a subdivision of a legal entity.', 'Parties can perform multiple roles in
a trade lifecycle. For example, the principal parties obligated to make payments from time
to time during the term of the trade, but may include other parties involved in, or
incidental to, the trade, such as parties acting in the role of novation transferor/
transferee, broker, calculation agent, etc. In FpML roles are defined in multiple places
within a document.'

<account> Account </account> [0..*]
'Optional account information used to precisely define the origination and destination
of financial instruments.'

End Group: PartiesAndAccounts.model
</...>

```

**Diagram**

**Schema Component Representation**

```

<xsd:complexType name="RequestEventStatus">
  <xsd:complexContent>
    <xsd:extension base=" NonCorrectableRequestMessage " >
      <xsd:sequence>
        <xsd:element name="eventIdentifier" type=" EventIdentifier " />
        <xsd:group ref=" PartiesAndAccounts.model " minOccurs="0" />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

[top](#)**Complex Type: RequestMessage**

<b>Super-types:</b>	<a href="#">Document</a> < <a href="#">Message</a> (by extension) < <b>RequestMessage</b> (by extension)
<b>Sub-types:</b>	<ul style="list-style-type: none"> <li>• <a href="#">CorrectableRequestMessage</a> (by extension)</li> <li>• <a href="#">NonCorrectableRequestMessage</a> (by extension)           <ul style="list-style-type: none"> <li>◦ <a href="#">RequestEventStatus</a> (by extension)</li> </ul> </li> </ul>
<b>Name</b>	RequestMessage

<b>Abstract</b>	yes
-----------------	-----

<b>Documentation</b>	A type defining the basic content of a message that requests the receiver to perform some business operation determined by the message type and its content.
----------------------	--

**XML Instance Representation**

```

<...
fpmVersion=" xsd:token (value comes from list: {'5-0'}) [1]
'Indicate which version of the FpML Schema an FpML message adheres to.'

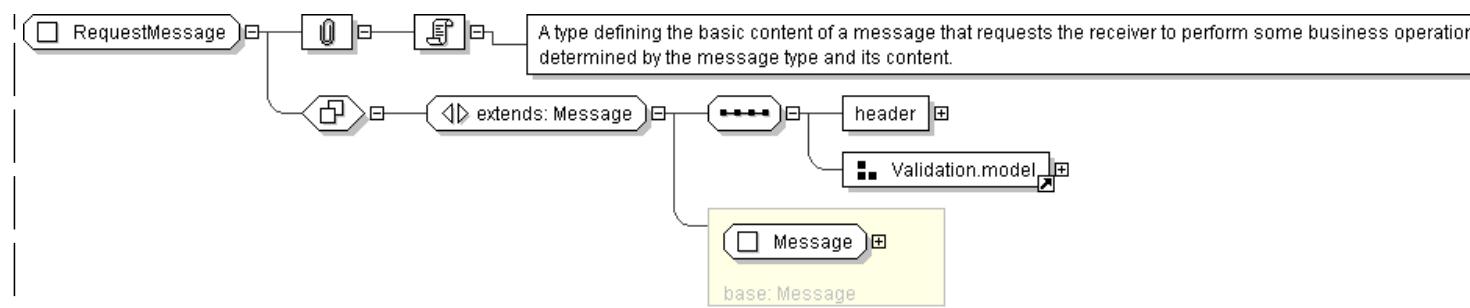
"
expectedBuild=" xsd:positiveInteger [0..1]
'This optional attribute can be supplied by a message creator in an FpML instance to
specify which build number of the schema was used to define the message when it was generated.'

"
actualBuild="8 [0..1]
'The specific build number of this schema version. This attribute is not included in
an instance document. Instead, it is supplied by the XML parser when the document is
validated against the FpML schema and indicates the build number of the schema file. Every
time FpML publishes a change to the schema, validation rules, or examples within a version
(e.g., version 4.2) the actual build number is incremented. If no changes have been
made between releases within a version (i.e. from Trial Recommendation to Recommendation)
the actual build number stays the same.'

">
  <header> RequestMessageHeader </header> [1]
  <validation> Validation </validation> [0..*]
</...>

```

**Diagram**

**Schema Component Representation**

```

<xsd:complexType name="RequestMessage" abstract="true">
  <xsd:complexContent>
    <xsd:extension base=" Message ">
      <xsd:sequence>
        <xsd:element name="header" type=" RequestMessageHeader " />
        <xsd:group ref=" Validation.model " />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
  
```

[top](#)**Complex Type: RequestMessageHeader**

<b>Super-types:</b>	<a href="#">MessageHeader</a> < RequestMessageHeader (by extension)
<b>Sub-types:</b>	None

<b>Name</b>	RequestMessageHeader
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">RequestMessage</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type refining the generic message header content to make it specific to request messages.

**XML Instance Representation**

```

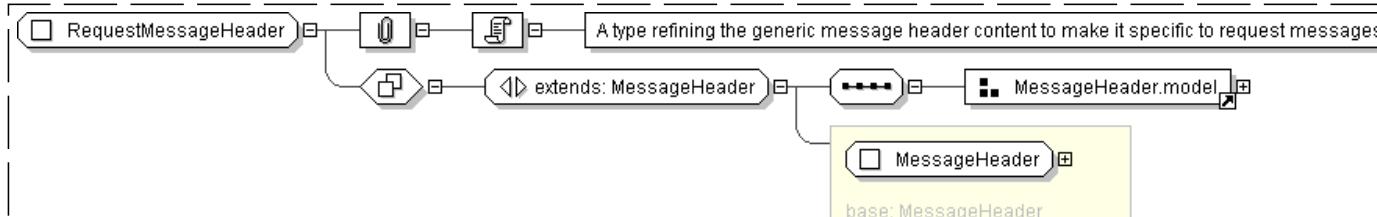
<...>
<messageId> MessageId </messageId> [1]
  'A unique identifier (within its coding scheme) assigned to the message by its creating party.'
<sentBy> MessageAddress </sentBy> [1]
  'The unique identifier (within its coding scheme) for the originator of a message instance.'
<sendTo> MessageAddress </sendTo> [0..*]
  'A unique identifier (within its coding scheme) indicating an intended recipient of a message.'
<copyTo> MessageAddress </copyTo> [0..*]
  'A unique identifier (within the specified coding scheme) giving the details of some party to whom a copy of this message will be sent for reference.'
<creationTimestamp> xsd:dateTime </creationTimestamp> [1]
  'The date and time (on the source system) when this message instance was created.'
  
```

```
<expiryTimestamp> xsd:dateTime </expiryTimestamp> [0..1]
'The date and time (on the source system) when this message instance will be
considered expired.'
```

```
<partyMessageInformation> PartyMessageInformation </partyMessageInformation> [0..*]
'Additional message information that may be provided by each involved party.'
```

```
<dsig:Signature> ... </dsig:Signature> [0..*]
```

&lt;/...&gt;

**Diagram****Schema Component Representation**

```
<xsd:complexType name="RequestMessageHeader">
  <xsd:complexContent>
    <xsd:extension base=" MessageHeader ">
      <xsd:sequence>
        <xsd:group ref=" MessageHeader.model "/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

[top](#)**Complex Type: ResponseMessage****Super-types:**[Document](#) < [Message](#) (by extension) < **ResponseMessage** (by extension)**Sub-types:**

- [Acknowledgement](#) (by extension)
- [EventStatusResponse](#) (by extension)

**Name**

ResponseMessage

**Abstract**

yes

**Documentation**

A type refining the generic message content model to make it specific to response messages.

**XML Instance Representation**

```
<...
fpmVersion=" xsd:token (value comes from list: {'5-0'}) [1]
'Indicate which version of the FpML Schema an FpML message adheres to.'

"
expectedBuild=" xsd:positiveInteger [0..1]
'This optional attribute can be supplied by a message creator in an FpML instance to
specify which build number of the schema was used to define the message when it was generated.'
```

"  
**actualBuild="8** [0..1]  
*'The specific build number of this schema version. This attribute is not included in an instance document. Instead, it is supplied by the XML parser when the document is validated against the FpML schema and indicates the build number of the schema file. Every time FpML publishes a change to the schema, validation rules, or examples within a version (e.g., version 4.2) the actual build number is incremented. If no changes have been made between releases within a version (i.e. from Trial Recommendation to Recommendation) the actual build number stays the same.'*

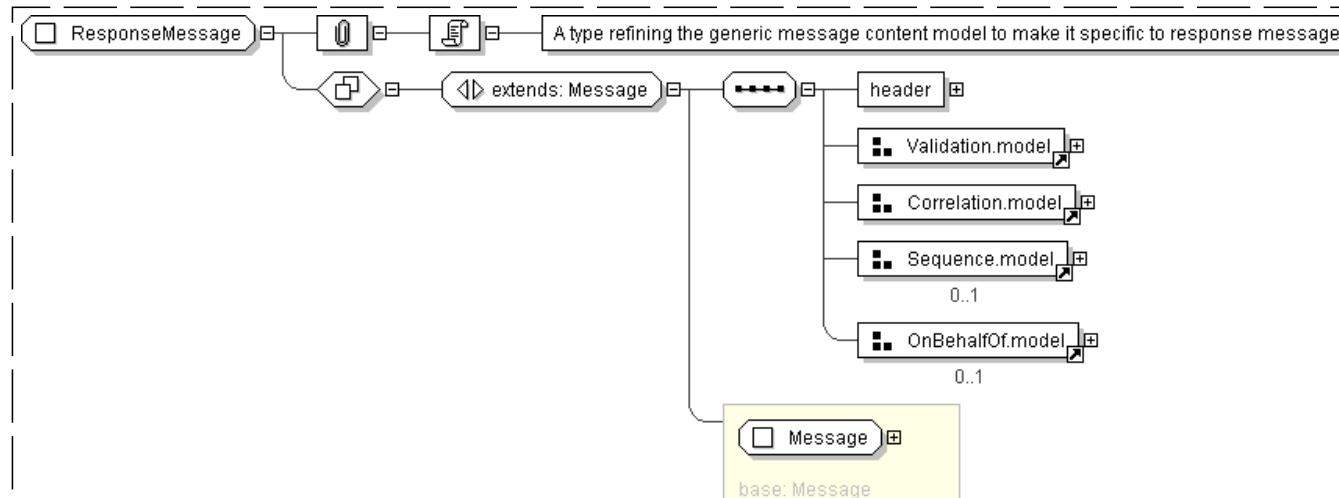
">  
**<header>** ResponseMessageHeader **</header>** [1]  
**<validation>** Validation **</validation>** [0..\*]  
**<correlationId>** CorrelationId **</correlationId>** [1]  
*'A qualified identifier used to correlate between messages'*

Start Group: Sequence.model [0..1]  
**<sequenceNumber>** xsd:positiveInteger **</sequenceNumber>** [1]  
*'A numeric value that can be used to order messages with the same correlation identifier from the same sender.'*

End Group: Sequence.model  
Start Group: OnBehalfOf.model [0..1]  
**<onBehalfOf>** OnBehalfOf **</onBehalfOf>** [1]  
*'Indicates which party (and accounts) a trade is being processed for.'*

End Group: OnBehalfOf.model  
</...>

#### Diagram



#### Schema Component Representation

```

<xsd:complexType name="ResponseMessage" abstract="true">
  <xsd:complexContent>
    <xsd:extension base=" Message ">
      <xsd:sequence>
        <xsd:element name="header" type=" ResponseMessageHeader " />
        <xsd:group ref=" Validation.model " />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
  
```

```

<xsd:group ref=" Correlation.model " />
<xsd:group ref=" Sequence.model " minOccurs="0" />
<xsd:group ref=" OnBehalfOf.model " minOccurs="0" />
</xsd:sequence>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>

```

[top](#)

## Complex Type: ResponseMessageHeader

**Super-types:** [MessageHeader](#) < **ResponseMessageHeader** (by extension)

**Sub-types:** None

**Name** ResponseMessageHeader

**Used by (from the same schema document)** Complex Type [ResponseMessage](#)

**Abstract** no

**Documentation** A type refining the generic message header to make it specific to response messages.

### XML Instance Representation

```

<...>
<messageId> MessageId </messageId> [1]
'A unique identifier (within its coding scheme) assigned to the message by its creating party.'

<inReplyTo> MessageId </inReplyTo> [1]
'A copy of the unique message identifier (within its own coding scheme) to which this message is responding.'

<sentBy> MessageAddress </sentBy> [1]
'The unique identifier (within its coding scheme) for the originator of a message instance.'

<sendTo> MessageAddress </sendTo> [0..*]
'A unique identifier (within its coding scheme) indicating an intended recipient of a message.'

<copyTo> MessageAddress </copyTo> [0..*]
'A unique identifier (within the specified coding scheme) giving the details of some party to whom a copy of this message will be sent for reference.'

<creationTimestamp> xsd:dateTime </creationTimestamp> [1]
'The date and time (on the source system) when this message instance was created.'

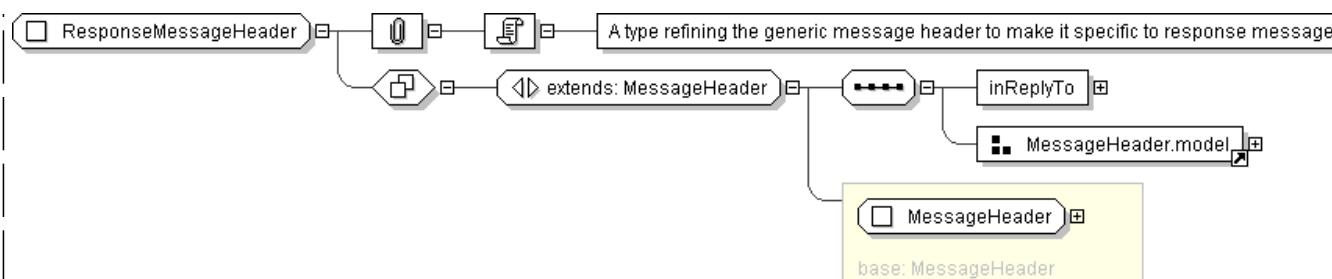
<expiryTimestamp> xsd:dateTime </expiryTimestamp> [0..1]
'The date and time (on the source system) when this message instance will be considered expired.'

<partyMessageInformation> PartyMessageInformation </partyMessageInformation> [0..*]
'Additional message information that may be provided by each involved party.'

<dsig:Signature> ... </dsig:Signature> [0..*]
</...>

```

### Diagram

**Schema Component Representation**

```

<xsd:complexType name="ResponseMessageHeader">
  <xsd:complexContent>
    <xsd:extension base=" MessageHeader ">
      <xsd:sequence>
        <xsd:element name="inReplyTo" type=" MessageId ">
        <xsd:group ref=" MessageHeader.model ">
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
  
```

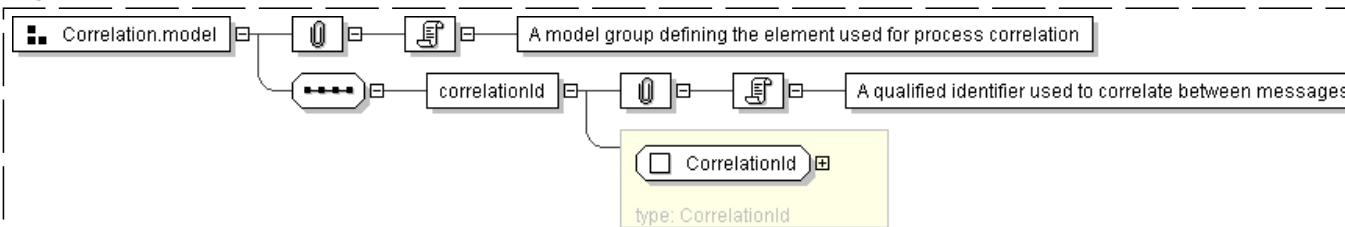
[top](#)**Model Group: Correlation.model**

Name	Correlation.model
Used by (from the same schema document)	Complex Type <a href="#">CorrectableRequestMessage</a> , Complex Type <a href="#">EventIdentifier</a> , Complex Type <a href="#">Exception</a> , Complex Type <a href="#">NonCorrectableRequestMessage</a> , Complex Type <a href="#">NotificationMessage</a> , Complex Type <a href="#">ResponseMessage</a>
Documentation	A model group defining the element used for process correlation

**XML Instance Representation**

```

<correlationId> CorrelationId </correlationId> [1]
' A qualified identifier used to correlate between messages'
  
```

**Diagram****Schema Component Representation**

```

<xsd:group name="CorrelationId">
  <xsd:sequence>
    <xsd:element name="correlationId" type=" CorrelationId ">
  </xsd:sequence>
</xsd:group>
  
```

## Model Group: Exception.model

<b>Name</b>	Exception.model
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">Exception</a>
<b>Documentation</b>	A model group which has exception elements.

### XML Instance Representation

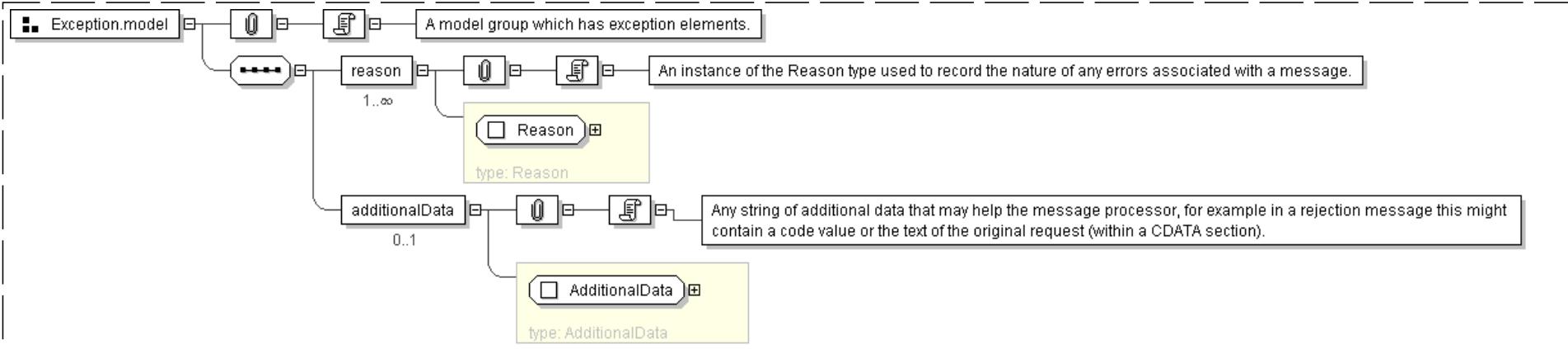
```
<reason> Reason </reason> [1..*]
```

'An instance of the Reason type used to record the nature of any errors associated with a message.'

```
<additionalData> AdditionalData </additionalData> [0..1]
```

'Any string of additional data that may help the message processor, for example in a rejection message this might contain a code value or the text of the original request (within a CDATA section).'

### Diagram



### Schema Component Representation

```

<xsd:group name="Exception.model">
  <xsd:sequence>
    <xsd:element name="reason" type="Reason" maxOccurs="unbounded" />
    <xsd:element name="additionalData" type="AdditionalData" minOccurs="0" />
  </xsd:sequence>
</xsd:group>
  
```

## Model Group: MessageHeader.model

<b>Name</b>	MessageHeader.model
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">ExceptionMessageHeader</a> , Complex Type <a href="#">NotificationMessageHeader</a> , Complex Type <a href="#">RequestMessageHeader</a> , Complex Type <a href="#">ResponseMessageHeader</a>

**Documentation**

Defines the structure that contains routing and identification information, which allows processing and transfer of the message. It contains only messaging information that is applicable to all messages. If the information is not message related or is not applicable to all messages then it is not defined in the message header.

**XML Instance Representation**

```
<sentBy> MessageAddress </sentBy> [1]
```

'The unique identifier (within its coding scheme) for the originator of a message instance.'

```
<sendTo> MessageAddress </sendTo> [0..*]
```

'A unique identifier (within its coding scheme) indicating an intended recipient of a message.'

```
<copyTo> MessageAddress </copyTo> [0..*]
```

'A unique identifier (within the specified coding scheme) giving the details of some party to whom a copy of this message will be sent for reference.'

```
<creationTimestamp> xsd:dateTime </creationTimestamp> [1]
```

'The date and time (on the source system) when this message instance was created.'

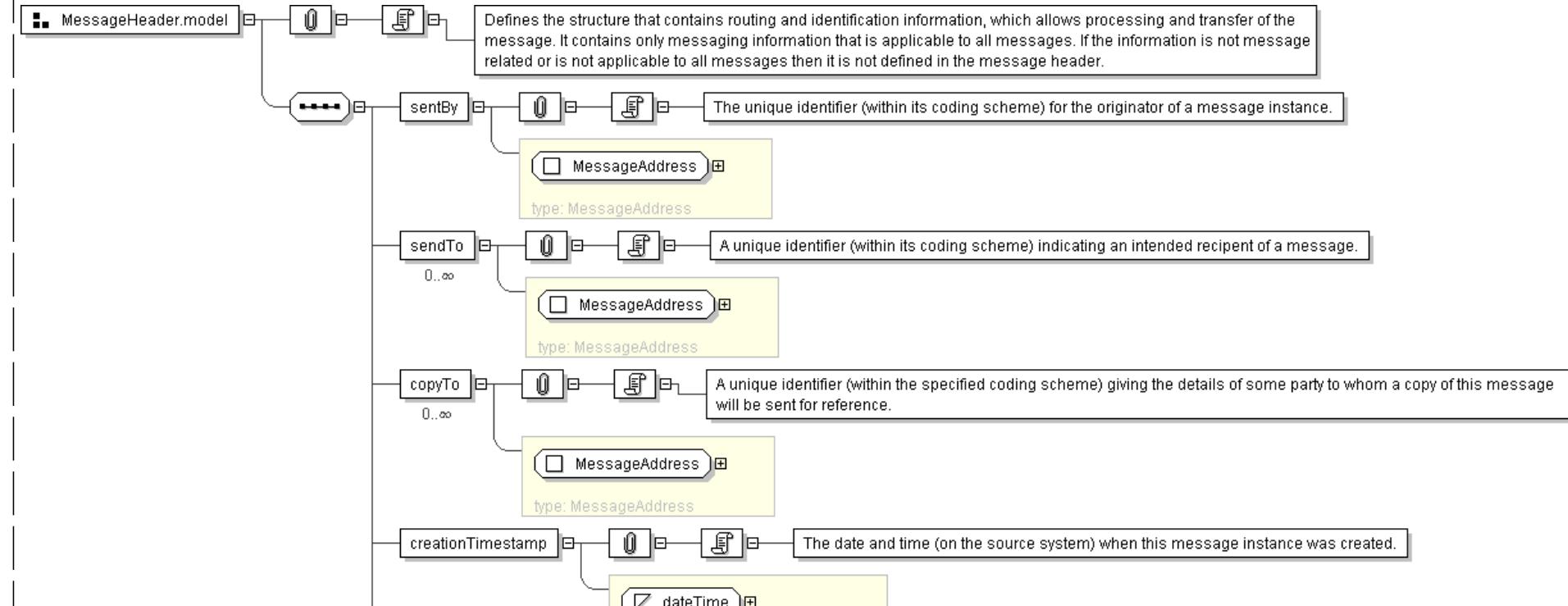
```
<expiryTimestamp> xsd:dateTime </expiryTimestamp> [0..1]
```

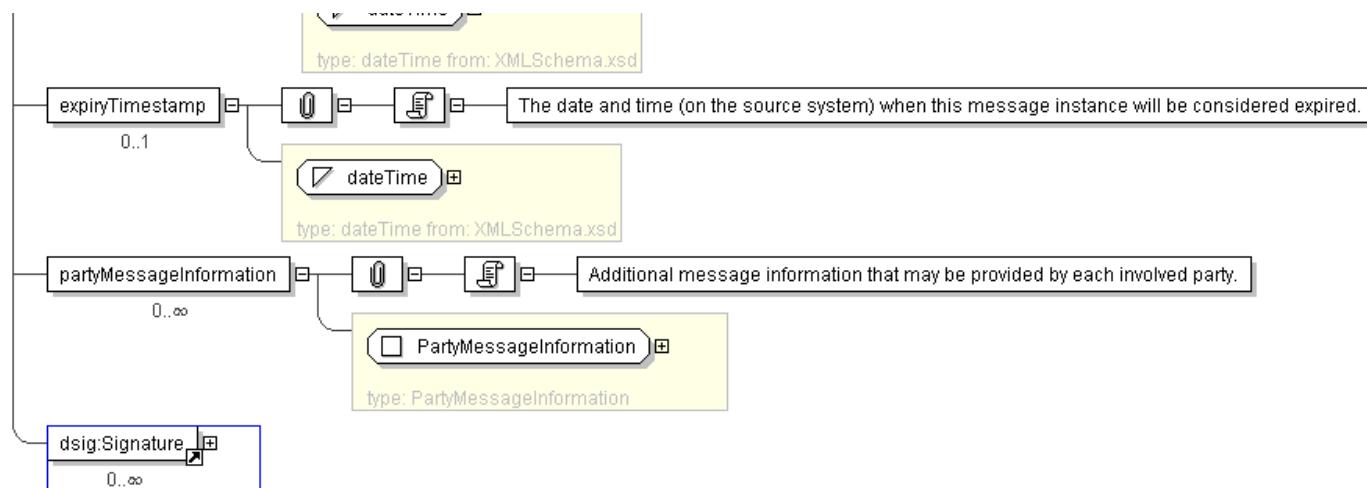
'The date and time (on the source system) when this message instance will be considered expired.'

```
<partyMessageInformation> PartyMessageInformation </partyMessageInformation> [0..*]
```

'Additional message information that may be provided by each involved party.'

```
<dsig:Signature> ... </dsig:Signature> [0..*]
```

**Diagram**

**Schema Component Representation**

```

<xsd:group name="MessageHeader.model">
  <xsd:sequence>
    <xsd:element name="sentBy" type=" MessageAddress " />
    <xsd:element name="sendTo" type=" MessageAddress " minOccurs="0" maxOccurs="unbounded" />
    <xsd:element name="copyTo" type=" MessageAddress " minOccurs="0" maxOccurs="unbounded" />
    <xsd:element name="creationTimestamp" type=" xsd:dateTime " />
    <xsd:element name="expiryTimestamp" type=" xsd:dateTime " minOccurs="0" />
    <xsd:element name="partyMessageInformation" type=" PartyMessageInformation " minOccurs="0" maxOccurs="unbounded" />
    <xsd:element ref=" dsig:Signature " minOccurs="0" maxOccurs="unbounded" />
  </xsd:sequence>
</xsd:group>

```

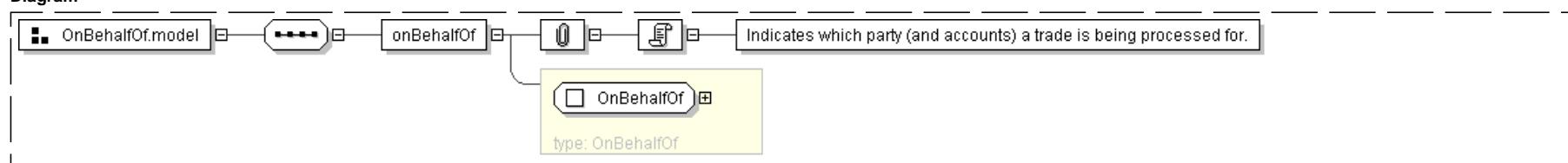
[top](#)**Model Group: OnBehalfOf.model**

<b>Name</b>	OnBehalfOf.model
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">CorrectableRequestMessage</a> , Complex Type <a href="#">NonCorrectableRequestMessage</a> , Complex Type <a href="#">NotificationMessage</a> , Complex Type <a href="#">ResponseMessage</a>

**XML Instance Representation**

```
<onBehalfOf> OnBehalfOf </onBehalfOf> [1]
```

'Indicates which party (and accounts) a trade is being processed for.'

**Diagram**

**Schema Component Representation**

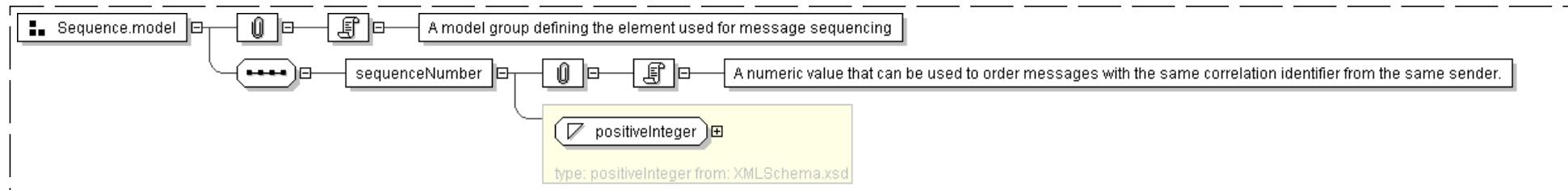
```
<xsd:group name="OnBehalfOf.model">
  <xsd:sequence>
    <xsd:element name="onBehalfOf" type=" OnBehalfOf "/>
  </xsd:sequence>
</xsd:group>
```

[top](#)**Model Group: Sequence.model**

<b>Name</b>	Sequence.model
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">CorrectableRequestMessage</a> , Complex Type <a href="#">EventIdentifier</a> , Complex Type <a href="#">Exception</a> , Complex Type <a href="#">NonCorrectableRequestMessage</a> , Complex Type <a href="#">NotificationMessage</a> , Complex Type <a href="#">ResponseMessage</a>
<b>Documentation</b>	A model group defining the element used for message sequencing

**XML Instance Representation**

```
<sequenceNumber> xsd:positiveInteger </sequenceNumber> [1]
'A numeric value that can be used to order messages with the same correlation identifier
from the same sender.'
```

**Diagram****Schema Component Representation**

```
<xsd:group name="Sequence.model">
  <xsd:sequence>
    <xsd:element name="sequenceNumber" type=" xsd:positiveInteger "/>
  </xsd:sequence>
</xsd:group>
```

[top](#)**Legend**

**Complex Type:**  
Schema Component Type

**AusAddress**  
Schema Component Name

**Super-types:** [Address](#) < AusAddress (by extension)  
**Sub-types:** • [QLDAddress](#) (by restriction)

If this schema component is a type definition, its type hierarchy is shown in a gray-bordered box.

Name	AusAddress
Abstract	no

The table above displays the properties of this schema component.

#### XML Instance Representation

```
<... country="Australia" >
<unitNo> string </unitNo> [0..1]
<houseNo> string </houseNo> [1]
<street> string </street> [1]
Start Choice [1]
<city> string </city> [1]
<town> string </town> [1]
End Choice
<state> AusStates </state> [1]
<postcode> string <>pattern = [1-9][0-9]{3}</> </postcode> [1]
</...>
```

The XML Instance Representation table above shows the schema component's content as an XML instance.

- The minimum and maximum occurrence of elements and attributes are provided in square brackets, e.g. [0..1].
- Model group information are shown in gray, e.g. Start Choice ... End Choice.
- For type derivations, the elements and attributes that have been added to or changed from the base type's content are shown in **bold**.
- If an element/attribute has a fixed value, the fixed value is shown in green, e.g. country="Australia".
- Otherwise, the type of the element/attribute is displayed.
  - If the element/attribute's type is in the schema, a link is provided to it.
  - For local simple type definitions, the constraints are displayed in angle brackets, e.g. <>pattern = [1-9][0-9]{3}</>.

#### Schema Component Representation

```
<complexType name="AusAddress">
<complexContent>
<extension base=" Address " >
<sequence>
<element name="state" type=" AusStates " />
<element name="postcode">
<simpleType>
<restriction base=" string ">
<pattern value="[1-9][0-9]{3}" />
</restriction>
</simpleType>
</element>
</sequence>
<attribute name="country" type=" string " fixed="Australia" />
</extension>
</complexContent>
</complexType>
```

The Schema Component Representation table above displays the underlying XML representation of the schema component. (Annotations are not shown.)

[top](#)

## Glossary

**Abstract** (Applies to complex type definitions and element declarations). An abstract element or complex type cannot be used to validate an element instance. If there is a reference to an abstract element, only element declarations that can substitute the abstract element can be used to validate the instance. For references to abstract type definitions, only derived types can be used.

**All Model Group** Child elements can be provided *in any order* in instances. See: <http://www.w3.org/TR/xmlschema-1/#element-all>.

**Choice Model Group** Only one from the list of child elements and model groups can be provided in instances. See: <http://www.w3.org/TR/xmlschema-1/#element-choice>.

**Collapse Whitespace Policy** Replace tab, line feed, and carriage return characters with space character (Unicode character 32). Then, collapse contiguous sequences of space characters into single space character, and remove leading and trailing space characters.

**Disallowed Substitutions** (Applies to element declarations). If *substitution* is specified, then *substitution group* members cannot be used in place of the given element declaration to validate element instances. If *derivation methods*, e.g. extension, restriction, are specified, then the given element declaration will not validate element instances that have types derived from the element declaration's type using the specified derivation methods. Normally, element instances can override their declaration's type by specifying an *xsi:type* attribute.

**Key Constraint** Like *Uniqueness Constraint*, but additionally requires that the specified value(s) must be provided. See: [http://www.w3.org/TR/xmlschema-1/#clentity-constraint\\_Definitions](http://www.w3.org/TR/xmlschema-1/#clentity-constraint_Definitions).

**Key Reference Constraint** Ensures that the specified value(s) must match value(s) from a *Key Constraint* or *Uniqueness Constraint*. See: [http://www.w3.org/TR/xmlschema-1/#cldentity-constraint\\_Definitions](http://www.w3.org/TR/xmlschema-1/#cldentity-constraint_Definitions).

**Model Group** Groups together element content, specifying the order in which the element content can occur and the number of times the group of element content may be repeated. See: [http://www.w3.org/TR/xmlschema-1/#Model\\_Groups](http://www.w3.org/TR/xmlschema-1/#Model_Groups).

**Nillable** (Applies to element declarations). If an element declaration is nillable, instances can use the *xsi:nil* attribute. The *xsi:nil* attribute is the boolean attribute, *nil*, from the <http://www.w3.org/2001/XMLSchema-instance> namespace. If an element instance has an *xsi:nil* attribute set to true, it can be left empty, even though its element declaration may have required content.

**Notation** A notation is used to identify the format of a piece of data. Values of elements and attributes that are of type, NOTATION, must come from the names of declared notations. See: [http://www.w3.org/TR/xmlschema-1/#cNotation\\_Declarations](http://www.w3.org/TR/xmlschema-1/#cNotation_Declarations).

**Preserve Whitespace Policy** Preserve whitespaces exactly as they appear in instances.

**Prohibited Derivations** (Applies to type definitions). Derivation methods that cannot be used to create sub-types from a given type definition.

**Prohibited Substitutions** (Applies to complex type definitions). Prevents sub-types that have been derived using the specified derivation methods from validating element instances in place of the given type definition.

**Replace Whitespace Policy** Replace tab, line feed, and carriage return characters with space character (Unicode character 32).

**Sequence Model Group** Child elements and model groups must be provided *in the specified order* in instances. See: <http://www.w3.org/TR/xmlschema-1/#element-sequence>.

**Substitution Group** Elements that are *members* of a substitution group can be used wherever the *head* element of the substitution group is referenced.

**Substitution Group Exclusions** (Applies to element declarations). Prohibits element declarations from nominating themselves as being able to substitute a given element declaration, if they have types that are derived from the original element's type using the specified derivation methods.

**Target Namespace** The target namespace identifies the namespace that components in this schema belongs to. If no target namespace is provided, then the schema components do not belong to any namespace.

**Uniqueness Constraint** Ensures uniqueness of an element/attribute value, or a combination of values, within a specified scope. See: [http://www.w3.org/TR/xmlschema-1/#clentity-constraint\\_Definitions](http://www.w3.org/TR/xmlschema-1/#clentity-constraint_Definitions).

[top](#)

# XML Schema Documentation

## Table of Contents

- [Schema Document Properties](#)
- [Global Definitions](#)
  - [Complex Type: Asian](#)
  - [Complex Type: AveragingObservationList](#)
  - [Complex Type: AveragingPeriod](#)
  - [Complex Type: AveragingSchedule](#)
  - [Complex Type: Barrier](#)
  - [Complex Type: CalendarSpread](#)
  - [Complex Type: ClassifiedPayment](#)
  - [Complex Type: Composite](#)
  - [Complex Type: CreditEventNotice](#)
  - [Complex Type: CreditEvents](#)
  - [Complex Type: CreditEventsReference](#)
  - [Complex Type: FailureToPay](#)
  - [Complex Type: FeaturePayment](#)
  - [Complex Type: FrequencyType](#)
  - [Complex Type: FxFeature](#)
  - [Complex Type: GracePeriodExtension](#)
  - [Complex Type: Knock](#)
  - [Complex Type: MarketDisruption](#)
  - [Complex Type: NotifyingParty](#)
  - [Complex Type: OptionBase](#)
  - [Complex Type: OptionBaseExtended](#)
  - [Complex Type: OptionFeature](#)
  - [Complex Type: OptionNumericStrike](#)
  - [Complex Type: OptionStrike](#)
  - [Complex Type: PassThrough](#)
  - [Complex Type: PassThroughItem](#)
  - [Complex Type: Premium](#)
  - [Complex Type: PubliclyAvailableInformation](#)
  - [Complex Type: Quanto](#)
  - [Complex Type: Restructuring](#)
  - [Complex Type: RestructuringType](#)
  - [Complex Type: StrategyFeature](#)
  - [Complex Type: StrikeSpread](#)
  - [Complex Type: Trigger](#)
  - [Complex Type: TriggerEvent](#)
  - [Complex Type: WeightedAveragingObservation](#)
- [Model Group: OptionBaseFeature.model](#)
- [Model Group: OptionDenomination.model](#)
- [Model Group: OptionFeature.model](#)
- [Model Group: OptionSettlement.model](#)
- [Legend](#)
- [Glossary](#)

top

## Schema Document Properties

<b>Target Namespace</b>	<a href="http://www.fpml.org/FpML-5/confirmation">http://www.fpml.org/FpML-5/confirmation</a>
<b>Version</b>	\$Revision: 2590 \$
<b>Element and Attribute Namespaces</b>	<ul style="list-style-type: none"> <li>• Global element and attribute declarations belong to this schema's target namespace.</li> <li>• By default, local element declarations belong to this schema's target namespace.</li> <li>• By default, local attribute declarations have no namespace.</li> </ul>
<b>Schema Composition</b>	<ul style="list-style-type: none"> <li>• This schema includes components from the following schema document(s):           <ul style="list-style-type: none"> <li>◦ <a href="#">fpml-asset-5-0.xsd</a></li> </ul> </li> </ul>

## Declared Namespaces

Prefix	Namespace
Default namespace	<a href="http://www.fpml.org/FpML-5/confirmation">http://www.fpml.org/FpML-5/confirmation</a>
xml	http://www.w3.org/XML/1998/namespace
xsd	http://www.w3.org/2001/XMLSchema
ecore	http://www.eclipse.org/emf/2002/Ecore

### Schema Component Representation

```
<xsd:schema targetNamespace="http://www.fpml.org/FpML-5/confirmation"
documentRoot="FpML" nsPrefix="conf" package="org.fpml.confirmation" version="$Revision: 2590
$" elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xsd:include schemaLocation="fpml-asset-5-0.xsd"/>
  ...
</xsd:schema>
```

top

## Global Definitions

### Complex Type: Asian

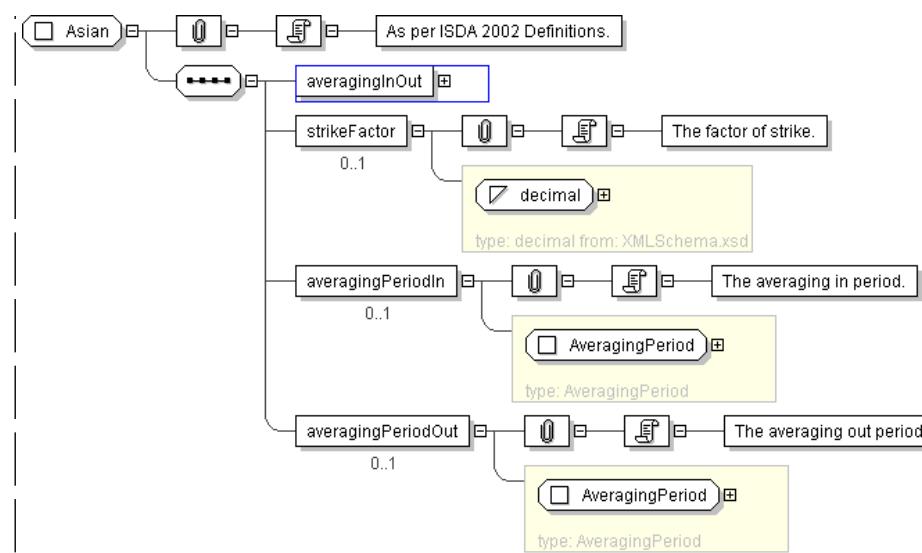
Super-types:	None
Sub-types:	None

Name	Asian
Used by (from the same schema document)	Model Group <a href="#">OptionFeature.model</a>
Abstract	no
Documentation	As per ISDA 2002 Definitions.

### XML Instance Representation

```
<...>
  <averagingInOut> AveragingInOutEnum </averagingInOut> [1]
  <strikeFactor> xsd:decimal </strikeFactor> [0..1]
    'The factor of strike.'
  <averagingPeriodIn> AveragingPeriod </averagingPeriodIn> [0..1]
    'The averaging in period.'
  <averagingPeriodOut> AveragingPeriod </averagingPeriodOut> [0..1]
    'The averaging out period.'
</...>
```

### Diagram

**Schema Component Representation**

```
<xsd:complexType name="Asian">
  <xsd:sequence>
    <xsd:element name="averagingInOut" type=" AveragingInOutEnum " />
    <xsd:element name="strikeFactor" type=" xsd:decimal " minOccurs="0 " />
    <xsd:element name="averagingPeriodIn" type=" AveragingPeriod " minOccurs="0 " />
    <xsd:element name="averagingPeriodOut" type=" AveragingPeriod " minOccurs="0 " />
  </xsd:sequence>
</xsd:complexType>
```

[top](#)**Complex Type: AveragingObservationList**

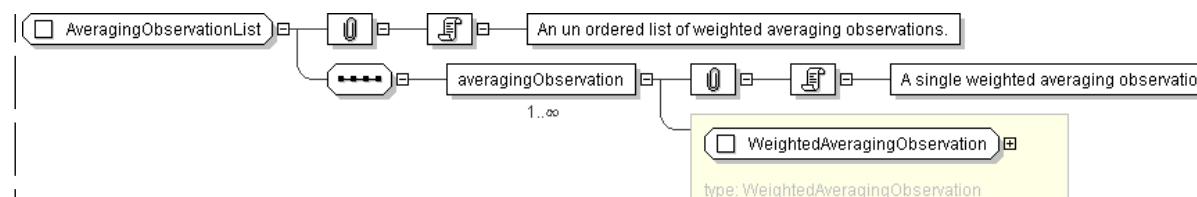
Super-types:	None
Sub-types:	None

Name	AveragingObservationList
Used by (from the same schema document)	Complex Type <a href="#">AveragingPeriod</a>
Abstract	no
Documentation	An un ordered list of weighted averaging observations.

**XML Instance Representation**

```
<...>
<averagingObservation> WeightedAveragingObservation </averagingObservation> [1..*]
  'A single weighted averaging observation.'
</...>
```

**Diagram**

**Schema Component Representation**

```

<xsd:complexType name="AveragingObservationList">
  <xsd:sequence>
    <xsd:element name="averagingObservation" type="WeightedAveragingObservation"
      " maxOccurs="unbounded" />
  </xsd:sequence>
</xsd:complexType>
  
```

[top](#)**Complex Type: AveragingPeriod**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None

<b>Name</b>	AveragingPeriod
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">Asian</a> , Complex Type <a href="#">Asian</a>
<b>Abstract</b>	no
<b>Documentation</b>	Period over which an average value is taken.

**XML Instance Representation**

```

<...>
<schedule> AveragingSchedule </schedule> [0..*]
  'A schedule for generating averaging observation dates.'

Start Choice [0..1]
  'A choice between unweighted and weighted averaging date and times.'

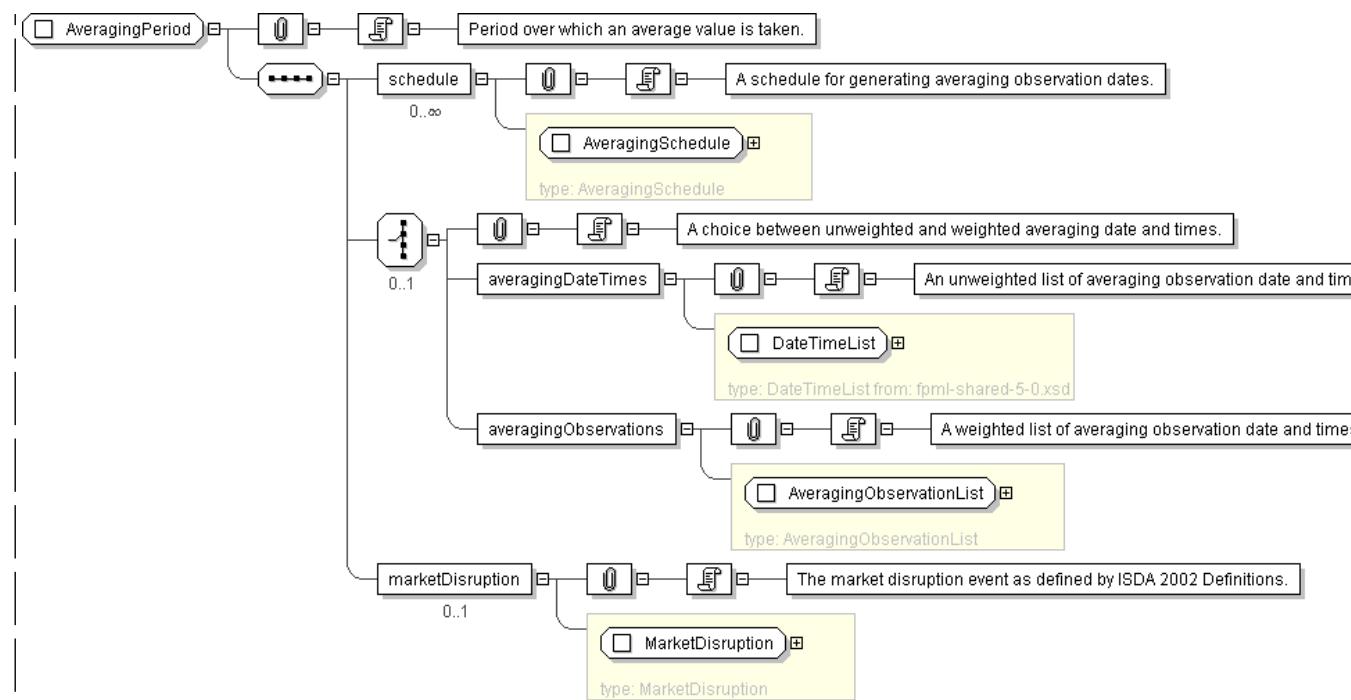
  <averagingDateTimes> DateTimeList </averagingDateTimes> [1]
  'An unweighted list of averaging observation date and times.'

  <averagingObservations> AveragingObservationList </averagingObservations> [1]
  'A weighted list of averaging observation date and times.'

End Choice
<marketDisruption> MarketDisruption </marketDisruption> [0..1]
  'The market disruption event as defined by ISDA 2002 Definitions.'

</...>
  
```

**Diagram**



#### Schema Component Representation

```

<xsd:complexType name="AveragingPeriod">
  <xsd:sequence>
    <xsd;element name="schedule" type=" AveragingSchedule " minOccurs="0" maxOccurs="unbounded"/>
    <xsd:choice minOccurs="0">
      <xsd;element name="averagingDateTimes" type=" DateTimeList "/>
      <xsd;element name="averagingObservations" type=" AveragingObservationList "/>
    </xsd:choice>
    <xsd;element name="marketDisruption" type=" MarketDisruption " minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>

```

[top](#)

#### Complex Type: AveragingSchedule

Super-types:	None
Sub-types:	None

Name	AveragingSchedule
Used by (from the same schema document)	Complex Type <a href="#">AveragingPeriod</a> , Complex Type <a href="#">TriggerEvent</a>
Abstract	no
Documentation	Method of generating a series of dates.

#### XML Instance Representation

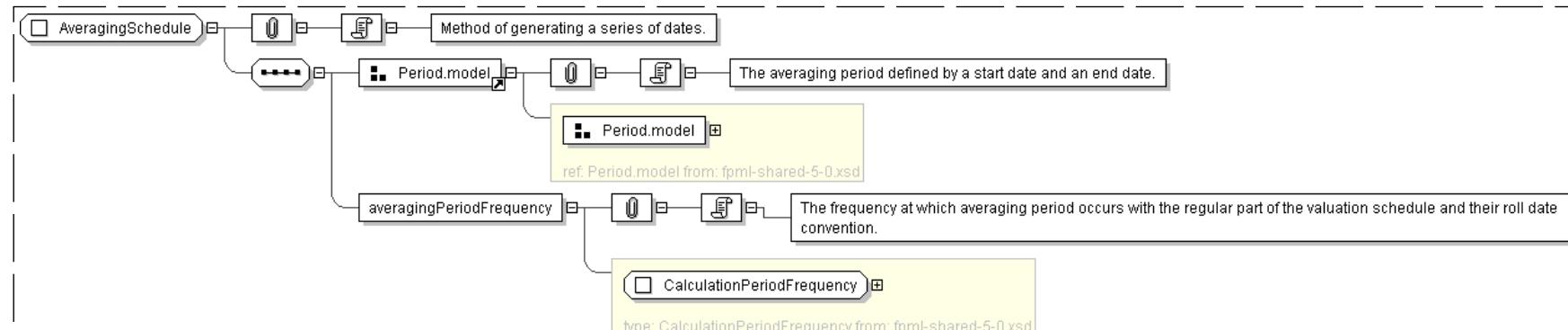
```

<...>
  <startDate> xsd:date </startDate> [1]
  'Date on which this period begins.'

  <endDate> xsd:date </endDate> [1]
  'Date on which this period ends.'

```

```
<averagingPeriodFrequency> CalculationPeriodFrequency </averagingPeriodFrequency> [1]
'The frequency at which averaging period occurs with the regular part of the valuation
schedule and their roll date convention.'
```

**Diagram****Schema Component Representation**

```
<xsd:complexType name="AveragingSchedule">
  <xsd:sequence>
    <xsd:group ref="# Period.model" />
    <!--frequency, frequencyType, weekNumber, day of the week were removed as their usage is
    covered in the CalculationPeriodFrequency -->
    <xsd:element name="averagingPeriodFrequency" type=" CalculationPeriodFrequency " />
  </xsd:sequence>
</xsd:complexType>
```

top

**Complex Type: Barrier**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None
<b>Name</b>	Barrier
<b>Used by (from the same schema document)</b>	Model Group <a href="#">OptionFeature.model</a>
<b>Abstract</b>	no
<b>Documentation</b>	As per ISDA 2002 Definitions.

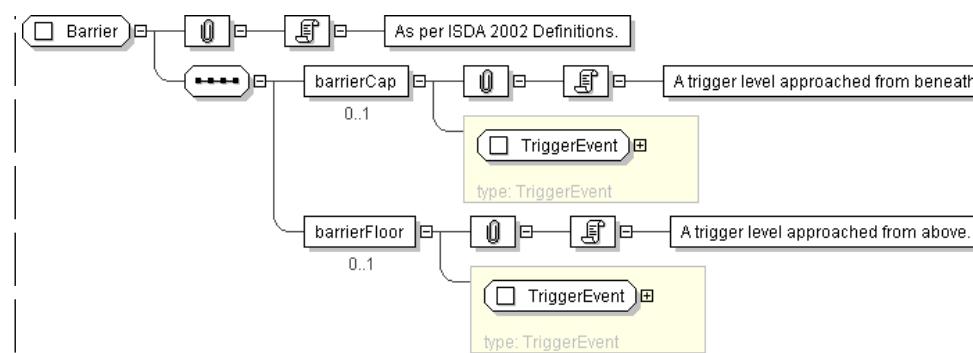
**XML Instance Representation**

```
<...>
<barrierCap> TriggerEvent </barrierCap> [0..1]
'A trigger level approached from beneath.'

<barrierFloor> TriggerEvent </barrierFloor> [0..1]
'A trigger level approached from above.'

</...>
```

**Diagram**

**Schema Component Representation**

```

<xsd:complexType name="Barrier">
  <xsd:sequence>
    <xsd:element name="barrierCap" type="#TriggerEvent" minOccurs="0"/>
    <xsd:element name="barrierFloor" type="#TriggerEvent" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>
  
```

[top](#)**Complex Type: CalendarSpread**

Super-types:	None
Sub-types:	None

Name	CalendarSpread
Used by (from the same schema document)	Complex Type <a href="#">StrategyFeature</a>
Abstract	no
Documentation	A type for defining a calendar spread feature.

**XML Instance Representation**

```

<...>
  <expirationDateTwo> AdjustableOrRelativeDate </expirationDateTwo> [1]
</...>
  
```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="CalendarSpread">
  <xsd:sequence>
    <xsd:element name="expirationDateTwo" type="AdjustableOrRelativeDate" />
  </xsd:sequence>
</xsd:complexType>
  
```

[top](#)**Complex Type: ClassifiedPayment**

Super-types:	<a href="#">NonNegativePayment</a> < <b>ClassifiedPayment</b> (by extension)
--------------	--

Sub-types:

None

**Name****Abstract****Documentation**

ClassifiedPayment

no

A classified non negative payment.

**XML Instance Representation**

```
<...>
<id=" xsd:ID [0..1]">
<payerPartyReference> PartyReference </payerPartyReference> [1]
'A reference to the party responsible for making the payments defined by this structure.'

<payerAccountReference> AccountReference </payerAccountReference> [0..1]
'A reference to the account responsible for making the payments defined by this structure.'

<receiverPartyReference> PartyReference </receiverPartyReference> [1]
'A reference to the party that receives the payments corresponding to this structure.'

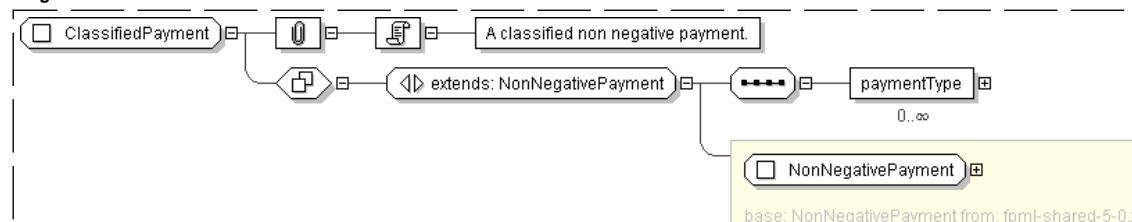
<receiverAccountReference> AccountReference </receiverAccountReference> [0..1]
'A reference to the account that receives the payments corresponding to this structure.'

<paymentDate> AdjustableOrRelativeDate </paymentDate> [1]
'The payment date, which can be expressed as either an adjustable or relative date.'

<paymentAmount> NonNegativeMoney </paymentAmount> [1]
'Non negative payment amount.'

<paymentType> PaymentType </paymentType> [0..*]
'Payment classification.'

</...>
```

**Diagram****Schema Component Representation**

```
<xsd:complexType name="ClassifiedPayment">
  <xsd:complexContent>
    <xsd:extension base=" NonNegativePayment ">
      <xsd:sequence>
        <xsd:element name="paymentType" type=" PaymentType " minOccurs="0" maxOccurs="unbounded" />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

top

**Complex Type: Composite**

Super-types:

None

Sub-types:

None

<b>Name</b>	Composite
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">FxFeature</a> , Complex Type <a href="#">FxFeature</a>
<b>Abstract</b>	no
<b>Documentation</b>	Specifies the conditions to be applied for converting into a reference currency when the actual currency rate is not determined upfront.

**XML Instance Representation**

```

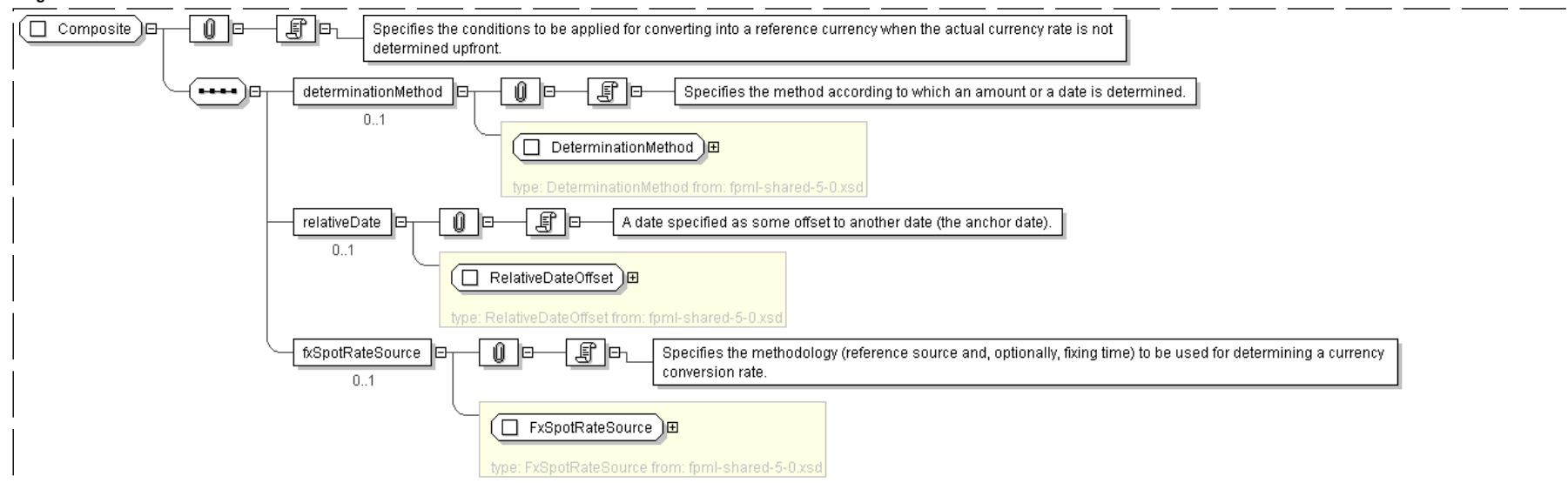
<...>
  <determinationMethod> DeterminationMethod </determinationMethod> [0..1]
  'Specifies the method according to which an amount or a date is determined.'

  <relativeDate> RelativeDateOffset </relativeDate> [0..1]
  'A date specified as some offset to another date (the anchor date).'

  <fxSpotRateSource> FxSpotRateSource </fxSpotRateSource> [0..1]
  'Specifies the methodology (reference source and, optionally, fixing time) to be used
  for determining a currency conversion rate.'

</...>

```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="Composite">
  <xsd:sequence>
    <xsd;element name="determinationMethod" type=" DeterminationMethod " minOccurs="0"/>
    <xsd;element name="relativeDate" type=" RelativeDateOffset " minOccurs="0"/>
    <xsd;element name="fxSpotRateSource" type=" FxSpotRateSource " minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>

```

top

**Complex Type: CreditEventNotice**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None

Name	CreditEventNotice
Used by (from the same schema document)	Complex Type <a href="#">CreditEvents</a>

Abstract	no
----------	----

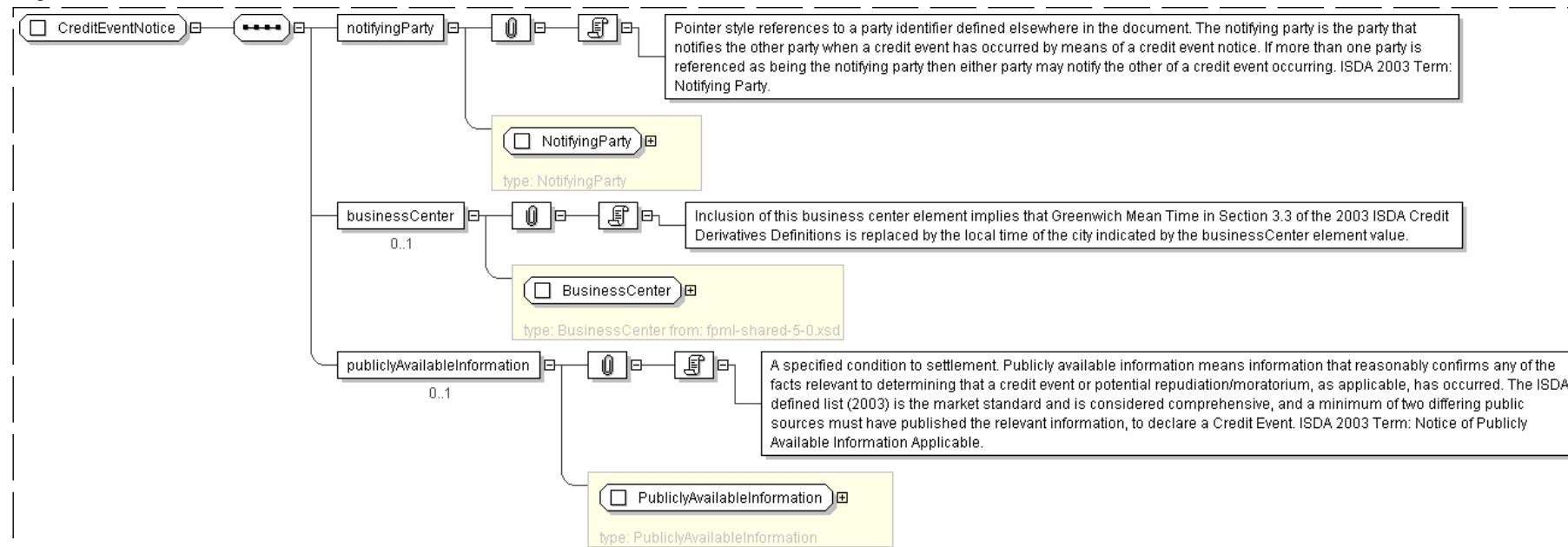
**XML Instance Representation**

```
<...>
<notifyingParty> NotifyingParty </notifyingParty> [1]
'Pointer style references to a party identifier defined elsewhere in the document.
The notifying party is the party that notifies the other party when a credit event has
occurred by means of a credit event notice. If more than one party is referenced as being
the notifying party then either party may notify the other of a credit event occurring.
ISDA 2003 Term: Notifying Party.'
```

```
<businessCenter> BusinessCenter </businessCenter> [0..1]
'Inclusion of this business center element implies that Greenwich Mean Time in Section 3.3
of the 2003 ISDA Credit Derivatives Definitions is replaced by the local time of the
city indicated by the businessCenter element value.'
```

```
<publiclyAvailableInformation> PubliclyAvailableInformation </
publiclyAvailableInformation> [0..1]
'A specified condition to settlement. Publicly available information means information
that reasonably confirms any of the facts relevant to determining that a credit event
or potential repudiation/moratorium, as applicable, has occurred. The ISDA defined list
(2003) is the market standard and is considered comprehensive, and a minimum of two
differing public sources must have published the relevant information, to declare a
Credit Event. ISDA 2003 Term: Notice of Publicly Available Information Applicable.'
```

```
</...>
```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="CreditEventNotice">
  <xsd:sequence>
    <xsd:element name="notifyingParty" type=" NotifyingParty " />
    <xsd:element name="businessCenter" type=" BusinessCenter " minOccurs="0"/>
    <xsd:element name="publiclyAvailableInformation" type=" PubliclyAvailableInformation
  
```

```

    " minOccurs="0" />
</xsd:sequence>
</xsd:complexType>
```

**Complex Type: CreditEvents**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None
<b>Name</b>	CreditEvents
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">Trigger</a>
<b>Abstract</b>	no

**XML Instance Representation**

```

<...
  id=" xsd:ID [0..1]">
  <bankruptcy> xsd:boolean </bankruptcy> [0..1]
  'A credit event. The reference entity has been dissolved or has become insolvent. It
  also covers events that may be a precursor to insolvency such as instigation of bankruptcy
  or insolvency proceedings. Sovereign trades are not subject to Bankruptcy as \"technically\"'
  a Sovereign cannot become bankrupt. ISDA 2003 Term: Bankruptcy.'

  <failureToPay> FailureToPay </failureToPay> [0..1]
  'A credit event. This credit event triggers, after the expiration of any applicable
  grace period, if the reference entity fails to make due payments in an aggregate amount of
  not less than the payment requirement on one or more obligations (e.g. a missed
  coupon payment). ISDA 2003 Term: Failure to Pay.'

  <failureToPayPrincipal> xsd:boolean </failureToPayPrincipal> [0..1]
  'A credit event. Corresponds to the failure by the Reference Entity to pay an
  expected principal amount or the payment of an actual principal amount that is less than
  the expected principal amount. ISDA 2003 Term: Failure to Pay Principal.'

  <failureToPayInterest> xsd:boolean </failureToPayInterest> [0..1]
  'A credit event. Corresponds to the failure by the Reference Entity to pay an expected
  interest amount or the payment of an actual interest amount that is less than the
  expected interest amount. ISDA 2003 Term: Failure to Pay Interest.'

  <obligationDefault> xsd:boolean </obligationDefault> [0..1]
  'A credit event. One or more of the obligations have become capable of being declared due
  and payable before they would otherwise have been due and payable as a result of, or on
  the basis of, the occurrence of a default, event of default or other similar condition or
  event other than failure to pay. ISDA 2003 Term: Obligation Default.'

  <obligationAcceleration> xsd:boolean </obligationAcceleration> [0..1]
  'A credit event. One or more of the obligations have been declared due and payable before
  they would otherwise have been due and payable as a result of, or on the basis of,
  the occurrence of a default, event of default or other similar condition or event other
  than failure to pay (preferred by the market over Obligation Default, because more
  definitive and encompasses the definition of Obligation Default - this is more favorable to
  the Seller). Subject to the default requirement amount. ISDA 2003 Term:
  Obligation Acceleration.'

  <repudiationMoratorium> xsd:boolean </repudiationMoratorium> [0..1]
  'A credit event. The reference entity, or a governmental authority, either refuses to
  recognise or challenges the validity of one or more obligations of the reference entity,
  or imposes a moratorium thereby postponing payments on one or more of the obligations of
  the reference entity. Subject to the default requirement amount. ISDA 2003 Term:
  Repudiation/Moratorium.'

  <restructuring> Restructuring </restructuring> [0..1]
```

'A credit event. A restructuring is an event that materially impacts the reference entity's obligations, such as an interest rate reduction, principal reduction, deferral of interest or principal, change in priority ranking, or change in currency or composition of payment. ISDA 2003 Term: Restructuring.'

<distressedRatingsDowngrade> xsd:boolean </distressedRatingsDowngrade> [0..1]

'A credit event. Results from the fact that the rating of the reference obligation is downgraded to a distressed rating level. From a usage standpoint, this credit event is typically not applicable in case of RMBS trades.'

<maturityExtension> xsd:boolean </maturityExtension> [0..1]

'A credit event. Results from the fact that the underlier fails to make principal payments as expected.'

<writedown> xsd:boolean </writedown> [0..1]

'A credit event. Results from the fact that the underlier writes down its outstanding principal amount.'

<impliedWritedown> xsd:boolean </impliedWritedown> [0..1]

'A credit event. Results from the fact that losses occur to the underlying instruments that do not result in reductions of the outstanding principal of the reference obligation.'

<defaultRequirement> Money </defaultRequirement> [0..1]

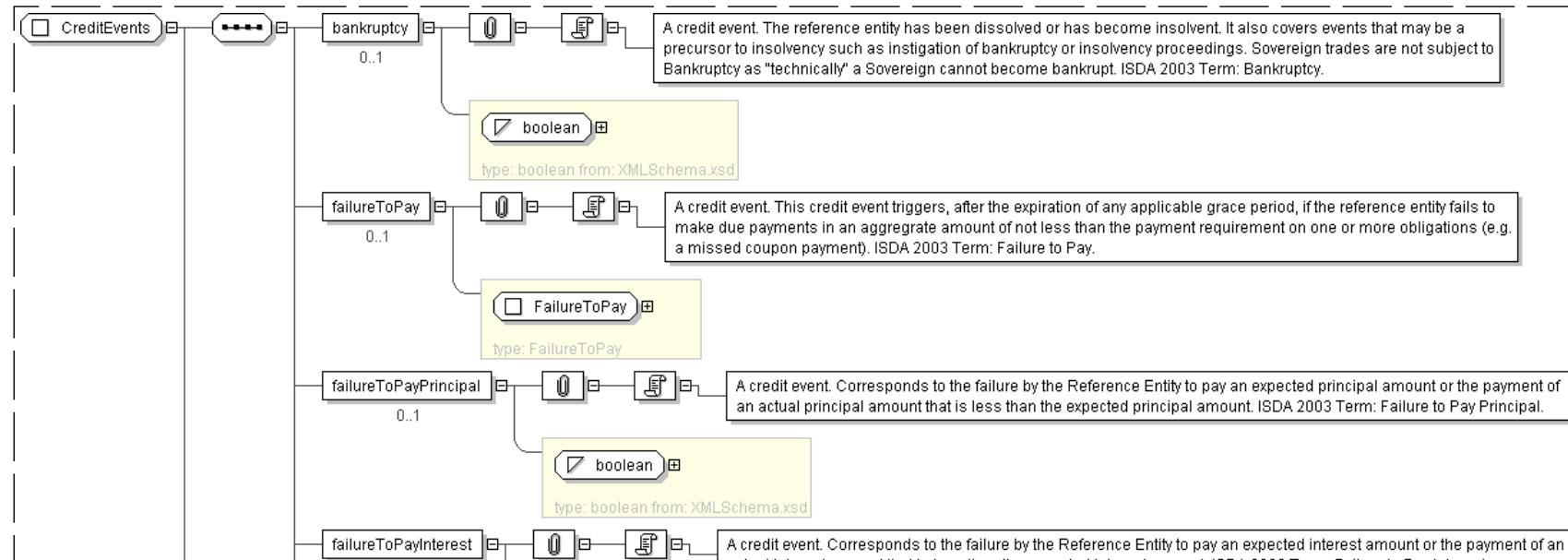
'In relation to certain credit events, serves as a threshold for Obligation Acceleration, Obligation Default, Repudiation/Moratorium and Restructuring. Market standard is USD 10,000,000 (JPY 1,000,000,000 for all Japanese Yen trades). This is applied on an aggregate or total basis across all Obligations of the Reference Entity. Used to prevent technical/operational errors from triggering credit events. ISDA 2003 Term: Default Requirement.'

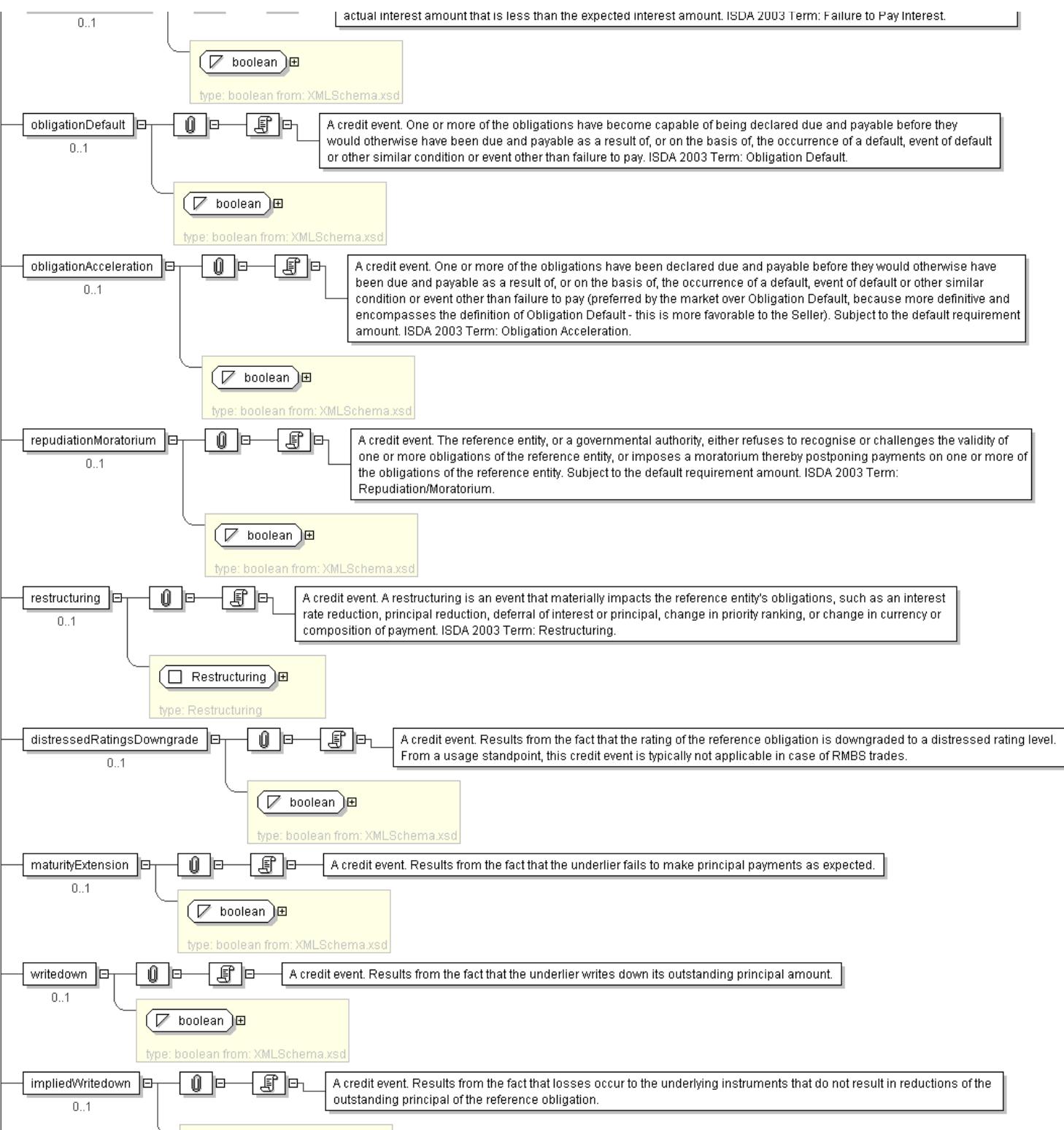
<creditEventNotice> CreditEventNotice </creditEventNotice> [0..1]

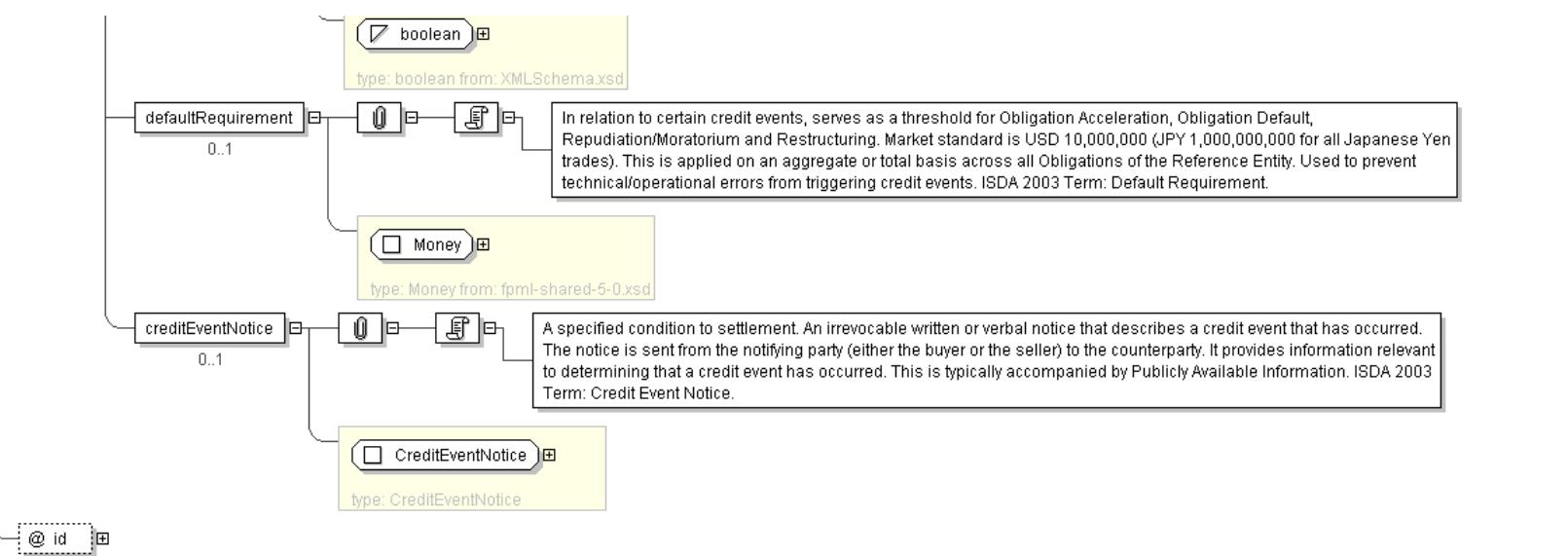
'A specified condition to settlement. An irrevocable written or verbal notice that describes a credit event that has occurred. The notice is sent from the notifying party (either the buyer or the seller) to the counterparty. It provides information relevant to determining that a credit event has occurred. This is typically accompanied by Publicly Available Information. ISDA 2003 Term: Credit Event Notice.'

</...>

#### Diagram





**Schema Component Representation**

```

<xsd:complexType name="CreditEvents">
  <xsd:sequence>
    <xsd:element name="bankruptcy" type="xsd:boolean" minOccurs="0"/>
    <xsd:element name="failureToPay" type="FailureToPay" minOccurs="0"/>
    <xsd:element name="failureToPayPrincipal" type="xsd:boolean" minOccurs="0"/>
    <xsd:element name="failureToPayInterest" type="xsd:boolean" minOccurs="0"/>
    <xsd:element name="obligationDefault" type="xsd:boolean" minOccurs="0"/>
    <xsd:element name="obligationAcceleration" type="xsd:boolean" minOccurs="0"/>
    <xsd:element name="repudiationMoratorium" type="xsd:boolean" minOccurs="0"/>
    <xsd:element name="restructuring" type="Restructuring" minOccurs="0"/>
    <xsd:element name="distressedRatingsDowngrade" type="xsd:boolean" minOccurs="0"/>
    <xsd:element name="maturityExtension" type="xsd:boolean" minOccurs="0"/>
    <xsd:element name="writedown" type="xsd:boolean" minOccurs="0"/>
    <xsd:element name="impliedWritedown" type="xsd:boolean" minOccurs="0"/>
    <xsd:element name="defaultRequirement" type="Money" minOccurs="0"/>
    <xsd:element name="creditEventNotice" type="CreditEventNotice" minOccurs="0"/>
  </xsd:sequence>
  <xsd:attribute name="id" type="xsd:ID" use="optional"/>
</xsd:complexType>

```

top

**Complex Type: CreditEventsReference**

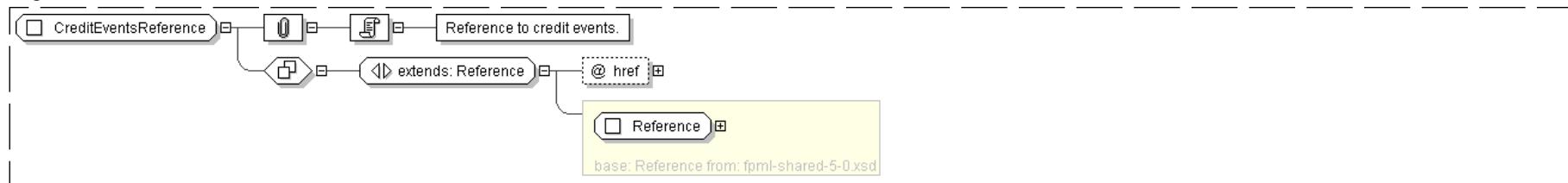
Super-types:	<a href="#">Reference</a> < <b>CreditEventsReference</b> (by extension)
Sub-types:	None
Name	CreditEventsReference
Used by (from the same schema document)	Complex Type <a href="#">Trigger</a>
Abstract	no
Documentation	Reference to credit events.

**XML Instance Representation**

```

<...
  href="#" type="xsd:IDREF" [1]" />

```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="CreditEventsReference">
  <xsd:complexContent>
    <xsd:extension base=" Reference ">
      <xsd:attribute name="href" type=" xsd:IDREF " use="required" reference="CreditEvents" />
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
  
```

top

**Complex Type: FailureToPay**

Super-types:	None
Sub-types:	None

Name	FailureToPay
Used by (from the same schema document)	Complex Type <a href="#">CreditEvents</a>
Abstract	no

**XML Instance Representation**

```

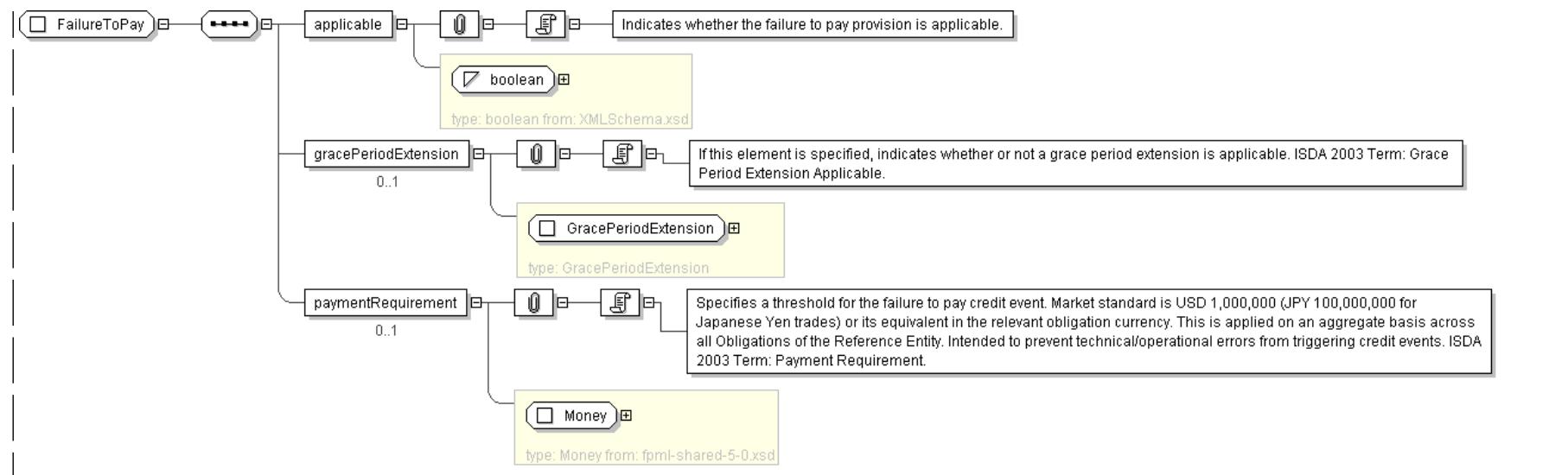
<...>
<applicable> xsd:boolean </applicable> [1]
  'Indicates whether the failure to pay provision is applicable.'

<gracePeriodExtension> GracePeriodExtension </gracePeriodExtension> [0..1]
  'If this element is specified, indicates whether or not a grace period extension is
  applicable. ISDA 2003 Term: Grace Period Extension Applicable.'

<paymentRequirement> Money </paymentRequirement> [0..1]
  'Specifies a threshold for the failure to pay credit event. Market standard is USD
  1,000,000 (JPY 100,000,000 for Japanese Yen trades) or its equivalent in the
  relevant obligation currency. This is applied on an aggregate basis across all Obligations
  of the Reference Entity. Intended to prevent technical/operational errors from
  triggering credit events. ISDA 2003 Term: Payment Requirement.'

</...>
  
```

**Diagram**

**Schema Component Representation**

```

<xsd:complexType name="FailureToPay">
  <xsd:sequence>
    <xsd:element name="applicable" type="xsd:boolean" />
    <xsd:element name="gracePeriodExtension" type="GracePeriodExtension" minOccurs="0"/>
    <xsd:element name="paymentRequirement" type="Money" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>
  
```

top

**Complex Type: FeaturePayment**Super-types: [PaymentBase](#) < **FeaturePayment** (by extension)

Sub-types: None

Name	FeaturePayment
Used by (from the same schema document)	Complex Type <a href="#">TriggerEvent</a>
Abstract	no
Documentation	Payment made following trigger occurrence.

**XML Instance Representation**

```

<... id=" xsd:ID [0..1]">
<payerPartyReference> PartyReference </payerPartyReference> [1]
'A reference to the party responsible for making the payments defined by this structure.'

<payerAccountReference> AccountReference </payerAccountReference> [0..1]
'A reference to the account responsible for making the payments defined by this structure.'

<receiverPartyReference> PartyReference </receiverPartyReference> [1]
'A reference to the party that receives the payments corresponding to this structure.'

<receiverAccountReference> AccountReference </receiverAccountReference> [0..1]
'A reference to the account that receives the payments corresponding to this structure.'
  
```

Start [Choice](#) [1]

```

<levelPercentage> xsd:decimal </levelPercentage> [1]
'The trigger level percentage.'

<amount> NonNegativeDecimal </amount> [1]
'The monetary quantity in currency units.'

End Choice
<time> TimeTypeEnum </time> [0..1]
'The feature payment time.'

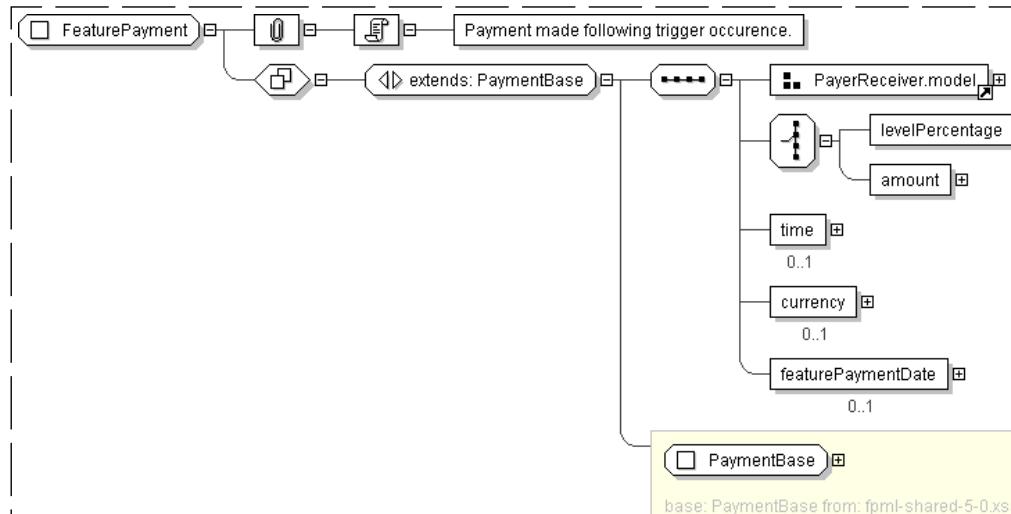
<currency> Currency </currency> [0..1]
'The currency in which an amount is denominated.'

<featurePaymentDate> AdjustableOrRelativeDate </featurePaymentDate> [0..1]
'The feature payment date.'

</...>

```

## Diagram



## Schema Component Representation

```

<xsd:complexType name="FeaturePayment">
  <xsd:complexContent>
    <xsd:extension base=" PaymentBase ">
      <xsd:sequence>
        <xsd:group ref=" PayerReceiver.model " />
        <xsd:choice>
          <xsd:element name="levelPercentage" type=" xsd:decimal "/>
          <xsd:element name="amount" type=" NonNegativeDecimal "/>
        </xsd:choice>
        <xsd:element name="time" type=" TimeTypeEnum " minOccurs="0"/>
        <xsd:element name="currency" type=" Currency " minOccurs="0"/>
        <xsd:element name="featurePaymentDate" type=" AdjustableOrRelativeDate " minOccurs="0"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

**Complex Type: FrequencyType**

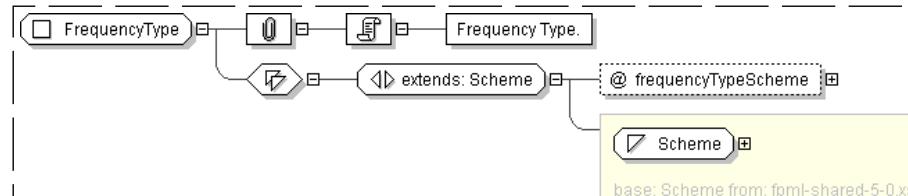
**Super-types:** [Scheme](#) < **FrequencyType** (by extension)

**Sub-types:** None

<b>Name</b>	FrequencyType
<b>Abstract</b>	no
<b>Documentation</b>	Frequency Type.

**XML Instance Representation**

```
<...>
<frequencyTypeScheme=" xsd:anyURI [0..1]">
  Scheme
</...>
```

**Diagram****Schema Component Representation**

```
<xsd:complexType name="FrequencyType">
  <xsd:simpleContent>
    <xsd:extension base=" Scheme ">
      <xsd:attribute name="frequencyTypeScheme" type=" xsd:anyURI " />
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
```

top

**Complex Type: FxFeature**

**Super-types:** None

**Sub-types:** None

<b>Name</b>	FxFeature
-------------	-----------

**Used by (from the same schema document)** Model Group [OptionBaseFeature.model](#)

<b>Abstract</b>	no
-----------------	----

**Documentation** A type for defining Fx Features.

**XML Instance Representation**

```
<...>
<referenceCurrency> IdentifiedCurrency </referenceCurrency> [1]
  'Specifies the reference currency of the trade.'
```

```
Start Choice [1]
<composite> Composite </composite> [1]
```

'If "Composite" is specified as the Settlement Type in the relevant Transaction Supplement, an amount in the Settlement Currency, determined by the Calculation Agent as being equal to the number of Options exercised or deemed exercised, multiplied by: (Settlement Price - Strike Price) / (Strike Price - Settlement Price) x Multiplier provided that if the above is equal to a negative amount the Option Cash Settlement Amount shall be deemed to be zero.'

```
<quanto> Quanto </quanto> [1]
```

'If "Quanto" is specified as the Settlement Type in the relevant Transaction Supplement, an amount, as determined by the Calculation Agent in accordance with the Section 8.2 of the Equity Definitions.'

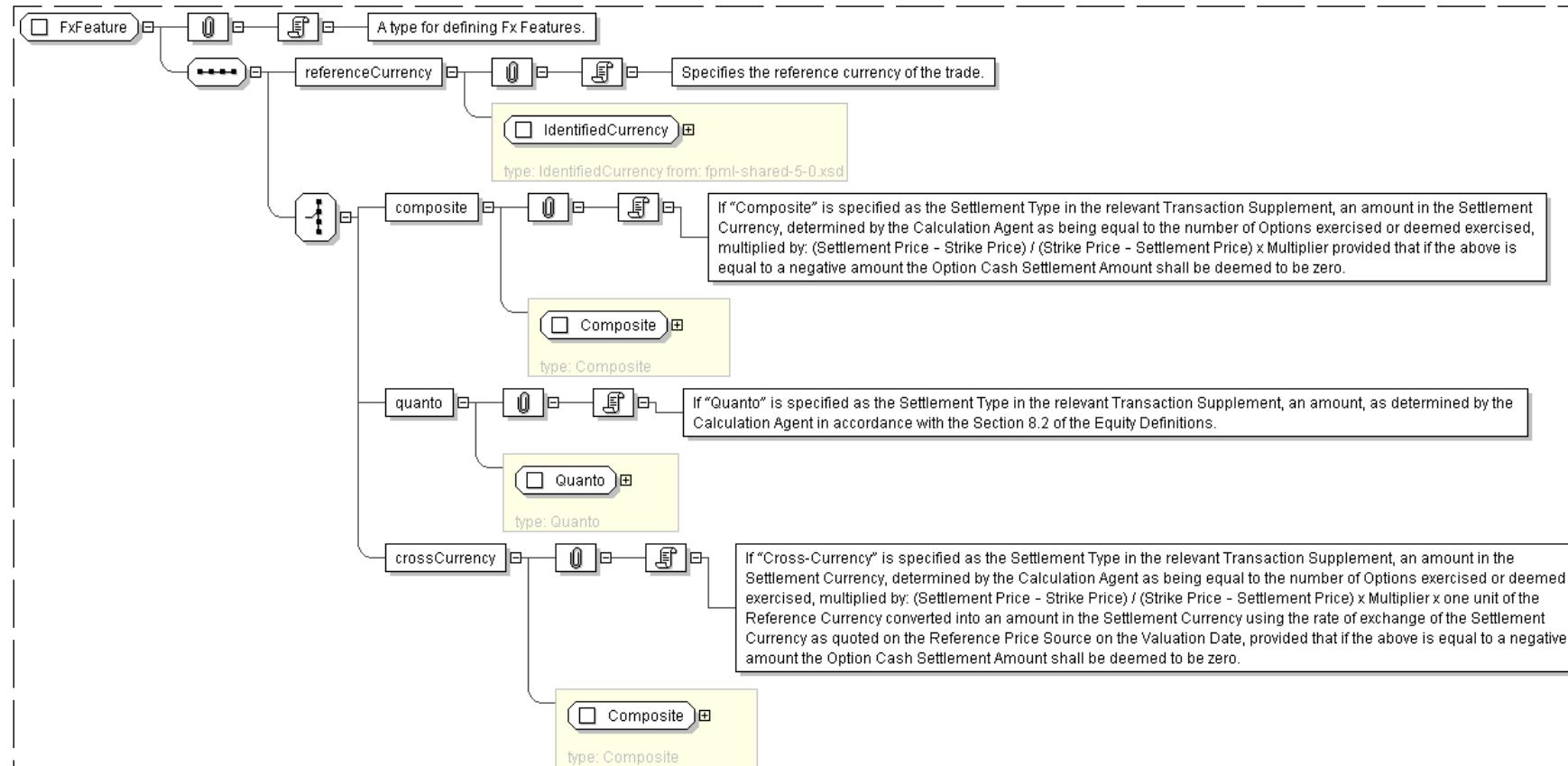
<crossCurrency> Composite </crossCurrency> [1]

'If "Cross-Currency" is specified as the Settlement Type in the relevant Transaction Supplement, an amount in the Settlement Currency, determined by the Calculation Agent as being equal to the number of Options exercised or deemed exercised, multiplied by: (Settlement Price - Strike Price) / (Strike Price - Settlement Price) x Multiplier x one unit of the Reference Currency converted into an amount in the Settlement Currency using the rate of exchange of the Settlement Currency as quoted on the Reference Price Source on the Valuation Date, provided that if the above is equal to a negative amount the Option Cash Settlement Amount shall be deemed to be zero.'

End Choice

</...>

#### Diagram



#### Schema Component Representation

```

<xsd:complexType name="FxFeature">
    <xsd:sequence>
        <xsd:element name="referenceCurrency" type=" IdentifiedCurrency " />
        <xsd:choice>
            <xsd:element name="composite" type=" Composite " />
            <xsd:element name="quanto" type=" Quanto " />
            <xsd:element name="crossCurrency" type=" Composite " />
        </xsd:choice>
    </xsd:sequence>
</xsd:complexType>

```

&lt;/xsd:complexType&gt;

[top](#)**Complex Type: GracePeriodExtension**

Super-types:	None
Sub-types:	None

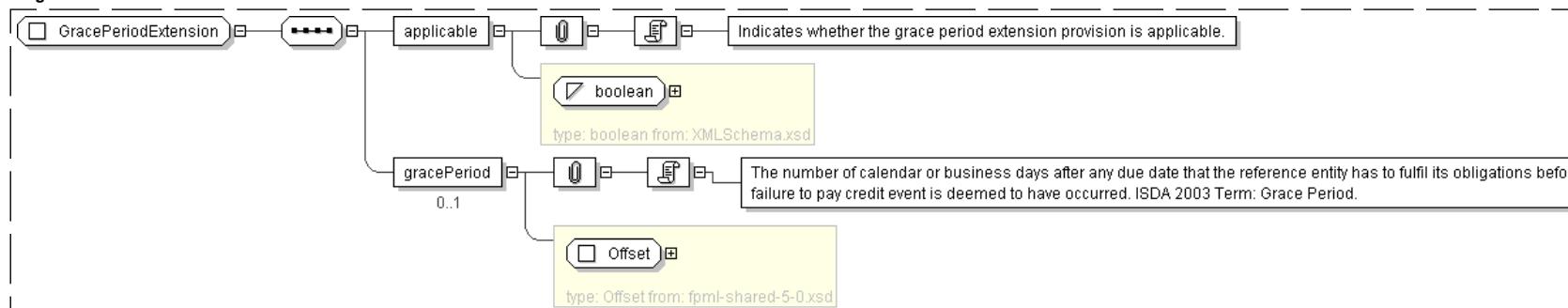
Name	GracePeriodExtension
Used by (from the same schema document)	Complex Type <a href="#">FailureToPay</a>
Abstract	no

**XML Instance Representation**

```
<...>
<applicable> xsd:boolean </applicable> [1]
'Indicates whether the grace period extension provision is applicable.'

<gracePeriod> Offset </gracePeriod> [0..1]
'The number of calendar or business days after any due date that the reference entity has
to fulfil its obligations before a failure to pay credit event is deemed to have occurred.
ISDA 2003 Term: Grace Period.'

</...>
```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="GracePeriodExtension">
  <xsd:sequence>
    <xsd:element name="applicable" type="xsd:boolean" />
    <xsd:element name="gracePeriod" type="Offset" minOccurs="0" />
  </xsd:sequence>
</xsd:complexType>

```

[top](#)**Complex Type: Knock**

Super-types:	None
Sub-types:	None

Name	Knock
Used by (from the same schema document)	Model Group <a href="#">OptionFeature.model</a>
Abstract	no

**Documentation**

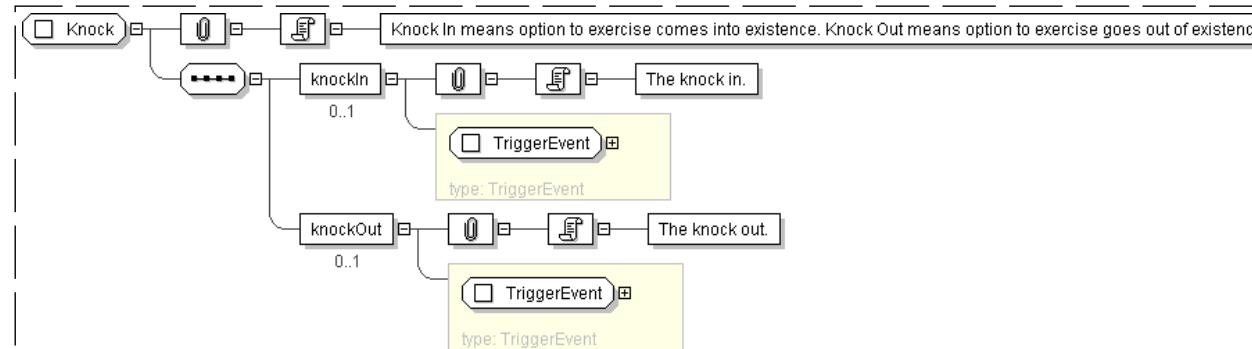
Knock In means option to exercise comes into existence. Knock Out means option to exercise goes out of existence.

**XML Instance Representation**

```
<...>
<knockIn> TriggerEvent </knockIn> [0..1]
'The knock in.'

<knockOut> TriggerEvent </knockOut> [0..1]
'The knock out.'

</...>
```

**Diagram****Schema Component Representation**

```
<xsd:complexType name="Knock">
  <xsd:sequence>
    <xsd:element name="knockIn" type="TriggerEvent" minOccurs="0"/>
    <xsd:element name="knockOut" type="TriggerEvent" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>
```

top

**Complex Type: MarketDisruption**

**Super-types:** [Scheme](#) < **MarketDisruption** (by extension)

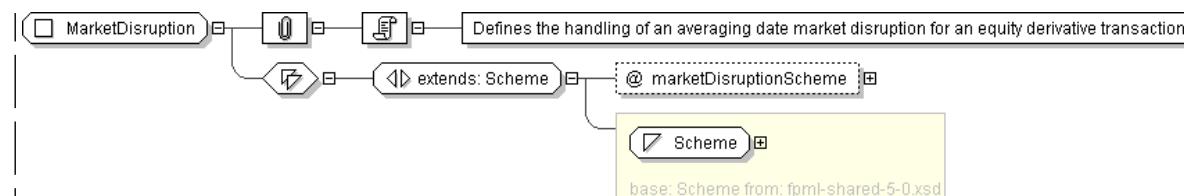
**Sub-types:** None

<b>Name</b>	MarketDisruption
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">AveragingPeriod</a>
<b>Abstract</b>	no
<b>Documentation</b>	Defines the handling of an averaging date market disruption for an equity derivative transaction.

**XML Instance Representation**

```
<...
marketDisruptionScheme=" xsd:anyURI [0..1]">
Scheme
</...>
```

**Diagram**

**Schema Component Representation**

```
<xsd:complexType name="MarketDisruption">
  <xsd:simpleContent>
    <xsd:extension base=" Scheme ">
      <xsd:attribute name="marketDisruptionScheme" type=" xsd:anyURI " default="http://www.fpml.org/coding-scheme/market-disruption"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
```

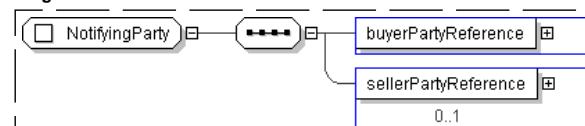
[top](#)**Complex Type: NotifyingParty**

Super-types:	None
Sub-types:	None

Name	NotifyingParty
Used by (from the same schema document)	Complex Type <a href="#">CreditEventNotice</a>
Abstract	no

**XML Instance Representation**

```
<...>
  <buyerPartyReference> PartyReference </buyerPartyReference> [1]
  <sellerPartyReference> PartyReference </sellerPartyReference> [0..1]
</...>
```

**Diagram****Schema Component Representation**

```
<xsd:complexType name="NotifyingParty">
  <xsd:sequence>
    <xsd:element name="buyerPartyReference" type=" PartyReference "/>
    <xsd:element name="sellerPartyReference" type=" PartyReference " minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>
```

[top](#)**Complex Type: OptionBase**

Super-types:	<a href="#">Product</a> < <b>OptionBase</b> (by extension)
Sub-types:	<ul style="list-style-type: none"> <li><a href="#">OptionBaseExtended</a> (by extension)</li> </ul>

Name	OptionBase
------	------------

**Abstract**

yes

**Documentation**

A type for defining the common features of options.

**XML Instance Representation**

```
<...>
<id=" xsd:ID [0..1]">
<productType> ProductType </productType> [0..*]
'A classification of the type of product. FpML defines a simple product categorization using
a coding scheme.'

<productId> ProductId </productId> [0..*]
'A product 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.'

<buyerPartyReference> PartyReference </buyerPartyReference> [1]
'A reference to the party that buys this instrument, i.e. pays for this instrument and
receives the rights defined by it. See 2000 ISDA definitions Article 11.1 (b). In the case
of FRAs this is the fixed rate payer.'

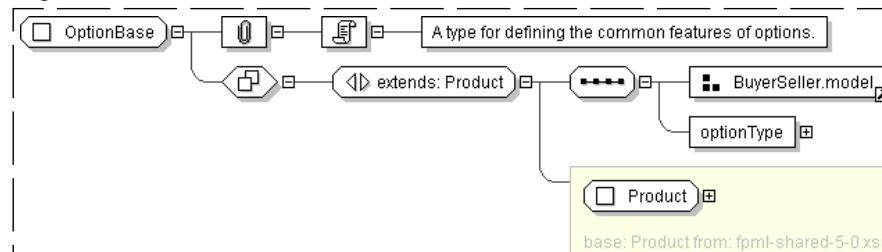
<buyerAccountReference> AccountReference </buyerAccountReference> [0..1]
'A reference to the account that buys this instrument.'

<sellerPartyReference> PartyReference </sellerPartyReference> [1]
'A reference to the party that sells ("writes") this instrument, i.e. that grants the
rights defined by this instrument and in return receives a payment for it. See 2000
ISDA definitions Article 11.1 (a). In the case of FRAs this is the floating rate payer.'

<sellerAccountReference> AccountReference </sellerAccountReference> [0..1]
'A reference to the account that sells this instrument.'

<optionType> OptionTypeEnum </optionType> [1]
'The type of option transaction. From a usage standpoint, put/call is the default option
type, while payer/receiver indicator is used for options index credit default
swaps, consistently with the industry practice. Straddle is used for the case of
straddle strategy, that combine a call and a put with the same strike.'

</...>
```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="OptionBase" abstract="true">
  <xsd:complexContent>
    <xsd:extension base=" Product ">
      <xsd:sequence>
        <xsd:group ref=" BuyerSeller.model "/>
        <xsd:element name="optionType" type=" OptionTypeEnum "/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

## Complex Type: OptionBaseExtended

Super-types:	<a href="#">Product &lt; OptionBase</a> (by extension) < <b>OptionBaseExtended</b> (by extension)
Sub-types:	None

Name	OptionBaseExtended
Abstract	yes
Documentation	Base type for options starting with the 4-3 release, until we refactor the schema as part of the 5-0 release series.

### XML Instance Representation

```
<...>
  id=" xsd:ID [0..1]">
    <productType> ProductType </productType> [0..*]
      'A classification of the type of product. FpML defines a simple product categorization using a coding scheme.'
```

```
<productId> ProductId </productId> [0..*]
  'A product 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.'
```

```
<buyerPartyReference> PartyReference </buyerPartyReference> [1]
  'A reference to the party that buys this instrument, ie. pays for this instrument and receives the rights defined by it. See 2000 ISDA definitions Article 11.1 (b). In the case of FRAs this is the fixed rate payer.'
```

```
<buyerAccountReference> AccountReference </buyerAccountReference> [0..1]
  'A reference to the account that buys this instrument.'
```

```
<sellerPartyReference> PartyReference </sellerPartyReference> [1]
  'A reference to the party that sells (\\"writes\\") this instrument, i.e. that grants the rights defined by this instrument and in return receives a payment for it. See 2000 ISDA definitions Article 11.1 (a). In the case of FRAs this is the floating rate payer.'
```

```
<sellerAccountReference> AccountReference </sellerAccountReference> [0..1]
  'A reference to the account that sells this instrument.'
```

```
<optionType> OptionTypeEnum </optionType> [1]
  'The type of option transaction. From a usage standpoint, put/call is the default option type, while payer/receiver indicator is used for options index credit/default swaps, consistently with the industry practice. Straddle is used for the case of straddle strategy, that combine a call and a put with the same strike.'
```

```
<premium> Premium </premium> [0..1]
  'The option premium payable by the buyer to the seller.'
```

```
<exercise> ... </exercise> [1]
<exerciseProcedure> ExerciseProcedure </exerciseProcedure> [1]
  'A set of parameters defining procedures associated with the exercise.'
```

```
<feature> OptionFeature </feature> [0..1]
  'An Option feature such as quanto, asian, barrier, knock.'
```

Start Choice [0..1]

'A choice between an explicit representation of the notional amount, or a reference to a notional amount defined elsewhere in this document.'

```
<notionalReference> NotionalAmountReference </notionalReference> [1]
```

```

|   <notionalAmount> Money </notionalAmount> [1]
| End Choice
Start Group: OptionDenomination.model [0..1]
<optionEntitlement> PositiveDecimal </optionEntitlement> [1]
'The number of units of underlyer per option comprised in the option transaction.'

<entitlementCurrency> Currency </entitlementCurrency> [0..1]
'TODO'

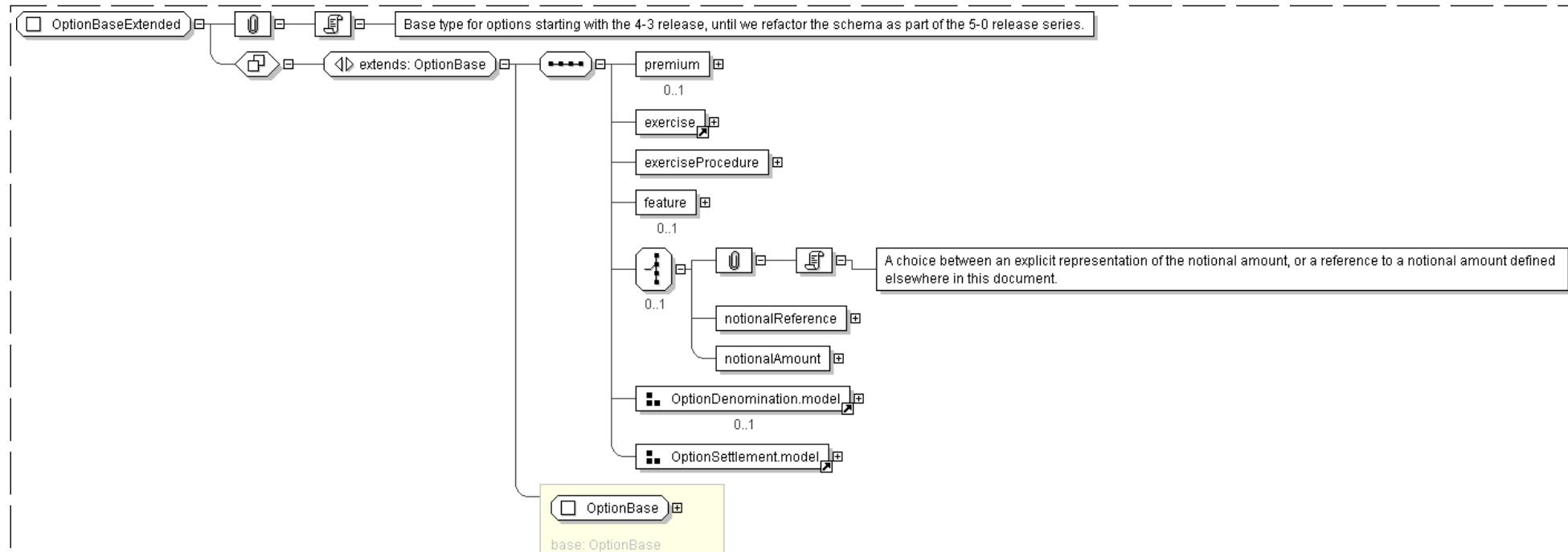
<numberOfOptions> PositiveDecimal </numberOfOptions> [0..1]
'The number of options comprised in the option transaction.'

End Group: OptionDenomination.model
<settlementType> SettlementTypeEnum </settlementType> [0..1]
<settlementDate> AdjustableOrRelativeDate </settlementDate> [0..1]
Start Group: SettlementAmountOrCurrency.model [0..1]
Start Choice [1]
    <settlementAmount> Money </settlementAmount> [1]
    'Settlement Amount'

    <settlementCurrency> Currency </settlementCurrency> [1]
    'Settlement Currency for use where the Settlement Amount cannot be known in advance'

End Choice
End Group: SettlementAmountOrCurrency.model
</...>

```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="OptionBaseExtended" abstract="true">
  <xsd:complexContent>
    <xsd:extension base=" OptionBase ">
      <xsd:sequence>

```

```

<xsd:element name="premium" type=" Premium " minOccurs="0"/>
<xsd:element ref=" exercise "/>
<xsd:element name="exerciseProcedure" type=" ExerciseProcedure " />
<xsd:element name="feature" type=" OptionFeature " minOccurs="0"/>
<xsd:choice minOccurs="0">
    <xsd:element name="notionalReference" type=" NotionalAmountReference " />
    <xsd:element name="notionalAmount" type=" Money " />
</xsd:choice>
<xsd:group ref=" OptionDenomination.model " minOccurs="0"/>
<xsd:group ref=" OptionSettlement.model " />
</xsd:sequence>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>

```

top

## Complex Type: OptionFeature

Super-types:	None
Sub-types:	None

Name	OptionFeature
Used by (from the same schema document)	Complex Type <a href="#">OptionBaseExtended</a>
Abstract	no
Documentation	A type for defining option features.

### XML Instance Representation

```

<...>
<fxFeature> FxFeature </fxFeature> [0..1]
'A quanto or composite FX feature.'

<strategyFeature> StrategyFeature </strategyFeature> [0..1]
'A simple strategy feature.'

<asian> Asian </asian> [0..1]
'An option where and average price is taken on valuation.'

<barrier> Barrier </barrier> [0..1]
'An option with a barrier feature.'

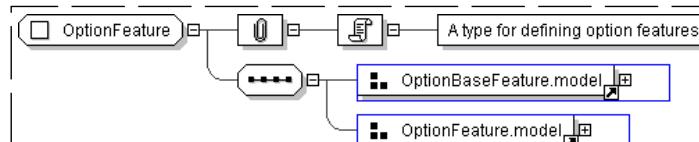
<knock> Knock </knock> [0..1]
'A knock feature.'

<passThrough> PassThrough </passThrough> [0..1]
'Pass through payments from the underlyer, such as dividends.'

</...>

```

### Diagram



### Schema Component Representation

```

<xsd:complexType name="OptionFeature">
<xsd:sequence>

```

```

<xsd:group ref="# OptionBaseFeature.model" />
<xsd:group ref="# OptionFeature.model" />
</xsd:sequence>
</xsd:complexType>

```

[top](#)

## Complex Type: OptionNumericStrike

**Super-types:** None  
**Sub-types:**

- [OptionStrike](#) (by extension)

<b>Name</b>	OptionNumericStrike
<b>Abstract</b>	no
<b>Documentation</b>	A type for defining the strike price for an option as a numeric value without currency.

### XML Instance Representation

```

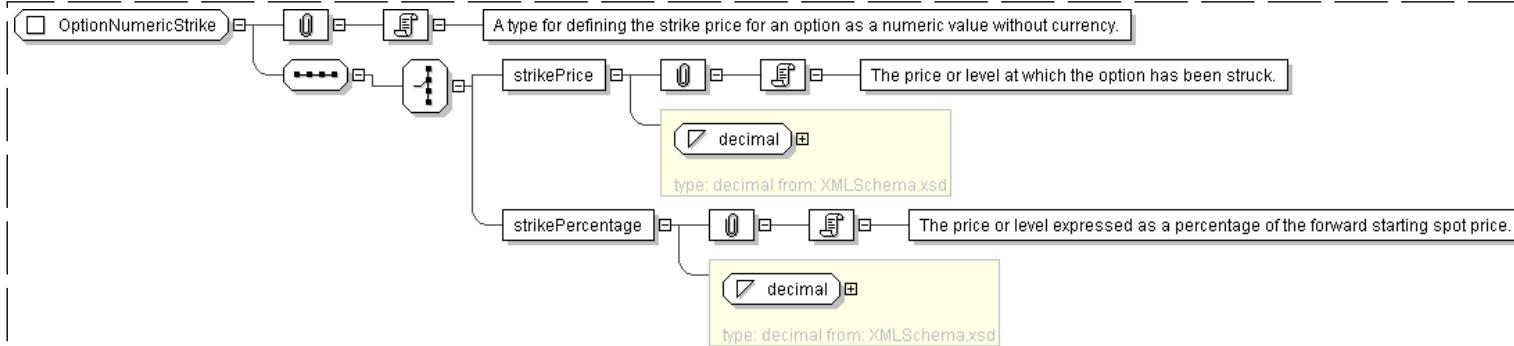
<...>
Start Choice [1]
  <strikePrice> xsd:decimal </strikePrice> [1]
    'The price or level at which the option has been struck.'

  <strikePercentage> xsd:decimal </strikePercentage> [1]
    'The price or level expressed as a percentage of the forward starting spot price.'

End Choice
</...>

```

### Diagram



### Schema Component Representation

```

<xsd:complexType name="OptionNumericStrike">
  <xsd:sequence>
    <xsd:choice>
      <xsd:element name="strikePrice" type="xsd:decimal" />
      <xsd:element name="strikePercentage" type="xsd:decimal" />
    </xsd:choice>
  </xsd:sequence>
</xsd:complexType>

```

[top](#)

## Complex Type: OptionStrike

**Super-types:** [OptionNumericStrike](#) < [OptionStrike](#) (by extension)

Sub-types: None

Name	OptionStrike
Used by (from the same schema document)	Complex Type <a href="#">StrikeSpread</a>
Abstract	no
Documentation	A type for defining the strike price for an equity option. The strike price is either: (i) in respect of an index option transaction, the level of the relevant index specified or otherwise determined in the transaction; or (ii) in respect of a share option transaction, the price per share specified or otherwise determined in the transaction. This can be expressed either as a percentage of notional amount or as an absolute value.

#### XML Instance Representation

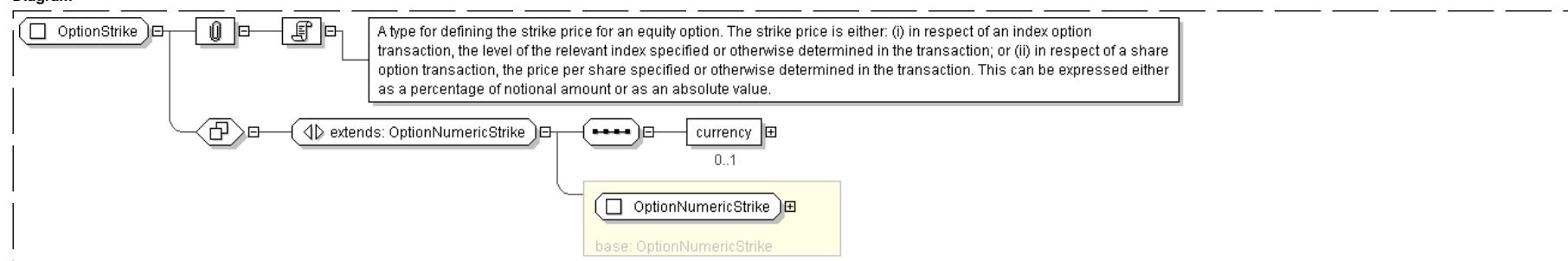
```
<...>
Start Choice [1]
  <strikePrice> xsd:decimal </strikePrice> [1]
    'The price or level at which the option has been struck.'

  <strikePercentage> xsd:decimal </strikePercentage> [1]
    'The price or level expressed as a percentage of the forward starting spot price.'

End Choice
<currency> Currency </currency> [0..1]
  'The currency in which an amount is denominated.'

</...>
```

#### Diagram



#### Schema Component Representation

```

<xsd:complexType name="OptionStrike">
  <xsd:complexContent>
    <xsd:extension base=" OptionNumericStrike ">
      <xsd:sequence>
        <xsd:element name="currency" type=" Currency " minOccurs="0" />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
  
```

top

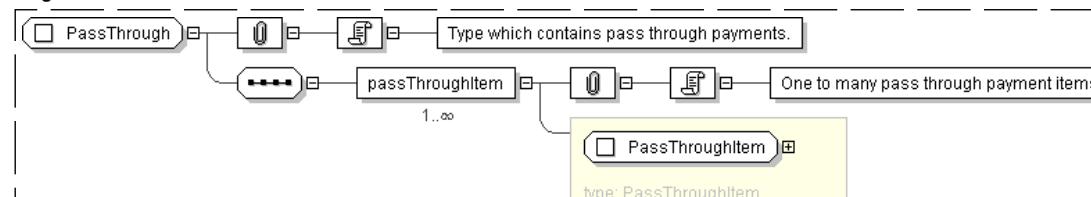
## Complex Type: PassThrough

Super-types: None  
Sub-types: None

Name	PassThrough
Used by (from the same schema document)	Model Group <a href="#">OptionFeature.model</a>
Abstract	no
Documentation	Type which contains pass through payments.

**XML Instance Representation**

```
<...>
<passThroughItem> PassThroughItem </passThroughItem> [1..*]
'One to many pass through payment items.'
</...>
```

**Diagram****Schema Component Representation**

```
<xsd:complexType name="PassThrough">
  <xsd:sequence>
    <xsd:element name="passThroughItem" type=" PassThroughItem " maxOccurs="unbounded" />
  </xsd:sequence>
</xsd:complexType>
```

[top](#)**Complex Type: PassThroughItem**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None

<b>Name</b>	PassThroughItem
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">PassThrough</a>
<b>Abstract</b>	no
<b>Documentation</b>	Type to represent a single pass through payment.

**XML Instance Representation**

```
<...>
<payerPartyReference> PartyReference </payerPartyReference> [1]
'A reference to the party responsible for making the payments defined by this structure.'

<payerAccountReference> AccountReference </payerAccountReference> [0..1]
'A reference to the account responsible for making the payments defined by this structure.'

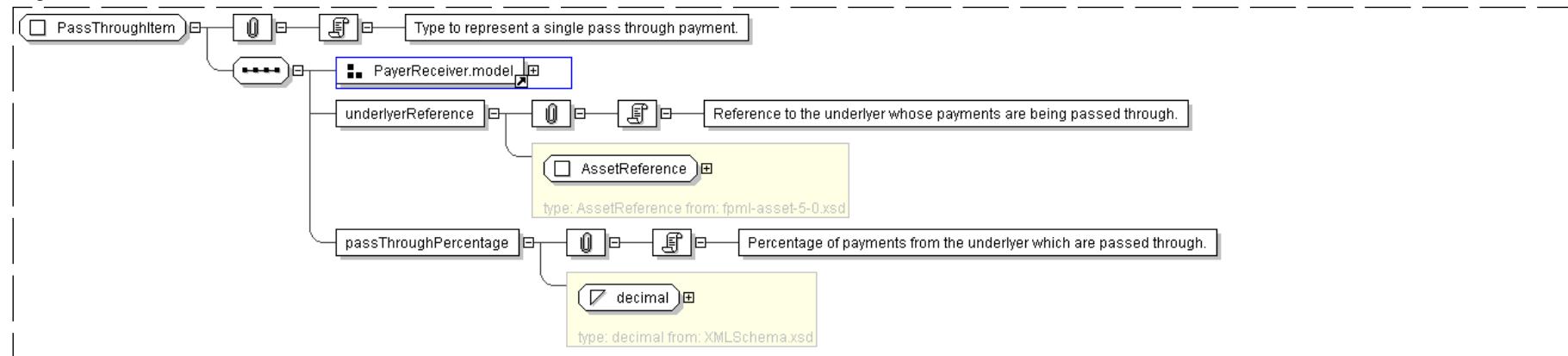
<receiverPartyReference> PartyReference </receiverPartyReference> [1]
'A reference to the party that receives the payments corresponding to this structure.'

<receiverAccountReference> AccountReference </receiverAccountReference> [0..1]
'A reference to the account that receives the payments corresponding to this structure.'

<underlyerReference> AssetReference </underlyerReference> [1]
'Reference to the underlyer whose payments are being passed through.'

<passThroughPercentage> xsd:decimal </passThroughPercentage> [1]
'Percentage of payments from the underlyer which are passed through.'

</...>
```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="PassThroughItem">
  <xsd:sequence>
    <xsd:group ref=" PayerReceiver.model " />
    <xsd:element name="underlyerReference" type=" AssetReference " />
    <xsd:element name="passThroughPercentage" type=" xsd:decimal " />
  </xsd:sequence>
</xsd:complexType>
  
```

top

**Complex Type: Premium**

Super-types:	<a href="#">SimplePayment</a> < <b>Premium</b> (by extension)
Sub-types:	None
Name	Premium
Used by (from the same schema document)	Complex Type <a href="#">OptionBaseExtended</a>
Abstract	no
Documentation	A type for defining a premium.

**XML Instance Representation**

```

<...>
<id=" xsd:ID [0..1]">
<payerPartyReference> PartyReference </payerPartyReference> [1]
  'A reference to the party responsible for making the payments defined by this structure.'
<payerAccountReference> AccountReference </payerAccountReference> [0..1]
  'A reference to the account responsible for making the payments defined by this structure.'
<receiverPartyReference> PartyReference </receiverPartyReference> [1]
  'A reference to the party that receives the payments corresponding to this structure.'
<receiverAccountReference> AccountReference </receiverAccountReference> [0..1]
  'A reference to the account that receives the payments corresponding to this structure.'
<paymentAmount> Money </paymentAmount> [1]
<paymentDate> AdjustableOrRelativeDate </paymentDate> [1]
  'The payment date. This date is subject to adjustment in accordance with any applicable business day convention.'
  
```

```

Start Group: Premium.model [0..1]
<premiumType> PremiumTypeEnum </premiumType> [0..1]
  'Forward start Premium type'

<pricePerOption> Money </pricePerOption> [0..1]
'The amount of premium to be paid expressed as a function of the number of options.'

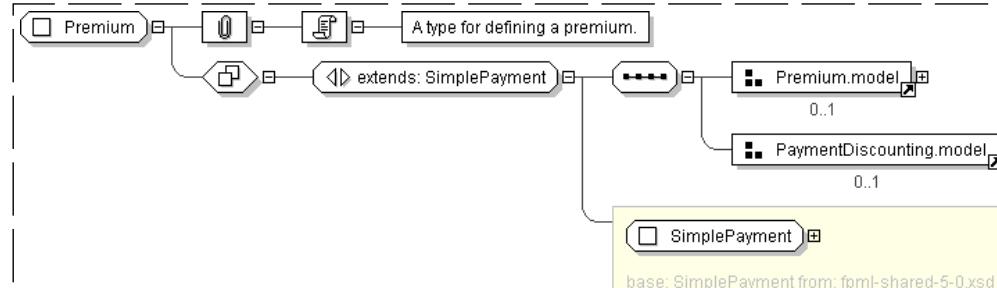
<percentageOfNotional> xsd:decimal </percentageOfNotional> [0..1]
'The amount of premium to be paid expressed as a percentage of the notional value of
the transaction. A percentage of 5% would be expressed as 0.05.'

End Group: Premium.model
Start Group: PaymentDiscounting.model [0..1]
<discountFactor> xsd:decimal </discountFactor> [0..1]
'The value representing the discount factor used to calculate the present value of the
cash flow.'

<presentValueAmount> Money </presentValueAmount> [0..1]
'The amount representing the present value of the forecast payment.'

End Group: PaymentDiscounting.model
</...>

```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="Premium">
  <xsd:complexContent>
    <xsd:extension base=" SimplePayment ">
      <xsd:sequence>
        <xsd:group ref=" Premium.model " minOccurs="0" />
        <xsd:group ref=" PaymentDiscounting.model " minOccurs="0" />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

top

**Complex Type: PubliclyAvailableInformation**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None

<b>Name</b>	PubliclyAvailableInformation
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">CreditEventNotice</a>
<b>Abstract</b>	no

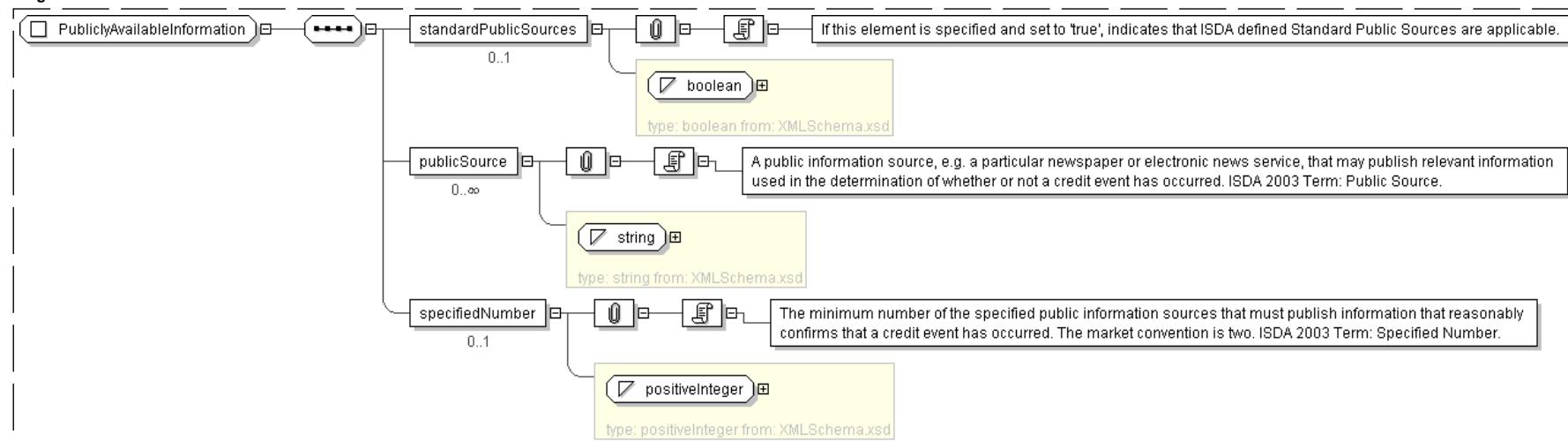
**XML Instance Representation**

```
<...>
<standardPublicSources> xsd:boolean </standardPublicSources> [0..1]
'If this element is specified and set to \'true\', indicates that ISDA defined Standard Public Sources are applicable.'

<publicSource> xsd:string </publicSource> [0..*]
'A public information source, e.g. a particular newspaper or electronic news service, that may publish relevant information used in the determination of whether or not a credit event has occurred. ISDA 2003 Term: Public Source.'

<specifiedNumber> xsd:positiveInteger </specifiedNumber> [0..1]
'The minimum number of the specified public information sources that must publish information that reasonably confirms that a credit event has occurred. The market convention is two. ISDA 2003 Term: Specified Number.'

</...>
```

**Diagram****Schema Component Representation**

```
<xsd:complexType name="PubliclyAvailableInformation">
  <xsd:sequence>
    <xsd:element name="standardPublicSources" type="xsd:boolean" minOccurs="0"/>
    <xsd:element name="publicSource" type="xsd:string" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element name="specifiedNumber" type="xsd:positiveInteger" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>
```

top

**Complex Type: Quanto**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None

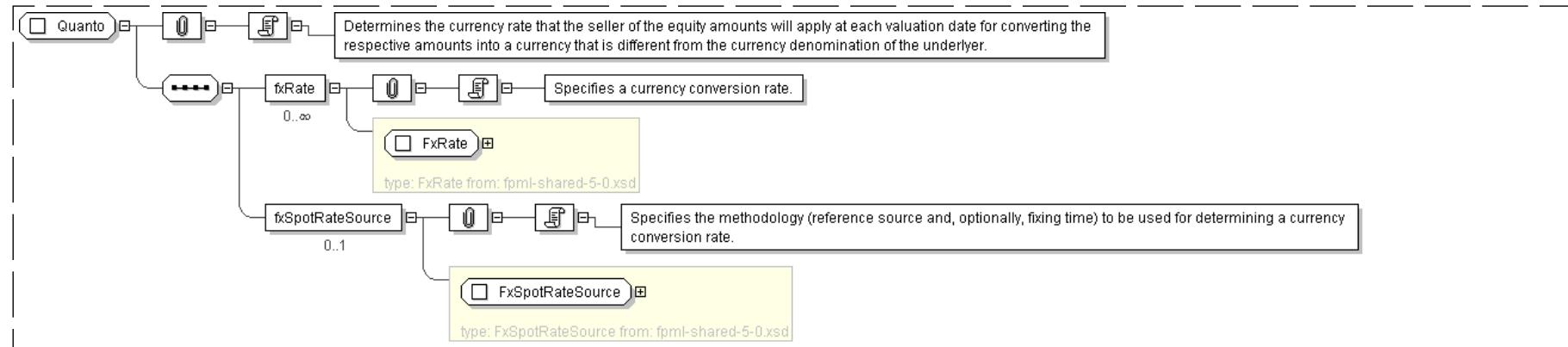
<b>Name</b>	Quanto
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">FxFeature</a>
<b>Abstract</b>	no
<b>Documentation</b>	Determines the currency rate that the seller of the equity amounts will apply at each valuation date for converting the respective amounts into a currency that is different from the currency denomination of the underlier.

**XML Instance Representation**

```
<...>
<fxRate> FxRate </fxRate> [0..*]
'Specifies a currency conversion rate.'

<fxSpotRateSource> FxSpotRateSource </fxSpotRateSource> [0..1]
'Specifies the methodology (reference source and, optionally, fixing time) to be used
for determining a currency conversion rate.

</...>
```

**Diagram****Schema Component Representation**

```
<xsd:complexType name="Quanto">
  <xsd:sequence>
    <xsd:element name="fxRate" type="FxRate" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element name="fxSpotRateSource" type="FxSpotRateSource" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>
```

top

**Complex Type: Restructuring**

Super-types:	None
Sub-types:	None
Name	Restructuring
Used by (from the same schema document)	Complex Type <a href="#">CreditEvents</a>
Abstract	no

**XML Instance Representation**

```
<...>
<applicable> xsd:boolean </applicable> [1]
'Indicates whether the restructuring provision is applicable.'

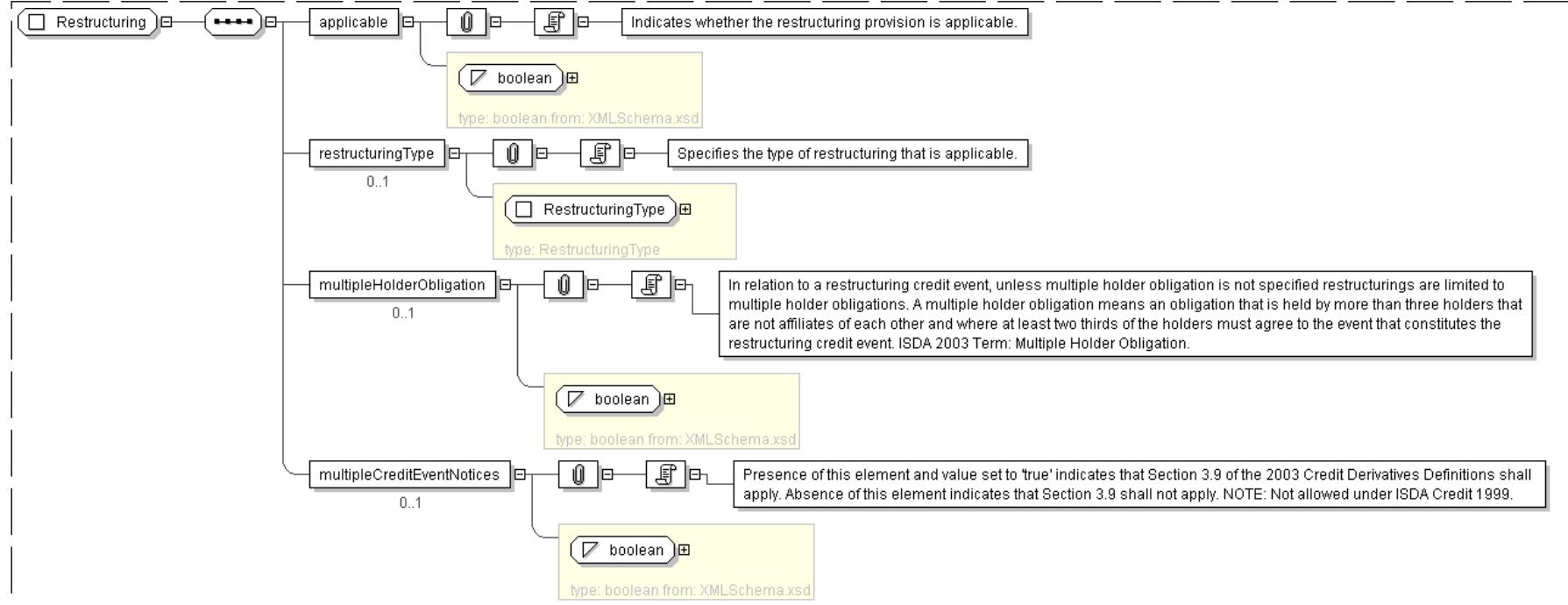
<restructuringType> RestructuringType </restructuringType> [0..1]
'Specifies the type of restructuring that is applicable.'

<multipleHolderObligation> xsd:boolean </multipleHolderObligation> [0..1]
'In relation to a restructuring credit event, unless multiple holder obligation is
not specified restructurings are limited to multiple holder obligations. A multiple
```

holder obligation means an obligation that is held by more than three holders that are not affiliates of each other and where at least two thirds of the holders must agree to the event that constitutes the restructuring credit event. ISDA 2003 Term: Multiple Holder Obligation.'

<multipleCreditEventNotices> xsd:boolean </multipleCreditEventNotices> [0..1]  
 'Presence of this element and value set to \'true\' indicates that Section 3.9 of the 2003 Credit Derivatives Definitions shall apply. Absence of this element indicates that Section 3.9 shall not apply. NOTE: Not allowed under ISDA Credit 1999.'

&lt;...&gt;

**Diagram****Schema Component Representation**

```

<xsd:complexType name="Restructuring">
  <xsd:sequence>
    <xsd:element name="applicable" type="xsd:boolean" />
    <xsd:element name="restructuringType" type="RestructuringType" minOccurs="0"/>
    <xsd:element name="multipleHolderObligation" type="xsd:boolean" minOccurs="0"/>
    <xsd:element name="multipleCreditEventNotices" type="xsd:boolean" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>
  
```

top

**Complex Type: RestructuringType**

Super-types:  
Sub-types:

[Scheme](#) < **RestructuringType** (by extension)  
None

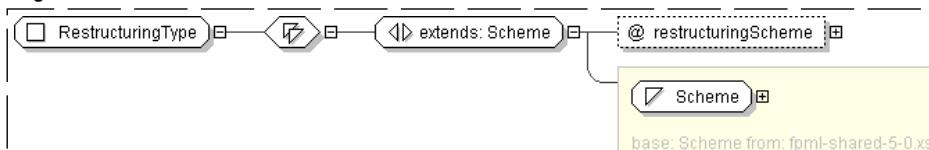
Name	RestructuringType
Used by (from the same schema document)	Complex Type <a href="#">Restructuring</a>

**Abstract**

no

**XML Instance Representation**

```
<...>
  restructuringScheme=" xsd:anyURI [0..1]">
  Scheme
</...>
```

**Diagram****Schema Component Representation**

```
<xsd:complexType name="RestructuringType">
  <xsd:simpleContent>
    <xsd:extension base=" Scheme ">
      <xsd:attribute name="restructuringScheme" type=" xsd:anyURI " default="http://www.fpml.org/coding-scheme/restructuring"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
```

top

**Complex Type: StrategyFeature**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None

<b>Name</b>	StrategyFeature
<b>Used by (from the same schema document)</b>	Model Group <a href="#">OptionBaseFeature.model</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type for defining equity option simple strike or calendar spread strategy features.

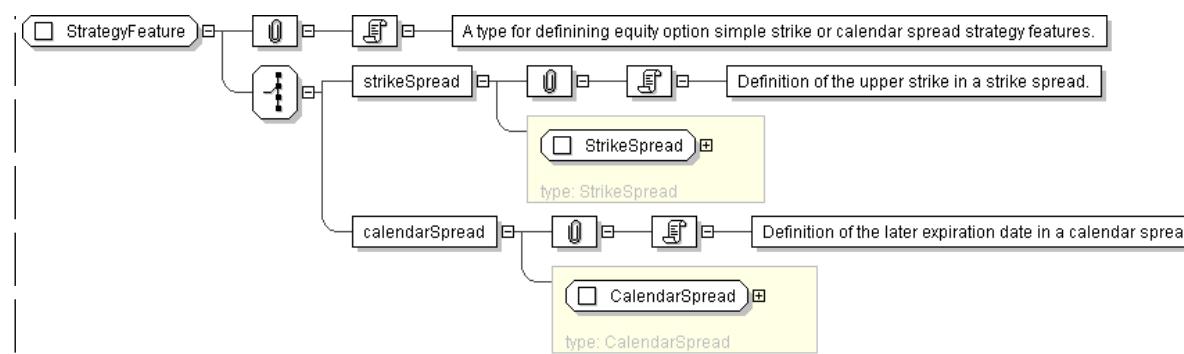
**XML Instance Representation**

```
<...>
  Start Choice [1]
    <strikeSpread> StrikeSpread </strikeSpread> [1]
    'Definition of the upper strike in a strike spread.'

    <calendarSpread> CalendarSpread </calendarSpread> [1]
    'Definition of the later expiration date in a calendar spread.'

  End Choice
</...>
```

**Diagram**

**Schema Component Representation**

```

<xsd:complexType name="StrategyFeature">
  <xsd:choice>
    <xsd:element name="strikeSpread" type="StrikeSpread" />
    <xsd:element name="calendarSpread" type="CalendarSpread" />
  </xsd:choice>
</xsd:complexType>
  
```

[top](#)**Complex Type: StrikeSpread**

Super-types:	None
Sub-types:	None

Name	StrikeSpread
Used by (from the same schema document)	Complex Type <a href="#">StrategyFeature</a>
Abstract	no
Documentation	A type for defining a strike spread feature.

**XML Instance Representation**

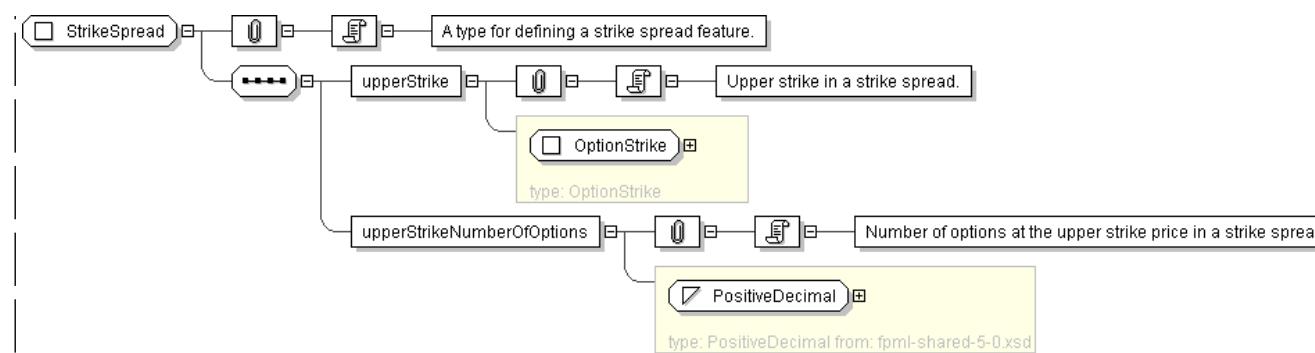
```

<...>
<upperStrike> OptionStrike </upperStrike> [1]
'Upper strike in a strike spread.'

<upperStrikeNumberOfOptions> PositiveDecimal </upperStrikeNumberOfOptions> [1]
'Number of options at the upper strike price in a strike spread.'

</...>
  
```

**Diagram**

**Schema Component Representation**

```

<xsd:complexType name="StrikeSpread">
  <xsd:sequence>
    <xsd:element name="upperStrike" type=" OptionStrike " />
    <xsd:element name="upperStrikeNumberOfOptions" type=" PositiveDecimal " />
  </xsd:sequence>
</xsd:complexType>
  
```

top

**Complex Type: Trigger**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None

<b>Name</b>	Trigger
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">TriggerEvent</a>
<b>Abstract</b>	no
<b>Documentation</b>	Trigger point at which feature is effective.

**XML Instance Representation**

```

<...>
Start Choice [1]
  <level> xsd:decimal </level> [1]
  'The trigger level.'
  <levelPercentage> xsd:decimal </levelPercentage> [1]
  'The trigger level percentage.'

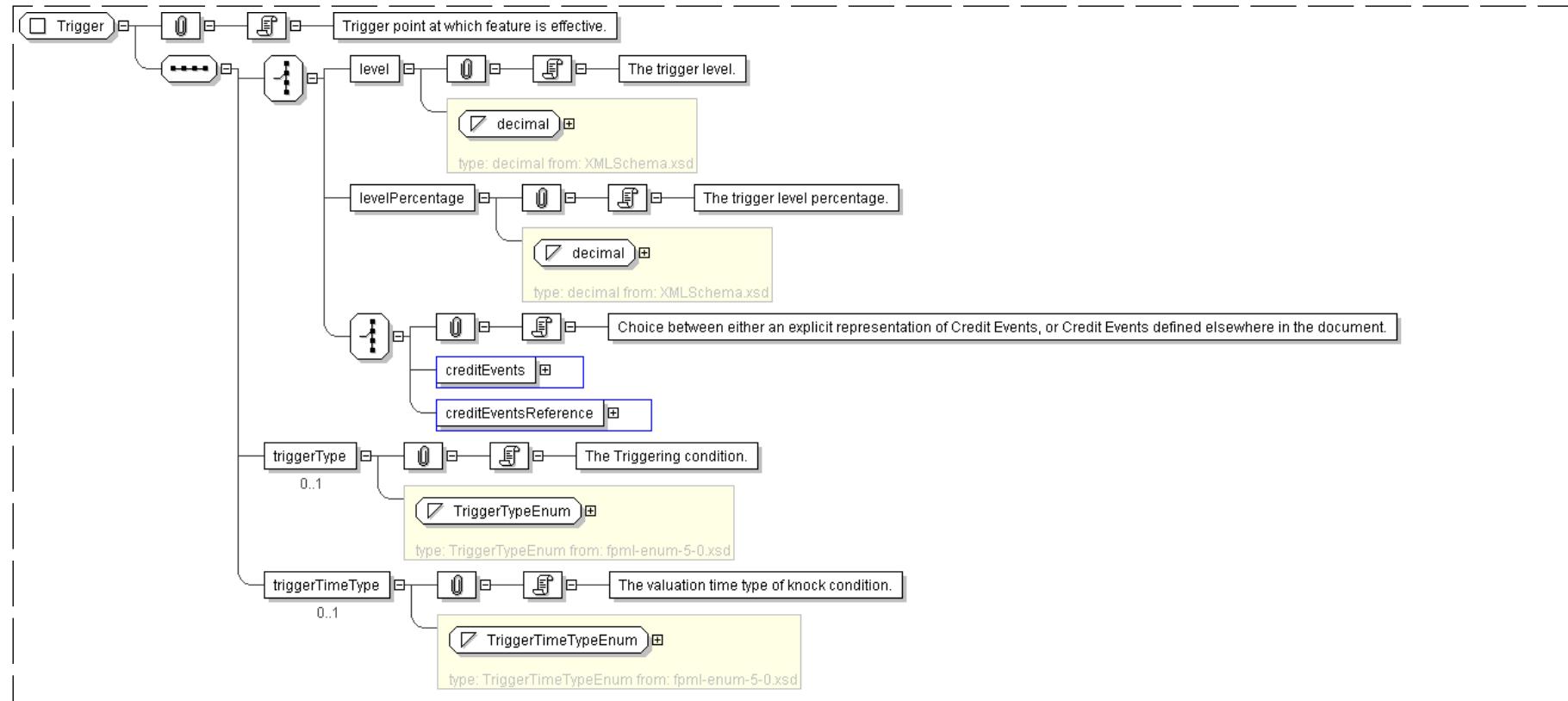
Start Choice [1]
  'Choice between either an explicit representation of Credit Events, or Credit Events
defined elsewhere in the document.'

    <creditEvents> CreditEvents </creditEvents> [1]
    <creditEventsReference> CreditEventsReference </creditEventsReference> [1]

End Choice
End Choice
<triggerType> TriggerTypeEnum </triggerType> [0..1]
  'The Triggering condition.'

  <triggerTimeType> TriggerTimeTypeEnum </triggerTimeType> [0..1]
  'The valuation time type of knock condition.'

</...>
  
```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="Trigger">
  <xsd:sequence>
    <xsd:choice>
      <xsd:element name="level" type="xsd:decimal"/>
      <xsd:element name="levelPercentage" type="xsd:decimal"/>
      <xsd:choice>
        <xsd:element name="creditEvents" type="CreditEvents"/>
        <xsd:element name="creditEventsReference" type="CreditEventsReference"/>
      </xsd:choice>
    </xsd:choice>
    <xsd:element name="triggerType" type="TriggerTypeEnum" minOccurs="0" maxOccurs="1"/>
    <xsd:element name="triggerTimeType" type="TriggerTimeTypeEnum" minOccurs="0" maxOccurs="1"/>
  </xsd:sequence>
</xsd:complexType>
  
```

top

**Complex Type: TriggerEvent**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None
<b>Name</b>	TriggerEvent
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">Barrier</a> , Complex Type <a href="#">Barrier</a> , Complex Type <a href="#">Knock</a> , Complex Type <a href="#">Knock</a>
<b>Abstract</b>	no

**Documentation**

Observation point for trigger.

**XML Instance Representation**

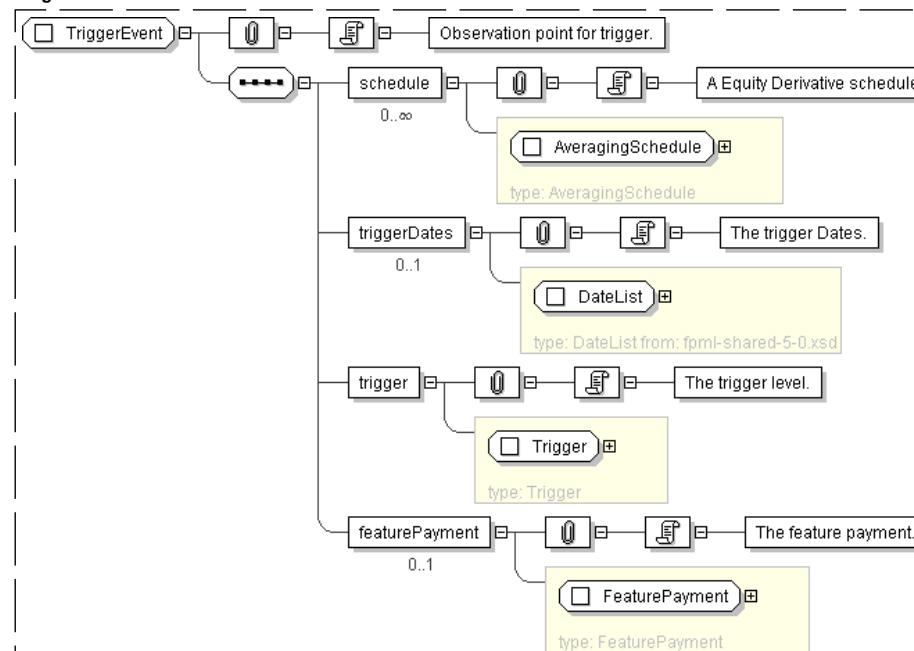
```
<...>
<schedule> AveragingSchedule </schedule> [0..*]
'A Equity Derivative schedule.'

<triggerDates> DateList </triggerDates> [0..1]
'The trigger Dates.'

<trigger> Trigger </trigger> [1]
'The trigger level.'

<featurePayment> FeaturePayment </featurePayment> [0..1]
'The feature payment.'

</...>
```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="TriggerEvent">
  <xsd:sequence>
    <xsd:element name="schedule" type=" AveragingSchedule " minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element name="triggerDates" type=" DateList " minOccurs="0"/>
    <xsd:element name="trigger" type=" Trigger "/>
    <xsd:element name="featurePayment" type=" FeaturePayment " minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>

```

<b>Super-types:</b>	None
<b>Sub-types:</b>	None

<b>Name</b>	WeightedAveragingObservation
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">AveragingObservationList</a>
<b>Abstract</b>	no
<b>Documentation</b>	A single weighted averaging observation.

**XML Instance Representation**

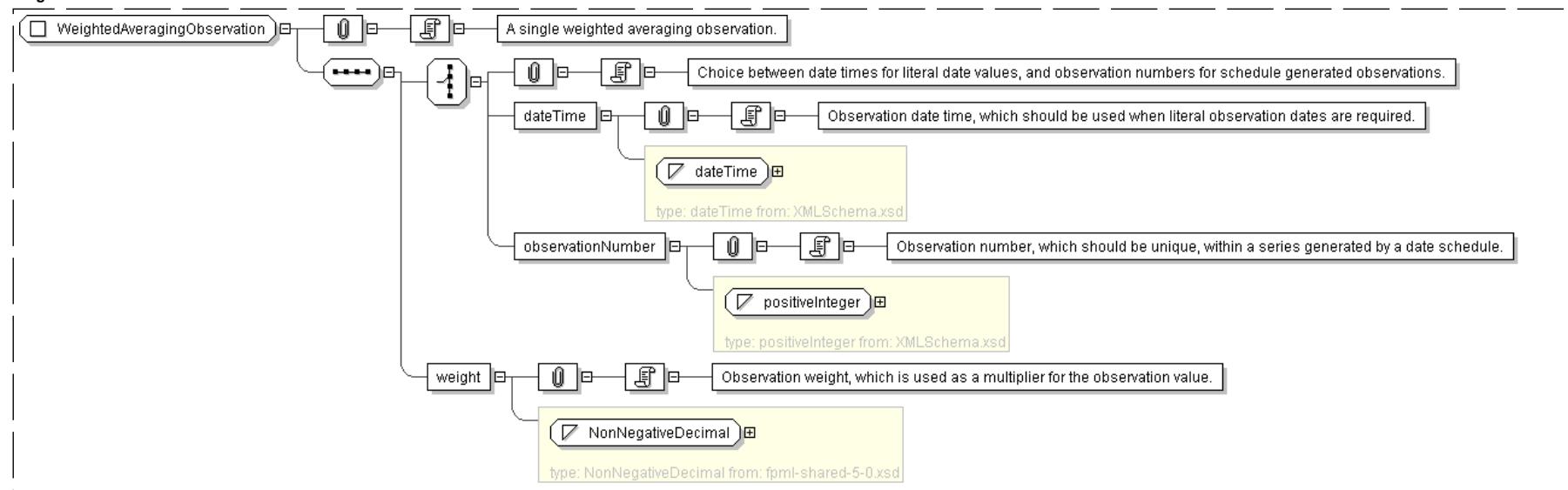
```
<...>
Start Choice [1]
'Choice between date times for literal date values, and observation numbers for
schedule generated observations.'

<dateTime> xsd:dateTime </dateTime> [1]
'Observation date time, which should be used when literal observation dates are required.'

<observationNumber> xsd:positiveInteger </observationNumber> [1]
'Observation number, which should be unique, within a series generated by a date schedule.'

End Choice
<weight> NonNegativeDecimal </weight> [1]
'Observation weight, which is used as a multiplier for the observation value.'

</...>
```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="WeightedAveragingObservation">
  <xsd:sequence>
    <xsd:choice>
      <xsd:element name="dateTime" type="xsd:dateTime" />
      <xsd:element name="observationNumber" type="xsd:positiveInteger" />
    </xsd:choice>
    <xsd:element name="weight" type="NonNegativeDecimal" />
  </xsd:sequence>
</xsd:complexType>

```

**Model Group: OptionBaseFeature.model**

<b>Name</b>	OptionBaseFeature.model
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">OptionFeature</a>
<b>Documentation</b>	A model group containing Option Base Feature Elements.

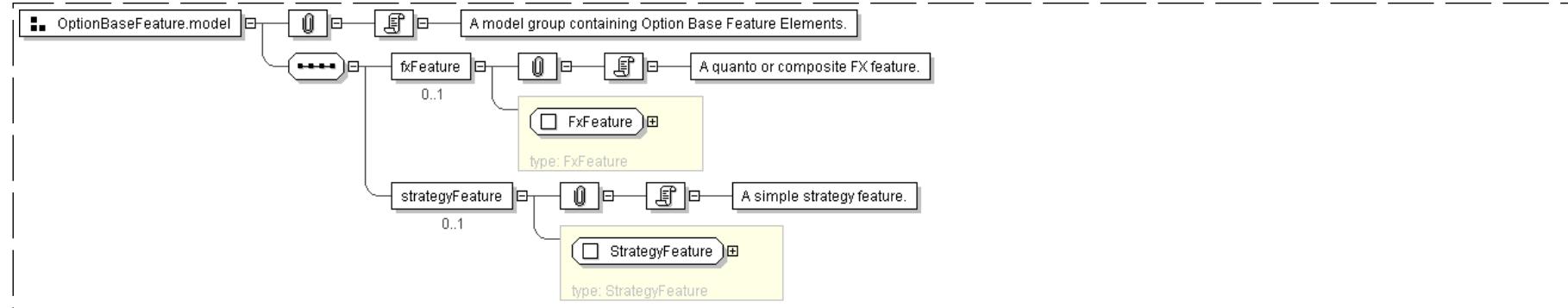
**XML Instance Representation**

```
<fxFeature> FxFeature </fxFeature> [0..1]
```

'A quanto or composite FX feature.'

```
<strategyFeature> StrategyFeature </strategyFeature> [0..1]
```

'A simple strategy feature.'

**Diagram****Schema Component Representation**

```

<xsd:group name="OptionBaseFeature.model">
  <xsd:sequence>
    <xsd:element name="fxFeature" type="FxFeature" minOccurs="0"/>
    <xsd:element name="strategyFeature" type="StrategyFeature" minOccurs="0"/>
  </xsd:sequence>
</xsd:group>
  
```

**Model Group: OptionDenomination.model**

<b>Name</b>	OptionDenomination.model
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">OptionBaseExtended</a>
<b>Documentation</b>	A model group containing the option denomination components.

**XML Instance Representation**

```
<optionEntitlement> PositiveDecimal </optionEntitlement> [1]
```

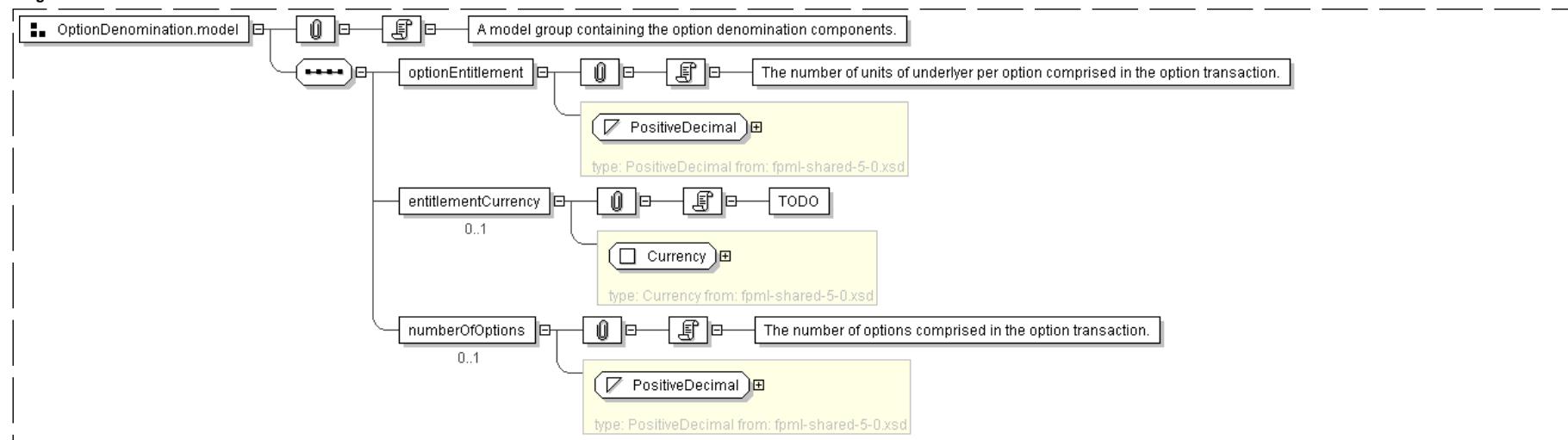
'The number of units of underlyer per option comprised in the option transaction.'

```
<entitlementCurrency> Currency </entitlementCurrency> [0..1]
```

'TODO'

```
<numberOfOptions> PositiveDecimal </numberOfOptions> [0..1]
```

'The number of options comprised in the option transaction.'

**Diagram****Schema Component Representation**

```

<xsd:group name="OptionDenomination.model">
  <xsd:sequence>
    <xsd:element name="optionEntitlement" type="PositiveDecimal" />
    <xsd:element name="entitlementCurrency" type="Currency" minOccurs="0"/>
    <xsd:element name="numberOfOptions" type="PositiveDecimal" minOccurs="0"/>
  </xsd:sequence>
</xsd:group>
  
```

top

**Model Group: OptionFeature.model**

<b>Name</b>	OptionFeature.model
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">OptionFeature</a>
<b>Documentation</b>	A model group containing Option Base Feature Elements.

**XML Instance Representation**

```

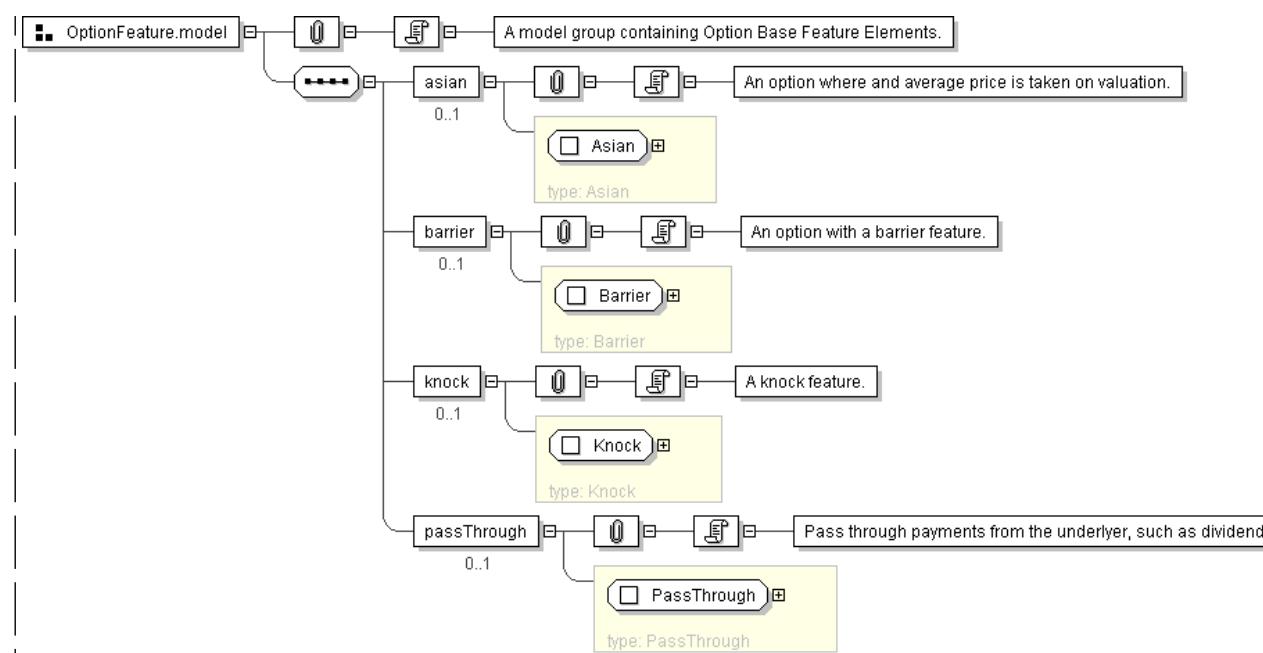
<asian> Asian </asian> [0..1]
'An option where an average price is taken on valuation.'

<barrier> Barrier </barrier> [0..1]
'An option with a barrier feature.'

<knock> Knock </knock> [0..1]
'A knock feature.'

<passThrough> PassThrough </passThrough> [0..1]
'Pass through payments from the underlyer, such as dividends.'
  
```

**Diagram**

**Schema Component Representation**

```

<xsd:group name="OptionFeature.model">
  <xsd:sequence>
    <xsd:element name="asian" type="Asian" minOccurs="0"/>
    <xsd:element name="barrier" type="Barrier" minOccurs="0"/>
    <xsd:element name="knock" type="Knock" minOccurs="0"/>
    <xsd:element name="passThrough" type="PassThrough" minOccurs="0"/>
  </xsd:sequence>
</xsd:group>
  
```

[top](#)**Model Group: OptionSettlement.model**

<b>Name</b>	OptionSettlement.model
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">OptionBaseExtended</a>
<b>Documentation</b>	A group which has Option Settlement elements.

**XML Instance Representation**

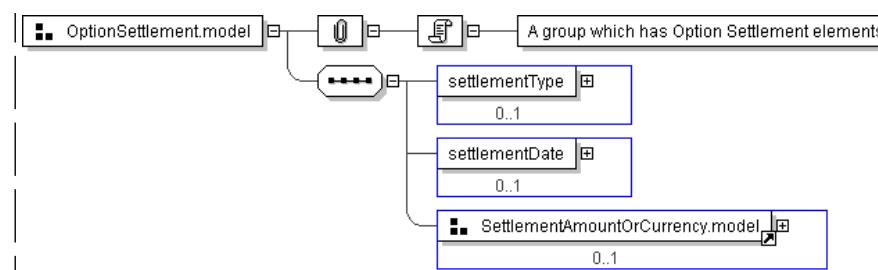
```

<settlementType> SettlementTypeEnum </settlementType> [0..1]
<settlementDate> AdjustableOrRelativeDate </settlementDate> [0..1]
Start Group: SettlementAmountOrCurrency.model [0..1]
Start Choice [1]
  <settlementAmount> Money </settlementAmount> [1]
  'Settlement Amount'

  <settlementCurrency> Currency </settlementCurrency> [1]
  'Settlement Currency for use where the Settlement Amount cannot be known in advance'

End Choice
End Group: SettlementAmountOrCurrency.model
  
```

**Diagram**

**Schema Component Representation**

```

<xsd:group name="OptionSettlement.model">
  <xsd:sequence>
    <!--Is this an optional or required elemnt ?-->
    <xsd:element name="settlementType" type=" SettlementTypeEnum " minOccurs="0 " />
    <xsd:element name="settlementDate" type=" AdjustableOrRelativeDate " minOccurs="0 " />
    <xsd:group ref=" SettlementAmountOrCurrency.model " minOccurs="0 " />
  </xsd:sequence>
</xsd:group>
  
```

[top](#)**Legend**

**Complex Type:**  
Schema Component Type

**AusAddress**  
Schema Component Name

**Super-types:** [Address](#) < AusAddress (by extension)  
**Sub-types:** • [OLDAddress](#) (by restriction)

If this schema component is a type definition, its type hierarchy is shown in a gray-bordered box.

Name	AusAddress
Abstract	no

The table above displays the properties of this schema component.

**XML Instance Representation**

```

<... country="Australia" >
<unitNo> string </unitNo> [0..1]
<houseNo> string </houseNo> [1]
<street> string </street> [1]
Start Choice [1]
<city> string </city> [1]
<town> string </town> [1]
End Choice
<state> AusStates </state> [1]
<postcode> string <><pattern = "[1-9][0-9]{3}></postcode> [1]
</...>
  
```

The XML Instance Representation table above shows the schema component's content as an XML instance.

- The minimum and maximum occurrence of elements and attributes are provided in square brackets, e.g. [0..1].
- Model group information are shown in gray, e.g. Start Choice ... End Choice.
- For type derivations, the elements and attributes that have been added to or changed from the base type's content are shown in **bold**.
- If an element/attribute has a fixed value, the fixed value is shown in green, e.g. country="Australia".
- Otherwise, the type of the element/attribute is displayed.
  - If the element/attribute's type is in the schema, a link is provided to it.
  - For local simple type definitions, the constraints are displayed in angle brackets, e.g. <><pattern = "[1-9][0-9]{3}></pattern>.

**Schema Component Representation**

```
<complexType name="AusAddress">
<complexContent>
<extension base=" Address ">
<sequence>
<element name="state" type=" AusStates " />
<element name="postcode">
<simpleType>
<restriction base=" string "
<pattern value="[1-9][0-9]{3}" />
</restriction>
</simpleType>
</element>
</sequence>
<attribute name="country" type=" string " fixed="Australia" />
</extension>
</complexContent>
</complexType>
```

The Schema Component Representation table above displays the underlying XML representation of the schema component. (Annotations are not shown.)

[top](#)
**Glossary**

**Abstract** (Applies to complex type definitions and element declarations). An abstract element or complex type cannot be used to validate an element instance. If there is a reference to an abstract element, only element declarations that can substitute the abstract element can be used to validate the instance. For references to abstract type definitions, only derived types can be used.

**All Model Group** Child elements can be provided *in any order* in instances. See: <http://www.w3.org/TR/xmlschema-1/#element-all>.

**Choice Model Group** *Only one* from the list of child elements and model groups can be provided in instances. See: <http://www.w3.org/TR/xmlschema-1/#element-choice>.

**Collapse Whitespace Policy** Replace tab, line feed, and carriage return characters with space character (Unicode character 32). Then, collapse contiguous sequences of space characters into single space character, and remove leading and trailing space characters.

**Disallowed Substitutions** (Applies to element declarations). If *substitution* is specified, then *substitution group* members cannot be used in place of the given element declaration to validate element instances. If *derivation methods*, e.g. extension, restriction, are specified, then the given element declaration will not validate element instances that have types derived from the element declaration's type using the specified derivation methods. Normally, element instances can override their declaration's type by specifying an *xsi:type* attribute.

**Key Constraint** Like [Uniqueness Constraint](#), but additionally requires that the specified value(s) must be provided. See: [http://www.w3.org/TR/xmlschema-1/#clentity-constraint\\_Definitions](http://www.w3.org/TR/xmlschema-1/#clentity-constraint_Definitions).

**Key Reference Constraint** Ensures that the specified value(s) must match value(s) from a [Key Constraint](#) or [Uniqueness Constraint](#). See: [http://www.w3.org/TR/xmlschema-1/#clentity-constraint\\_Definitions](http://www.w3.org/TR/xmlschema-1/#clentity-constraint_Definitions).

**Model Group** Groups together element content, specifying the order in which the element content can occur and the number of times the group of element content may be repeated. See: [http://www.w3.org/TR/xmlschema-1/#Model\\_Groups](http://www.w3.org/TR/xmlschema-1/#Model_Groups).

**Nullable** (Applies to element declarations). If an element declaration is nullable, instances can use the *xsi:nil* attribute. The *xsi:nil* attribute is the boolean attribute, *nil*, from the <http://www.w3.org/2001/XMLSchema-instance> namespace. If an element instance has an *xsi:nil* attribute set to true, it can be left empty, even though its element declaration may have required content.

**Notation** A notation is used to identify the format of a piece of data. Values of elements and attributes that are of type, NOTATION, must come from the names of declared notations. See: [http://www.w3.org/TR/xmlschema-1/#cNotation\\_Declarations](http://www.w3.org/TR/xmlschema-1/#cNotation_Declarations).

**Preserve Whitespace Policy** Preserve whitespaces exactly as they appear in instances.

**Prohibited Derivations** (Applies to type definitions). Derivation methods that cannot be used to create sub-types from a given type definition.

**Prohibited Substitutions** (Applies to complex type definitions). Prevents sub-types that have been derived using the specified derivation methods

from validating element instances in place of the given type definition.

**Replace Whitespace Policy** Replace tab, line feed, and carriage return characters with space character (Unicode character 32).

**Sequence Model Group** Child elements and model groups must be provided *in the specified order* in instances. See: <http://www.w3.org/TR/xmlschema-1/#element-sequence>.

**Substitution Group** Elements that are *members* of a substitution group can be used wherever the *head* element of the substitution group is referenced.

**Substitution Group Exclusions** (Applies to element declarations). Prohibits element declarations from nominating themselves as being able to substitute a given element declaration, if they have types that are derived from the original element's type using the specified derivation methods.

**Target Namespace** The target namespace identifies the namespace that components in this schema belongs to. If no target namespace is provided, then the schema components do not belong to any namespace.

**Uniqueness Constraint** Ensures uniqueness of an element/attribute value, or a combination of values, within a specified scope. See: [http://www.w3.org/TR/xmlschema-1/#clentity-constraint\\_Definitions](http://www.w3.org/TR/xmlschema-1/#clentity-constraint_Definitions).

[top](#)

---

Generated by [eXoYgen!> XML Editor](#) using a modified version of [xs3p](#) that adds schema diagrams and chunking support.

# XML Schema Documentation

## Table of Contents

- [Schema Document Properties](#)
- [Global Declarations](#)
  - [Element: equitySwapTransactionSupplement](#)
- [Global Definitions](#)
  - [Complex Type: EquitySwapTransactionSupplement](#)
- [Legend](#)
- [Glossary](#)

[top](#)

## Schema Document Properties

<b>Target Namespace</b>	<a href="http://www.fpml.org/FpML-5/confirmation">http://www.fpml.org/FpML-5/confirmation</a>
<b>Version</b>	\$Revision: 918 \$
<b>Element and Attribute Namespaces</b>	<ul style="list-style-type: none"> <li>• Global element and attribute declarations belong to this schema's target namespace.</li> <li>• By default, local element declarations belong to this schema's target namespace.</li> <li>• By default, local attribute declarations have no namespace.</li> </ul>
<b>Schema Composition</b>	<ul style="list-style-type: none"> <li>• This schema includes components from the following schema document(s):           <ul style="list-style-type: none"> <li>◦ <a href="#">fpml-eq-shared-5-0.xsd</a></li> </ul> </li> </ul>

## Declared Namespaces

Prefix	Namespace
Default namespace	<a href="http://www.fpml.org/FpML-5/confirmation">http://www.fpml.org/FpML-5/confirmation</a>
xml	http://www.w3.org/XML/1998/namespace
fpml-annotation	http://www.fpml.org/annotation
xsd	http://www.w3.org/2001/XMLSchema
ecore	http://www.eclipse.org/emf/2002/Ecore

## Schema Component Representation

```

<xsd:schema targetNamespace="http://www.fpml.org/FpML-5/confirmation"
  documentRoot="FpML" nsPrefix="conf" package="org.fpml.confirmation" version="$Revision: 918
$" elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xsd:include schemaLocation="fpml-eq-shared-5-0.xsd"/>
  ...
</xsd:schema>

```

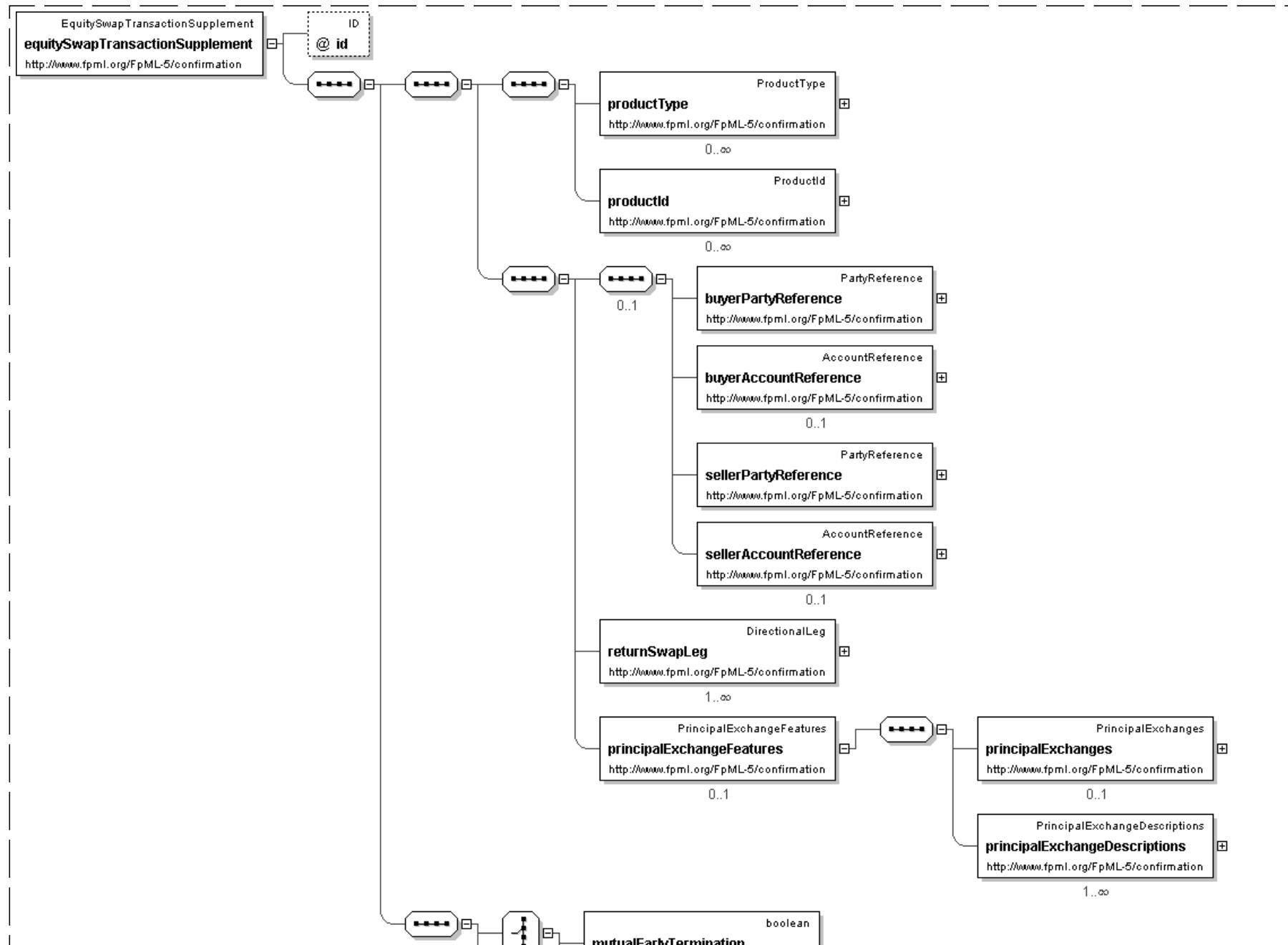
[top](#)

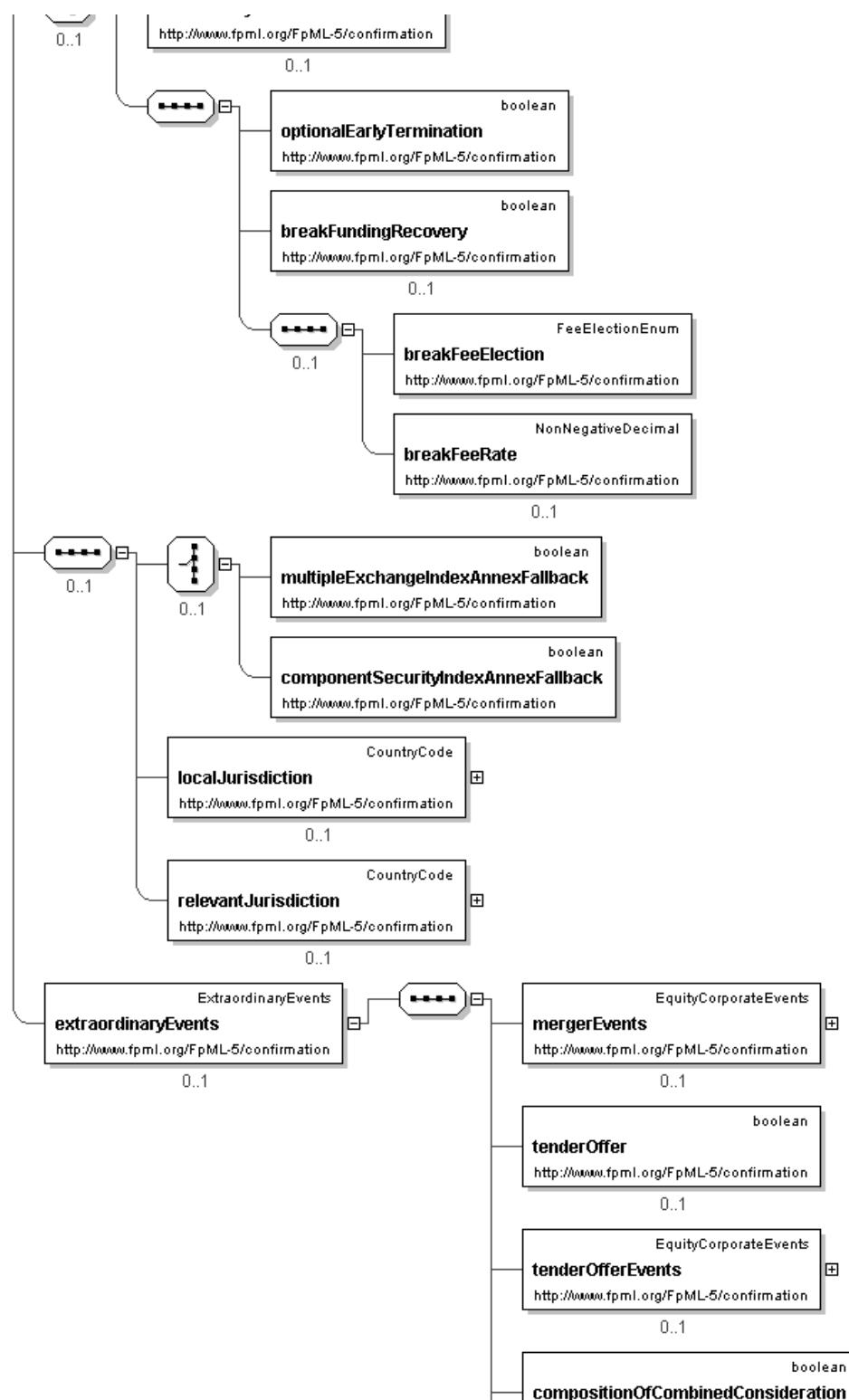
## Global Declarations

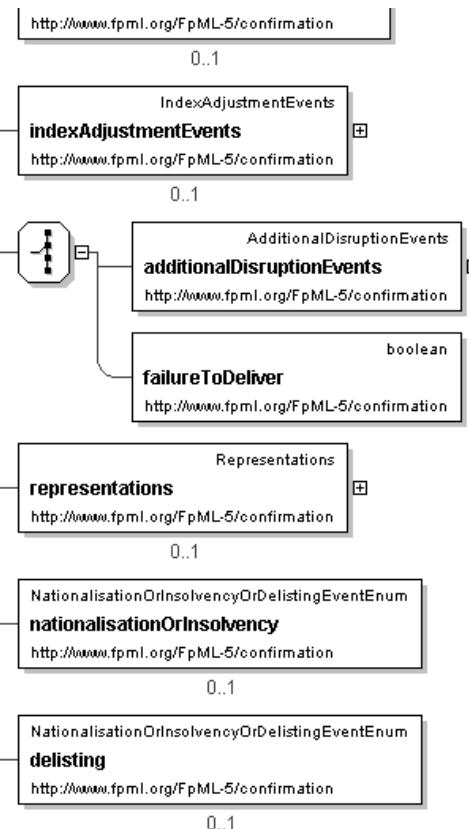
### Element: equitySwapTransactionSupplement

- This element can be used wherever the following element is referenced:
  - [product](#)

<b>Name</b>	equitySwapTransactionSupplement
<b>Type</b>	<a href="#">EquitySwapTransactionSupplement</a>
<b>Nillable</b>	no
<b>Abstract</b>	no
<b>Documentation</b>	Specifies the structure of the equity swap transaction supplement.

**Logical Diagram**





#### XML Instance Representation

```

<equitySwapTransactionSupplement
  id=" xsd:ID [0..1]">
  <productType> ProductType </productType> [0..*]
  'A classification of the type of product. FpML defines a simple product categorization using
  a coding scheme.'
  
```

```

  <productId> ProductId </productId> [0..*]
  'A product 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.'
  
```

Start Group: [BuyerSeller.model](#) [0..1]

'BuyerSeller.model has been included as an optional child of ReturnSwapBase to support the situation where an implementor wishes to indicate who has manufactured the Swap through representing them as the Seller. It may be removed in future major revisions.'

```

  <buyerPartyReference> PartyReference </buyerPartyReference> [1]
  'A reference to the party that buys this instrument, ie. pays for this instrument and
  receives the rights defined by it. See 2000 ISDA definitions Article 11.1 (b). In the case
  of FRAs this the fixed rate payer.'
  
```

```

  <buyerAccountReference> AccountReference </buyerAccountReference> [0..1]
  
```

'A reference to the account that buys this instrument.'

<sellerPartyReference> PartyReference </sellerPartyReference> [1]

'A reference to the party that sells ("writes") this instrument, i.e. that grants the rights defined by this instrument and in return receives a payment for it. See 2000 ISDA definitions Article 11.1 (a). In the case of FRAs this is the floating rate payer.'

<sellerAccountReference> AccountReference </sellerAccountReference> [0..1]

'A reference to the account that sells this instrument.'

End Group: BuyerSeller.model

<returnSwapLeg> ... </returnSwapLeg> [1..\*]

<principalExchangeFeatures> PrincipalExchangeFeatures </principalExchangeFeatures> [0..1]

'This is used to document a Fully Funded Return Swap.'

Start Group: MutualOrOptionalEarlyTermination.model [0..1]

Start Choice [1]

<mutualEarlyTermination> xsd:boolean </mutualEarlyTermination> [0..1]

'Used for specifying whether the Mutual Early Termination Right that is detailed in the Master Confirmation will apply.'

<optionalEarlyTermination> xsd:boolean </optionalEarlyTermination> [1]

'A Boolean element used for specifying whether the Optional Early Termination clause detailed in the agreement will apply.'

<breakFundingRecovery> xsd:boolean </breakFundingRecovery> [0..1]

'A Boolean element used for specifying whether the Break Funding Recovery detailed in the agreement will apply.'

Start Sequence [0..1]

<breakFeeElection> FeeElectionEnum </breakFeeElection> [1]

'Defines the fee type.'

<breakFeeRate> NonNegativeDecimal </breakFeeRate> [0..1]

End Sequence

End Choice

End Group: MutualOrOptionalEarlyTermination.model

Start Group: EquityUnderlyerProvisions.model [0..1]

Start Group: IndexAnnexFallback.model [0..1]

Start Choice [1]

<multipleExchangeIndexAnnexFallback> xsd:boolean </multipleExchangeIndexAnnexFallback> [1]

'For an index option transaction, a flag to indicate whether a relevant Multiple Exchange Index Annex is applicable to the transaction. This annex defines additional provisions which are applicable where an index is comprised of component securities that are traded on multiple exchanges.'

<componentSecurityIndexAnnexFallback> xsd:boolean </componentSecurityIndexAnnexFallback> [1]

'For an index option transaction, a flag to indicate whether a relevant Component Security Index Annex is applicable to the transaction.'

End Choice

End Group: IndexAnnexFallback.model

<localJurisdiction> CountryCode </localJurisdiction> [0..1]

'Local Jurisdiction is a term used in the AEJ Master Confirmation, which is used to determine local taxes, which shall mean taxes, duties, and similar charges imposed by the taxing authority of the Local Jurisdiction If this element is not present'

*Local Jurisdiction is Not Applicable.'*

<relevantJurisdiction> CountryCode </relevantJurisdiction> [0..1]

'Relevant Jurisdiction is a term used in the AEJ Master Confirmation, which is used to determine local taxes, which shall mean taxes, duties and similar charges that would be imposed by the taxing authority of the Country of Underlyer on a Hypothetical Broker Dealer assuming the Applicable Hedge Positions are held by its office in the Relevant Jurisdiction. If this element is not present Relevant Jurisdiction is Not Applicable.'

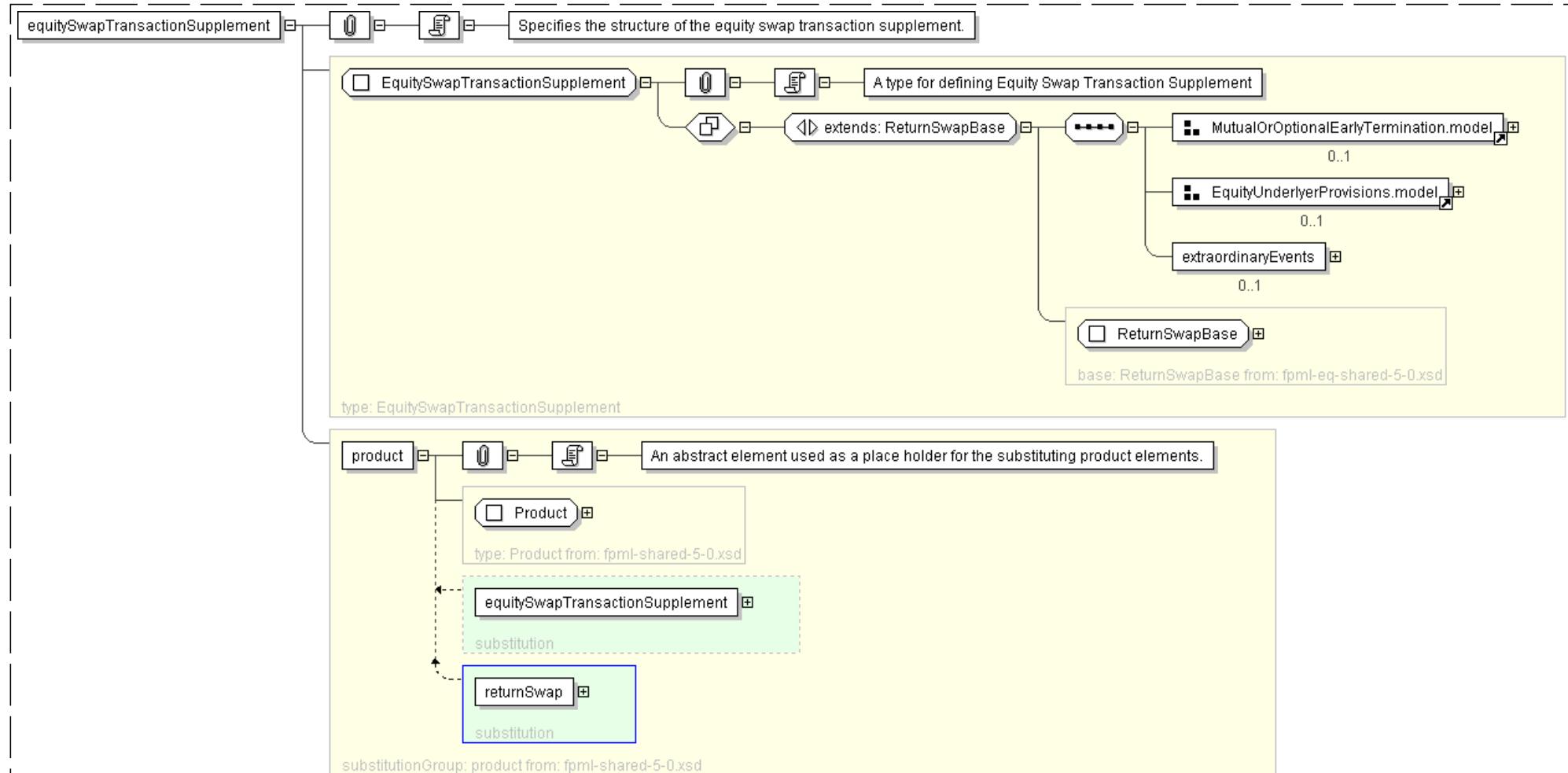
End Group: EquityUnderlyerProvisions.model

<extraordinaryEvents> ExtraordinaryEvents </extraordinaryEvents> [0..1]

'Where the underlying is shares, specifies events affecting the issuer of those shares that may require the terms of the transaction to be adjusted.'

</equitySwapTransactionSupplement>

#### Diagram



#### Schema Component Representation

```
<xsd:element name="equitySwapTransactionSupplement" type="EquitySwapTransactionSupplement"
  " substitutionGroup="product"/>
```

## Global Definitions

### Complex Type: **EquitySwapTransactionSupplement**

**Super-types:** [ReturnSwapBase](#) < **EquitySwapTransactionSupplement** (by extension)

**Sub-types:** None

<b>Name</b>	EquitySwapTransactionSupplement
<b>Used by (from the same schema document)</b>	Element <a href="#">equitySwapTransactionSupplement</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type for defining Equity Swap Transaction Supplement

#### XML Instance Representation

```
<...
  id="xsd:ID [0..1]">
  <productType> ProductType </productType> [0..*]
  'A classification of the type of product. FpML defines a simple product categorization using
  a coding scheme.'
```

```
<productId> ProductId </productId> [0..*]
  'A product 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.'
```

Start Group: [BuyerSeller.model](#) [0..1]

'BuyerSeller.model has been included as an optional child of ReturnSwapBase to support
the situation where an implementor wishes to indicate who has manufactured the Swap
through representing them as the Seller. It may be removed in future major revisions.'

```
<buyerPartyReference> PartyReference </buyerPartyReference> [1]
  'A reference to the party that buys this instrument, ie. pays for this instrument and
  receives the rights defined by it. See 2000 ISDA definitions Article 11.1 (b). In the case
  of FRAs this is the fixed rate payer.'
```

```
<buyerAccountReference> AccountReference </buyerAccountReference> [0..1]
  'A reference to the account that buys this instrument.'
```

```
<sellerPartyReference> PartyReference </sellerPartyReference> [1]
  'A reference to the party that sells ("writes") this instrument, i.e. that grants the
  rights defined by this instrument and in return receives a payment for it. See 2000
  ISDA definitions Article 11.1 (a). In the case of FRAs this is the floating rate payer.'
```

```
<sellerAccountReference> AccountReference </sellerAccountReference> [0..1]
  'A reference to the account that sells this instrument.'
```

End Group: [BuyerSeller.model](#)

<[returnSwapLegreturnSwapLeg](#)> [1..\*]

```
<principalExchangeFeatures> PrincipalExchangeFeatures </principalExchangeFeatures> [0..1]
```

'This is used to document a Fully Funded Return Swap.'

Start Group: MutualOrOptionalEarlyTermination.model [0..1]

Start Choice [1]

```
<mutualEarlyTermination> xsd:boolean </mutualEarlyTermination> [0..1]
```

'Used for specifying whether the Mutual Early Termination Right that is detailed in the Master Confirmation will apply.'

```
<optionalEarlyTermination> xsd:boolean </optionalEarlyTermination> [1]
```

'A Boolean element used for specifying whether the Optional Early Termination clause detailed in the agreement will apply.'

```
<breakFundingRecovery> xsd:boolean </breakFundingRecovery> [0..1]
```

'A Boolean element used for specifying whether the Break Funding Recovery detailed in the agreement will apply.'

Start Sequence [0..1]

```
<breakFeeElection> FeeElectionEnum </breakFeeElection> [1]
```

'Defines the fee type.'

```
<breakFeeRate> NonNegativeDecimal </breakFeeRate> [0..1]
```

End Sequence

End Choice

End Group: MutualOrOptionalEarlyTermination.model

Start Group: EquityUnderlyerProvisions.model [0..1]

Start Group: IndexAnnexFallback.model [0..1]

Start Choice [1]

```
<multipleExchangeIndexAnnexFallback> xsd:boolean </multipleExchangeIndexAnnexFallback> [1]
```

'For an index option transaction, a flag to indicate whether a relevant Multiple Exchange Index Annex is applicable to the transaction. This annex defines additional provisions which are applicable where an index is comprised of component securities that are traded on multiple exchanges.'

```
<componentSecurityIndexAnnexFallback> xsd:boolean </componentSecurityIndexAnnexFallback> [1]
```

'For an index option transaction, a flag to indicate whether a relevant Component Security Index Annex is applicable to the transaction.'

End Choice

End Group: IndexAnnexFallback.model

```
<localJurisdiction> CountryCode </localJurisdiction> [0..1]
```

'Local Jurisdiction is a term used in the AEJ Master Confirmation, which is used to determine local taxes, which shall mean taxes, duties, and similar charges imposed by the taxing authority of the Local Jurisdiction If this element is not present Local Jurisdiction is Not Applicable.'

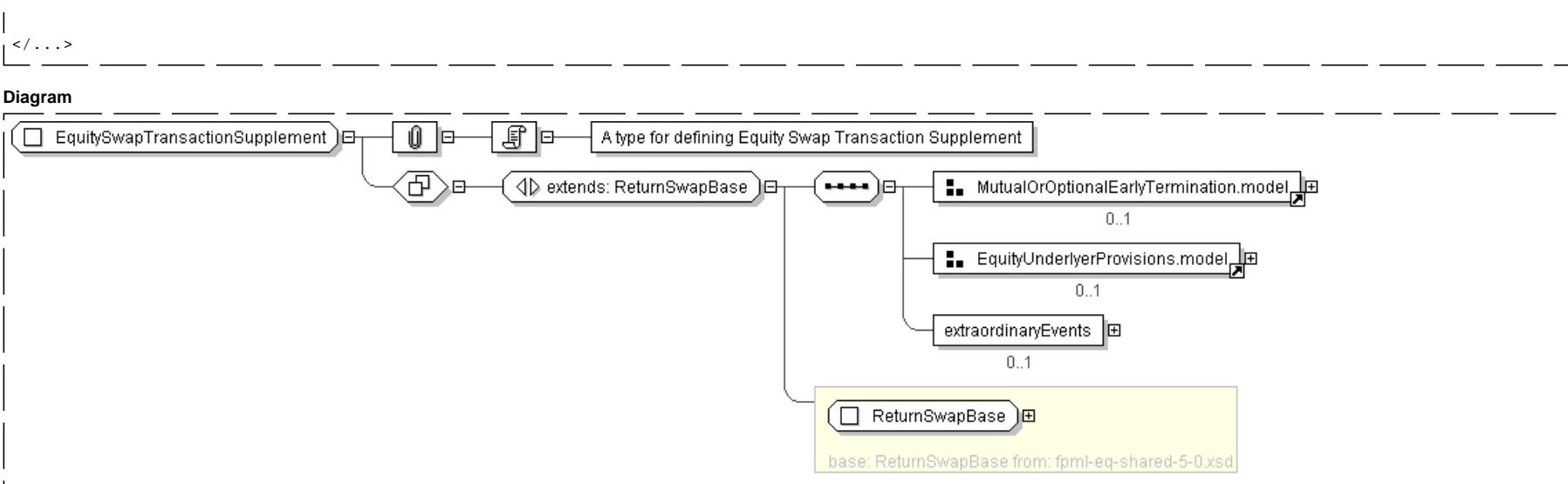
```
<relevantJurisdiction> CountryCode </relevantJurisdiction> [0..1]
```

'Relevant Jurisdiction is a term used in the AEJ Master Confirmation, which is used to determine local taxes, which shall mean taxes, duties and similar charges that would be imposed by the taxing authority of the Country of Underlyer on a Hypothetical Broker Dealer assuming the Applicable Hedge Positions are held by its office in the Relevant Jurisdiction. If this element is not present Relevant Jurisdiction is Not Applicable.'

End Group: EquityUnderlyerProvisions.model

```
<extraordinaryEvents> ExtraordinaryEvents </extraordinaryEvents> [0..1]
```

'Where the underlying is shares, specifies events affecting the issuer of those shares that may require the terms of the transaction to be adjusted.'



### Schema Component Representation

```

<xsd:complexType name="EquitySwapTransactionSupplement">
  <xsd:complexContent>
    <xsd:extension base=" ReturnSwapBase ">
      <xsd:sequence>
        <xsd:group ref=" MutualOrOptionalEarlyTermination.model " minOccurs="0" />
        <xsd:group ref=" EquityUnderlyerProvisions.model " minOccurs="0" />
        <xsd:element name="extraordinaryEvents" type=" ExtraordinaryEvents " minOccurs="0" />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
  
```

top

### Legend

**Complex Type:**  
Schema Component Type

**AusAddress**  
Schema Component Name

**Super-types:** [Address](#) < AusAddress (by extension)

**Sub-types:** • [OLDAddress](#) (by restriction)

If this schema component is a type definition, its type hierarchy is shown in a gray-bordered box.

<b>Name</b>	AusAddress
<b>Abstract</b>	no

The table above displays the properties of this schema component.

### XML Instance Representation

```

< . . . country="Australia" >
  
```

```

<unitNo> string </unitNo> [0..1]
<houseNo> string </houseNo> [1]
<street> string </street> [1]
Start Choice [1]
<city> string </city> [1]
<town> string </town> [1]
End Choice
<state> AusStates </state> [1]
<postcode> string <>pattern = [1-9][0-9]{3}</> </postcode> [1]
</...>

```

The XML Instance Representation table above shows the schema component's content as an XML instance.

- The minimum and maximum occurrence of elements and attributes are provided in square brackets, e.g. [0..1].
- Model group information are shown in gray, e.g. Start Choice ... End Choice.
- For type derivations, the elements and attributes that have been added to or changed from the base type's content are shown in **bold**.
- If an element/attribute has a fixed value, the fixed value is shown in green, e.g. country="Australia".
- Otherwise, the type of the element/attribute is displayed.
  - If the element/attribute's type is in the schema, a link is provided to it.
  - For local simple type definitions, the constraints are displayed in angle brackets, e.g. <>pattern = [1-9][0-9]{3}</>.

### Schema Component Representation

```

<complexType name="AusAddress">
<complexContent>
<extension base="Address">
<sequence>
<element name="state" type="AusStates" />
<element name="postcode">
<simpleType>
<restriction base="string">
<pattern value="[1-9][0-9]{3}" />
</restriction>
</simpleType>
</element>
</sequence>
<attribute name="country" type="string" fixed="Australia" />
</extension>
</complexContent>
</complexType>

```

The Schema Component Representation table above displays the underlying XML representation of the schema component. (Annotations are not shown.)

[top](#)

## Glossary

**Abstract** (Applies to complex type definitions and element declarations). An abstract element or complex type cannot be used to validate an element instance. If there is a reference to an abstract element, only element declarations that can substitute the abstract element can be used to validate the instance. For references to abstract type definitions, only derived types can be used.

**All Model Group** Child elements can be provided *in any order* in instances. See: <http://www.w3.org/TR/xmlschema-1/#element-all>.

**Choice Model Group** Only one from the list of child elements and model groups can be provided in instances. See: <http://www.w3.org/TR/xmlschema-1/#element-choice>.

**Collapse Whitespace Policy** Replace tab, line feed, and carriage return characters with space character (Unicode character 32). Then, collapse contiguous sequences of space characters into single space character, and remove leading and trailing space characters.

**Disallowed Substitutions** (Applies to element declarations). If *substitution* is specified, then [substitution group](#) members cannot be used in place of the given element declaration to validate element instances. If *derivation methods*, e.g. extension, restriction, are specified, then the given element declaration will not validate element instances that have types derived from the element declaration's type using the specified derivation methods. Normally, element instances can override their declaration's type by specifying an `xsi:type` attribute.

**Key Constraint** Like [Uniqueness Constraint](#), but additionally requires that the specified value(s) must be provided. See: [http://www.w3.org/TR/xmlschema-1/#clidentity-constraint\\_Definitions](http://www.w3.org/TR/xmlschema-1/#clidentity-constraint_Definitions).

**Key Reference Constraint** Ensures that the specified value(s) must match value(s) from a [Key Constraint](#) or [Uniqueness Constraint](#). See: [http://www.w3.org/TR/xmlschema-1/#clidentity-constraint\\_Definitions](http://www.w3.org/TR/xmlschema-1/#clidentity-constraint_Definitions).

**Model Group** Groups together element content, specifying the order in which the element content can occur and the number of times the group of element content may be repeated. See: [http://www.w3.org/TR/xmlschema-1/#Model\\_Groups](http://www.w3.org/TR/xmlschema-1/#Model_Groups).

**Nillable** (Applies to element declarations). If an element declaration is nillable, instances can use the `xsi:nil` attribute. The `xsi:nil` attribute is the boolean attribute, *nil*, from the <http://www.w3.org/2001/XMLSchema-instance> namespace. If an element instance has an `xsi:nil` attribute set to true, it can be left empty, even though its element declaration may have required content.

**Notation** A notation is used to identify the format of a piece of data. Values of elements and attributes that are of type, NOTATION, must come from the names of declared notations. See: [http://www.w3.org/TR/xmlschema-1/#cNotation\\_Declarations](http://www.w3.org/TR/xmlschema-1/#cNotation_Declarations).

**Preserve Whitespace Policy** Preserve whitespaces exactly as they appear in instances.

**Prohibited Derivations** (Applies to type definitions). Derivation methods that cannot be used to create sub-types from a given type definition.

**Prohibited Substitutions** (Applies to complex type definitions). Prevents sub-types that have been derived using the specified derivation methods from validating element instances in place of the given type definition.

**Replace Whitespace Policy** Replace tab, line feed, and carriage return characters with space character (Unicode character 32).

**Sequence Model Group** Child elements and model groups must be provided *in the specified order* in instances. See: <http://www.w3.org/TR/xmlschema-1/#element-sequence>.

**Substitution Group** Elements that are *members* of a substitution group can be used wherever the *head* element of the substitution group is referenced.

**Substitution Group Exclusions** (Applies to element declarations). Prohibits element declarations from nominating themselves as being able to substitute a given element declaration, if they have types that are derived from the original element's type using the specified derivation methods.

**Target Namespace** The target namespace identifies the namespace that components in this schema belongs to. If no target namespace is provided, then the schema components do not belong to any namespace.

**Uniqueness Constraint** Ensures uniqueness of an element/attribute value, or a combination of values, within a specified scope. See: [http://www.w3.org/TR/xmlschema-1/#clidentity-constraint\\_Definitions](http://www.w3.org/TR/xmlschema-1/#clidentity-constraint_Definitions).

[top](#)

# XML Schema Documentation

## Table of Contents

- [Schema Document Properties](#)
- [Global Declarations](#)
  - Element: [americanExercise](#)
  - Element: [bermudaExercise](#)
  - Element: [europeanExercise](#)
  - Element: [exercise](#)
  - Element: [product](#)
- [Global Definitions](#)
  - Complex Type: [Account](#)
  - Complex Type: [AccountId](#)
  - Complex Type: [AccountName](#)
  - Complex Type: [AccountReference](#)
  - Complex Type: [Address](#)
  - Complex Type: [AdjustableDate](#)
  - Complex Type: [AdjustableDate2](#)
  - Complex Type: [AdjustableDates](#)
  - Complex Type: [AdjustableDatesOrRelativeDateOffset](#)
  - Complex Type: [AdjustableOrAdjustedDate](#)
  - Complex Type: [AdjustableOrRelativeDate](#)
  - Complex Type: [AdjustableOrRelativeDates](#)
  - Complex Type: [AdjustableRelativeOrPeriodicDates](#)
  - Complex Type: [AdjustableRelativeOrPeriodicDates2](#)
  - Complex Type: [AdjustedRelativeDateOffset](#)
  - Complex Type: [AgreementType](#)
  - Complex Type: [AgreementVersion](#)
  - Complex Type: [AmericanExercise](#)
  - Complex Type: [AmountReference](#)
  - Complex Type: [AmountSchedule](#)
  - Complex Type: [AssetClass](#)
  - Complex Type: [AutomaticExercise](#)
  - Complex Type: [AverageDailyTradingVolumeLimit](#)
  - Complex Type: [Beneficiary](#)
  - Complex Type: [BermudaExercise](#)
  - Complex Type: [BrokerConfirmation](#)
  - Complex Type: [BrokerConfirmationType](#)
  - Complex Type: [BusinessCenter](#)
  - Complex Type: [BusinessCenterTime](#)
  - Complex Type: [BusinessCenters](#)
  - Complex Type: [BusinessCentersReference](#)
  - Complex Type: [BusinessDateRange](#)
  - Complex Type: [BusinessDayAdjustments](#)
  - Complex Type: [BusinessDayAdjustmentsReference](#)
  - Complex Type: [CalculationAgent](#)
  - Complex Type: [CalculationPeriodFrequency](#)
  - Complex Type: [CashSettlementReferenceBanks](#)
  - Complex Type: [CashflowId](#)
  - Complex Type: [CashflowType](#)
  - Complex Type: [ClearanceSystem](#)
  - Complex Type: [ContractualDefinitions](#)
  - Complex Type: [ContractualMatrix](#)
  - Complex Type: [ContractualSupplement](#)
  - Complex Type: [ContractualTermsSupplement](#)
  - Complex Type: [CorrespondentInformation](#)
  - Complex Type: [CountryCode](#)
  - Complex Type: [CreditRating](#)
  - Complex Type: [CreditSeniority](#)
  - Complex Type: [CreditSupportAgreement](#)

- Complex Type: [CreditSupportAgreementIdentifier](#)
- Complex Type: [CreditSupportAgreementType](#)
- Complex Type: [Currency](#)
- Complex Type: [DateList](#)
- Complex Type: [DateOffset](#)
- Complex Type: [DateRange](#)
- Complex Type: [DateReference](#)
- Complex Type: [DateTimeList](#)
- Complex Type: [DayCountFraction](#)
- Complex Type: [DeterminationMethod](#)
- Complex Type: [DeterminationMethodReference](#)
- Complex Type: [Documentation](#)
- Complex Type: [Empty](#)
- Complex Type: [EntityId](#)
- Complex Type: [EntityName](#)
- Complex Type: [EuropeanExercise](#)
- Complex Type: [Exchangeld](#)
- Complex Type: [Exercise](#)
- Complex Type: [ExerciseFee](#)
- Complex Type: [ExerciseFeeSchedule](#)
- Complex Type: [ExerciseNotice](#)
- Complex Type: [ExerciseProcedure](#)
- Complex Type: [FloatingRate](#)
- Complex Type: [FloatingRateCalculation](#)
- Complex Type: [FloatingRateIndex](#)
- Complex Type: [ForecastRateIndex](#)
- Complex Type: [Formula](#)
- Complex Type: [FormulaComponent](#)
- Complex Type: [Frequency](#)
- Complex Type: [FutureValueAmount](#)
- Complex Type: [FxCashSettlement](#)
- Complex Type: [FxFixing](#)
- Complex Type: [FxRate](#)
- Complex Type: [FxSpotRateSource](#)
- Complex Type: [GenericAgreement](#)
- Complex Type: [GoverningLaw](#)
- Complex Type: [GrossCashflow](#)
- Complex Type: [IdentifiedCurrency](#)
- Complex Type: [IdentifiedCurrencyReference](#)
- Complex Type: [IdentifiedDate](#)
- Complex Type: [IdentifiedPayerReceiver](#)
- Complex Type: [IndustryClassification](#)
- Complex Type: [InformationProvider](#)
- Complex Type: [InformationSource](#)
- Complex Type: [InstrumentId](#)
- Complex Type: [InterestAccrualsCompoundingMethod](#)
- Complex Type: [InterestAccrualsMethod](#)
- Complex Type: [IntermediaryInformation](#)
- Complex Type: [InterpolationMethod](#)
- Complex Type: [Leg](#)
- Complex Type: [LegalEntity](#)
- Complex Type: [LegalEntityReference](#)
- Complex Type: [MainPublication](#)
- Complex Type: [ManualExercise](#)
- Complex Type: [MasterAgreement](#)
- Complex Type: [MasterAgreementType](#)
- Complex Type: [MasterAgreementVersion](#)
- Complex Type: [MasterConfirmation](#)
- Complex Type: [MasterConfirmationAnnexType](#)
- Complex Type: [MasterConfirmationType](#)
- Complex Type: [MatchId](#)
- Complex Type: [Math](#)
- Complex Type: [MatrixTerm](#)
- Complex Type: [MatrixType](#)

- Complex Type: [MimeType](#)
- Complex Type: [Money](#)
- Complex Type: [MoneyBase](#)
- Complex Type: [MultipleExercise](#)
- Complex Type: [NonNegativeAmountSchedule](#)
- Complex Type: [NonNegativeMoney](#)
- Complex Type: [NonNegativePayment](#)
- Complex Type: [NonNegativeSchedule](#)
- Complex Type: [NonNegativeStep](#)
- Complex Type: [NotionalAmount](#)
- Complex Type: [NotionalAmountReference](#)
- Complex Type: [NotionalReference](#)
- Complex Type: [Offset](#)
- Complex Type: [OffsetPrevailingTime](#)
- Complex Type: [PartialExercise](#)
- Complex Type: [Party](#)
- Complex Type: [PartyId](#)
- Complex Type: [PartyName](#)
- Complex Type: [PartyReference](#)
- Complex Type: [PartyRelationship](#)
- Complex Type: [PartyRelationshipDocumentation](#)
- Complex Type: [PartyRole](#)
- Complex Type: [PartyRoleType](#)
- Complex Type: [Payment](#)
- Complex Type: [PaymentBase](#)
- Complex Type: [PaymentBaseExtended](#)
- Complex Type: [PaymentDetails](#)
- Complex Type: [PaymentReference](#)
- Complex Type: [PaymentType](#)
- Complex Type: [Period](#)
- Complex Type: [PeriodicDates](#)
- Complex Type: [PositiveAmountSchedule](#)
- Complex Type: [PositiveMoney](#)
- Complex Type: [PositivePayment](#)
- Complex Type: [PositiveSchedule](#)
- Complex Type: [PositiveStep](#)
- Complex Type: [PrevailingTime](#)
- Complex Type: [PricingStructure](#)
- Complex Type: [PricingStructureReference](#)
- Complex Type: [PrincipalExchanges](#)
- Complex Type: [Product](#)
- Complex Type: [ProductId](#)
- Complex Type: [ProductReference](#)
- Complex Type: [ProductType](#)
- Complex Type: [QuotedCurrencyPair](#)
- Complex Type: [Rate](#)
- Complex Type: [RateObservation](#)
- Complex Type: [RateReference](#)
- Complex Type: [RateSourcePage](#)
- Complex Type: [Reference](#)
- Complex Type: [ReferenceAmount](#)
- Complex Type: [ReferenceBank](#)
- Complex Type: [ReferenceBankId](#)
- Complex Type: [RelatedParty](#)
- Complex Type: [RelativeDateOffset](#)
- Complex Type: [RelativeDateSequence](#)
- Complex Type: [RelativeDates](#)
- Complex Type: [RequiredIdentifierDate](#)
- Complex Type: [ResetFrequency](#)
- Complex Type: [ReturnSwapNotionalAmountReference](#)
- Complex Type: [Rounding](#)
- Complex Type: [Routing](#)
- Complex Type: [RoutingExplicitDetails](#)
- Complex Type: [RoutingId](#)

- Complex Type: [RoutingIds](#)
- Complex Type: [RoutingIdsAndExplicitDetails](#)
- Complex Type: [Schedule](#)
- Complex Type: [ScheduleReference](#)
- Complex Type: [SettlementInformation](#)
- Complex Type: [SettlementInstruction](#)
- Complex Type: [SettlementMethod](#)
- Complex Type: [SettlementPriceDefaultElection](#)
- Complex Type: [SettlementPriceSource](#)
- Complex Type: [SettlementRateSource](#)
- Complex Type: [SharedAmericanExercise](#)
- Complex Type: [SimplePayment](#)
- Complex Type: [SplitSettlement](#)
- Complex Type: [SpreadSchedule](#)
- Complex Type: [SpreadScheduleReference](#)
- Complex Type: [SpreadScheduleType](#)
- Complex Type: [Step](#)
- Complex Type: [StepBase](#)
- Complex Type: [StreetAddress](#)
- Complex Type: [Strike](#)
- Complex Type: [StrikeSchedule](#)
- Complex Type: [Stub](#)
- Complex Type: [StubValue](#)
- Complex Type: [TimezoneLocation](#)
- Model Group: [AdjustableDate.model](#)
- Model Group: [BusinessCentersOrReference.model](#)
- Model Group: [BuyerSeller.model](#)
- Model Group: [FloatingRateIndex.model](#)
- Model Group: [PartialExercise.model](#)
- Model Group: [PartiesAndAccounts.model](#)
- Model Group: [PartyAndAccountReferences.model](#)
- Model Group: [PartyInformation.model](#)
- Model Group: [PayerReceiver.model](#)
- Model Group: [PaymentDiscounting.model](#)
- Model Group: [Period.model](#)
- Model Group: [Premium.model](#)
- Model Group: [Product.model](#)
- Model Group: [RoutingExplicitDetails.model](#)
- Model Group: [RoutingIdentification.model](#)
- Model Group: [SettlementAmountOrCurrency.model](#)
- Model Group: [VersionHistory.model](#)
- Simple Type: [CorrelationValue](#)
- Simple Type: [HourMinuteTime](#)
- Simple Type: [NonNegativeDecimal](#)
- Simple Type: [PositiveDecimal](#)
- Simple Type: [RestrictedPercentage](#)
- Simple Type: [Scheme](#)
- Simple Type: [Token60](#)

- [Legend](#)

- [Glossary](#)

[top](#)

## Schema Document Properties

<b>Target Namespace</b>	<a href="http://www.fpml.org/FpML-5/confirmation">http://www.fpml.org/FpML-5/confirmation</a>
<b>Version</b>	\$Revision: 2807 \$
<b>Element and Attribute Namespaces</b>	<ul style="list-style-type: none"> <li>• Global element and attribute declarations belong to this schema's target namespace.</li> <li>• By default, local element declarations belong to this schema's target namespace.</li> <li>• By default, local attribute declarations have no namespace.</li> </ul>
<b>Schema Composition</b>	<ul style="list-style-type: none"> <li>• This schema includes components from the following schema document(s):           <ul style="list-style-type: none"> <li>◦ <a href="#">fpml-enum-5.0.xsd</a></li> </ul> </li> </ul>

## Declared Namespaces

Prefix	Namespace
Default namespace	<a href="http://www.fpml.org/FpML-5/confirmation">http://www.fpml.org/FpML-5/confirmation</a>
xml	http://www.w3.org/XML/1998/namespace
fpml-annotation	http://www.fpml.org/annotation
xsd	http://www.w3.org/2001/XMLSchema
ecore	http://www.eclipse.org/emf/2002/Ecore

### Schema Component Representation

```
<xsd:schema targetNamespace="http://www.fpml.org/FpML-5/confirmation"
  documentRoot="FpML" nsPrefix="conf" package="org.fpml.confirmation" version="$Revision: 2807
$" elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xsd:include schemaLocation="fpml-enum-5-0.xsd"/>
  ...
</xsd:schema>
```

[top](#)

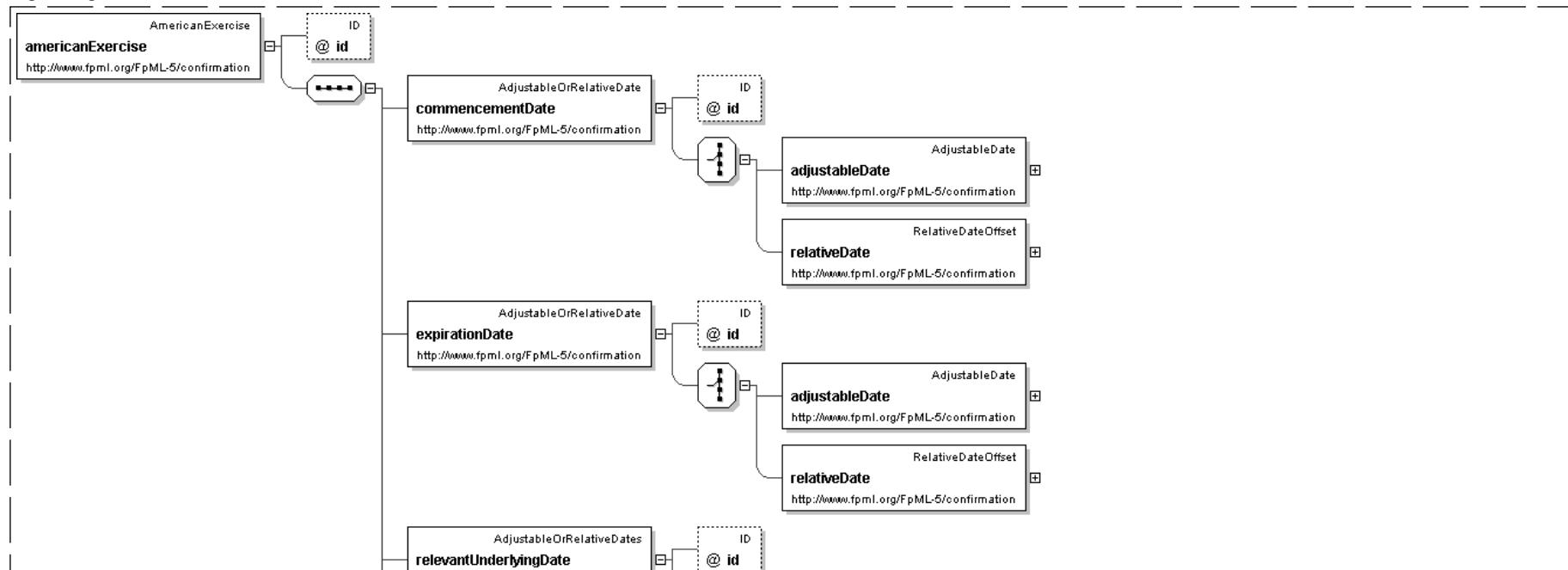
## Global Declarations

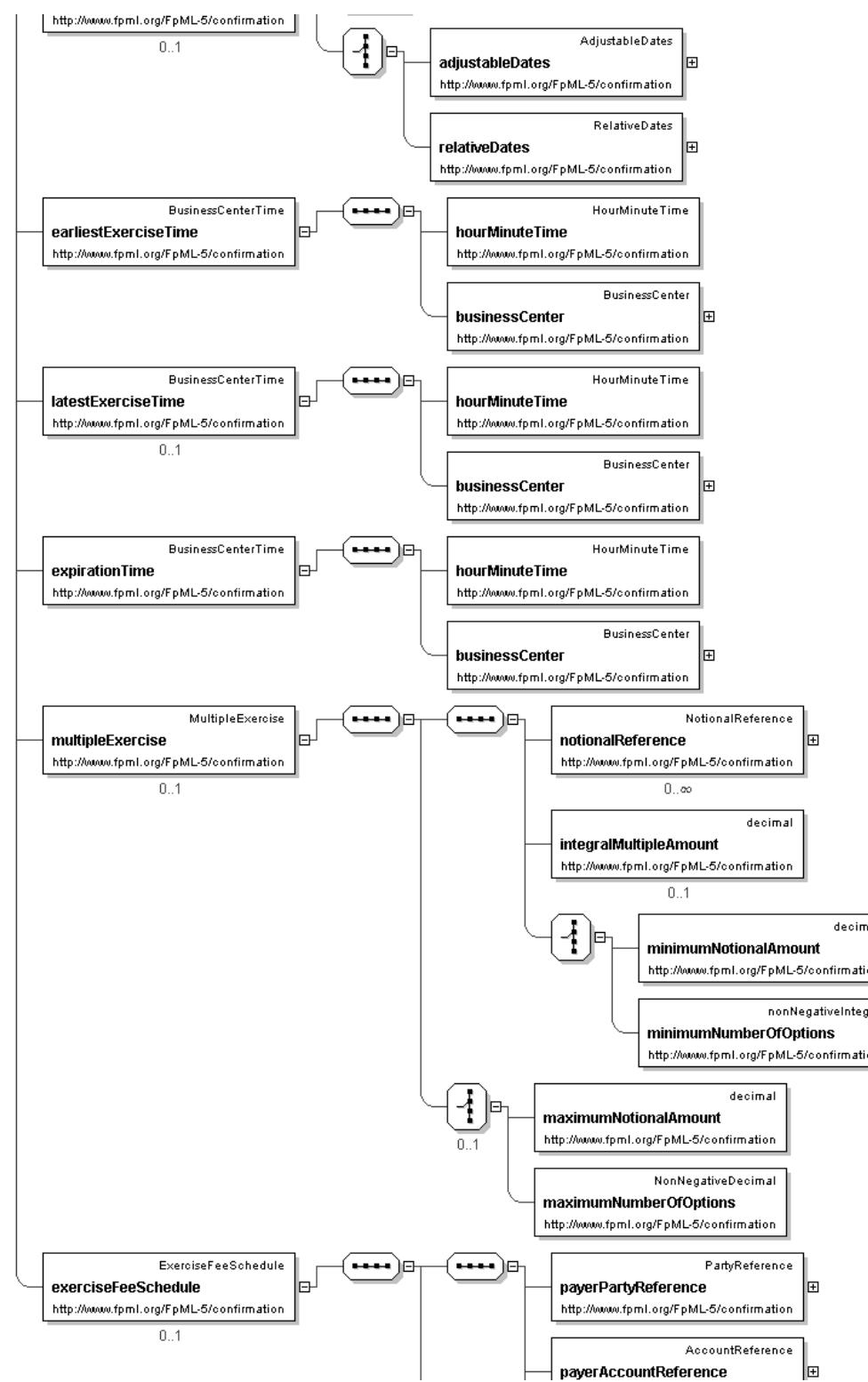
### Element: americanExercise

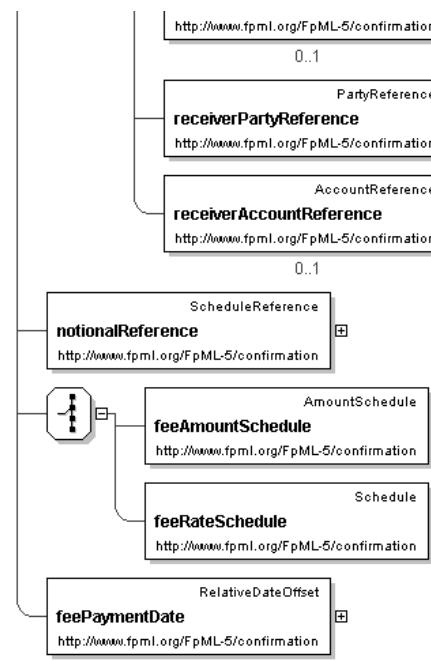
- This element can be used wherever the following element is referenced:
  - [exercise](#)

Name	americanExercise
Type	<a href="#">AmericanExercise</a>
Nullable	no
Abstract	no
Documentation	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.

### Logical Diagram





**XML Instance Representation**

```

<americanExercise
id=" xsd:ID [0..1]>
  <commencementDate> AdjustableOrRelativeDate </commencementDate> [1]
  'The first day of the exercise period for an American style option.'

  <expirationDate> AdjustableOrRelativeDate </expirationDate> [1]
  'The last day within an exercise period for an American style option. For a European
  style option it is the only day within the exercise period.'

  <relevantUnderlyingDate> AdjustableOrRelativeDates </relevantUnderlyingDate> [0..1]
  'The date on the underlying set by the exercise of an option. What this date is depends on
  the option (e.g. in a swaption it is the effective date, in an extendible/cancelable
  provision it is the termination date).'

  <earliestExerciseTime> BusinessCenterTime </earliestExerciseTime> [1]
  '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 Bermuda
  style option the commencement date to, and including, the expiration date , in the case of
  an American option.'

  <latestExerciseTime> BusinessCenterTime </latestExerciseTime> [0..1]
  'For a Bermuda or American style option, the latest time on an exercise business day
  (excluding the expiration date) within the exercise period that notice can be given by
  the 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> BusinessCenterTime </expirationTime> [1]
  'The latest time for exercise on expirationDate.'

  <multipleExercise> MultipleExercise </multipleExercise> [0..1]
  '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.'
  
```

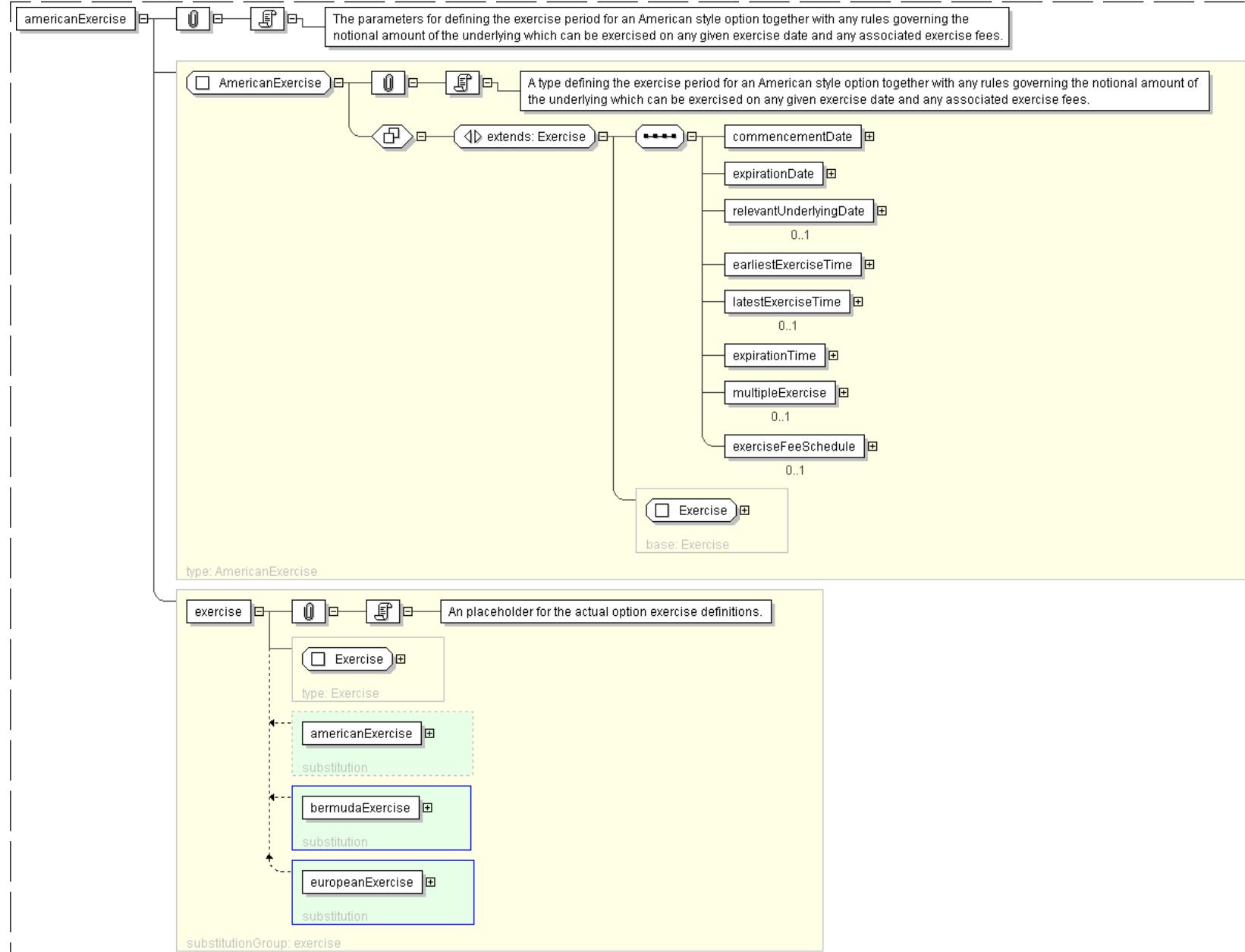
*amount, and if an integral multiple amount is specified, the notional amount exercised must be equal to, or be an integral multiple of, the integral multiple amount.'*

<exerciseFeeSchedule> ExerciseFeeSchedule </exerciseFeeSchedule> [0..1]

'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.'

</americanExercise>

#### Diagram



## Schema Component Representation

```
<xsd:element name="americanExercise" type=" AmericanExercise " substitutionGroup="exercise"/>
```

[top](#)

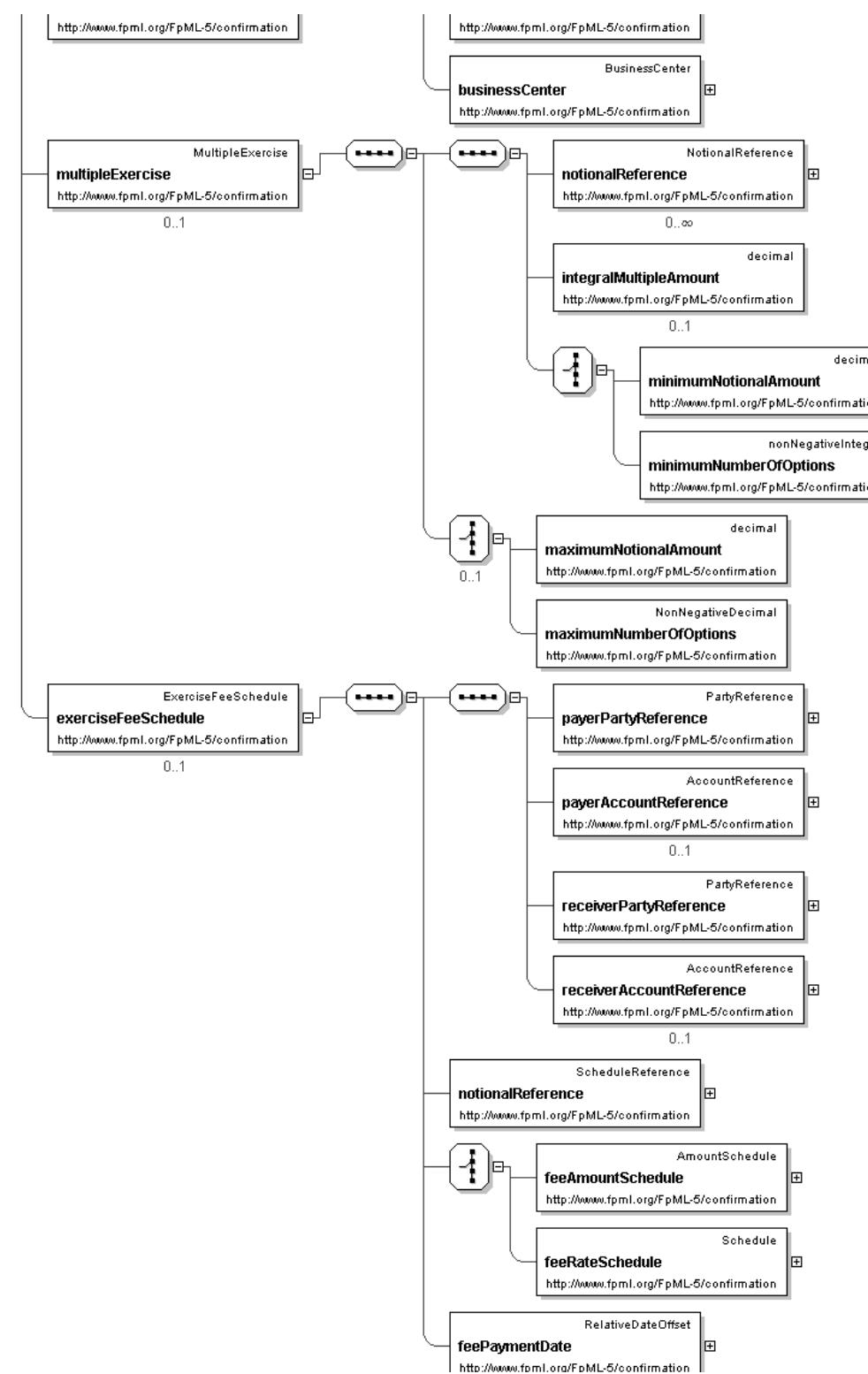
## Element: bermudaExercise

- This element can be used wherever the following element is referenced:
  - [exercise](#)

Name	bermudaExercise
Type	<a href="#">BermudaExercise</a>
Nillable	no
Abstract	no
Documentation	The parameters for defining the exercise period for a Bermuda 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.

## Logical Diagram





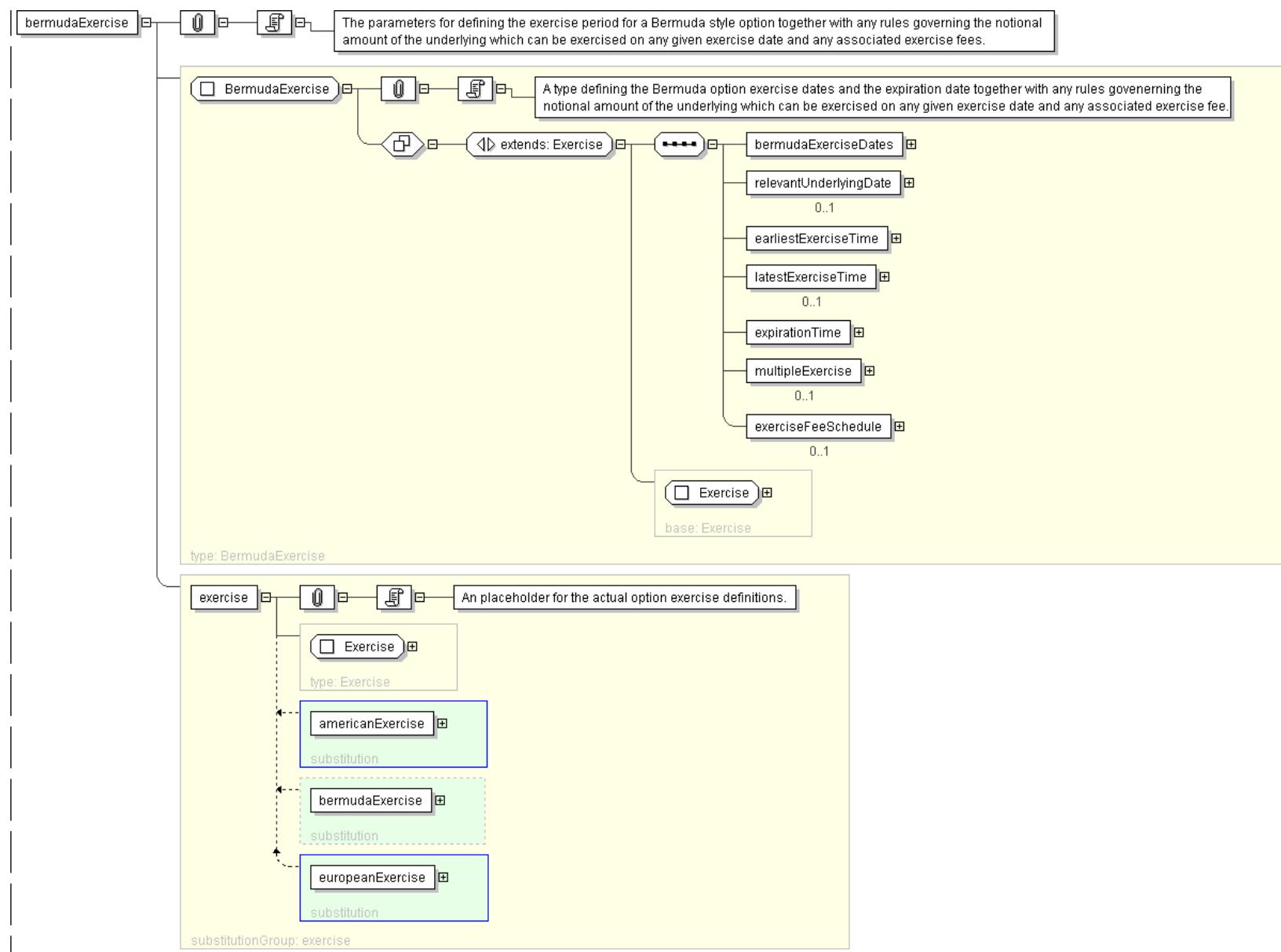
**XML Instance Representation**

```

<bermudaExercise
id=" xsd:ID [0..1]>
  <bermudaExerciseDates> AdjustableOrRelativeDates </bermudaExerciseDates> [1]
    'The dates that define the Bermuda option exercise dates and the expiration date. The last specified 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> AdjustableOrRelativeDates </relevantUnderlyingDate> [0..1]
    'The day on the underlying set by the exercise of an option. What this date is depends on the option (e.g. in a swaption it is the effective date, in an extendible/cancelable provision it is the termination date).'
  <earliestExerciseTime> BusinessCenterTime </earliestExerciseTime> [1]
    '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 Bermuda style option the commencement date to, and including, the expiration date , in the case of an American option.'
  <latestExerciseTime> BusinessCenterTime </latestExerciseTime> [0..1]
    'For a Bermuda or American style option, the latest time on an exercise business day (excluding the expiration date) within the exercise period that notice can be given by the 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> BusinessCenterTime </expirationTime> [1]
    'The latest time for exercise on expirationDate.'
  <multipleExercise> MultipleExercise </multipleExercise> [0..1]
    '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 intergral multiple of, the integral multiple amount.'
  <exerciseFeeSchedule> ExerciseFeeSchedule </exerciseFeeSchedule> [0..1]
    '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.'
</bermudaExercise>

```

**Diagram**

**Schema Component Representation**

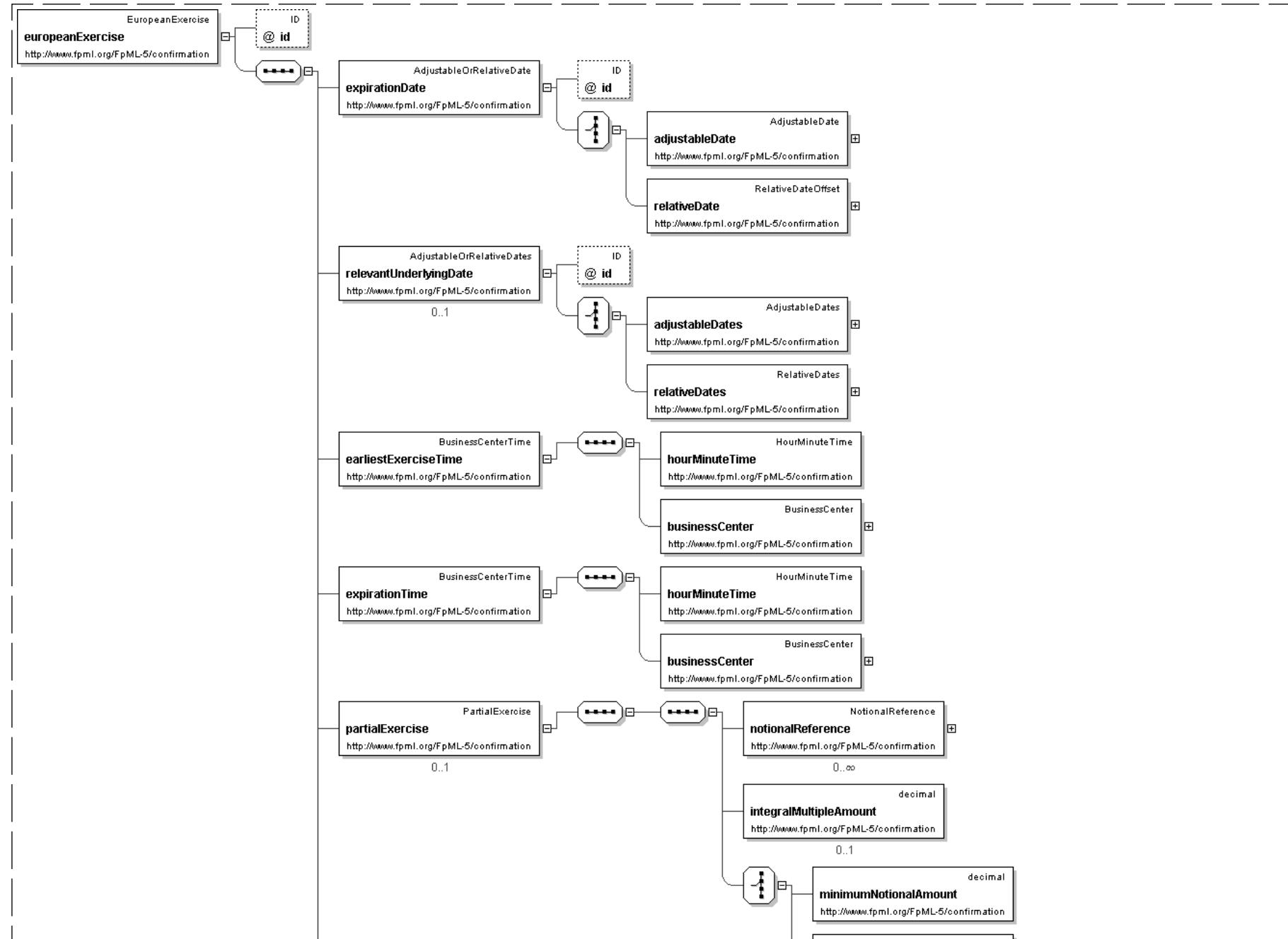
```
<xsd:element name="bermudaExercise" type="BermudaExercise" substitutionGroup="exercise"/>
```

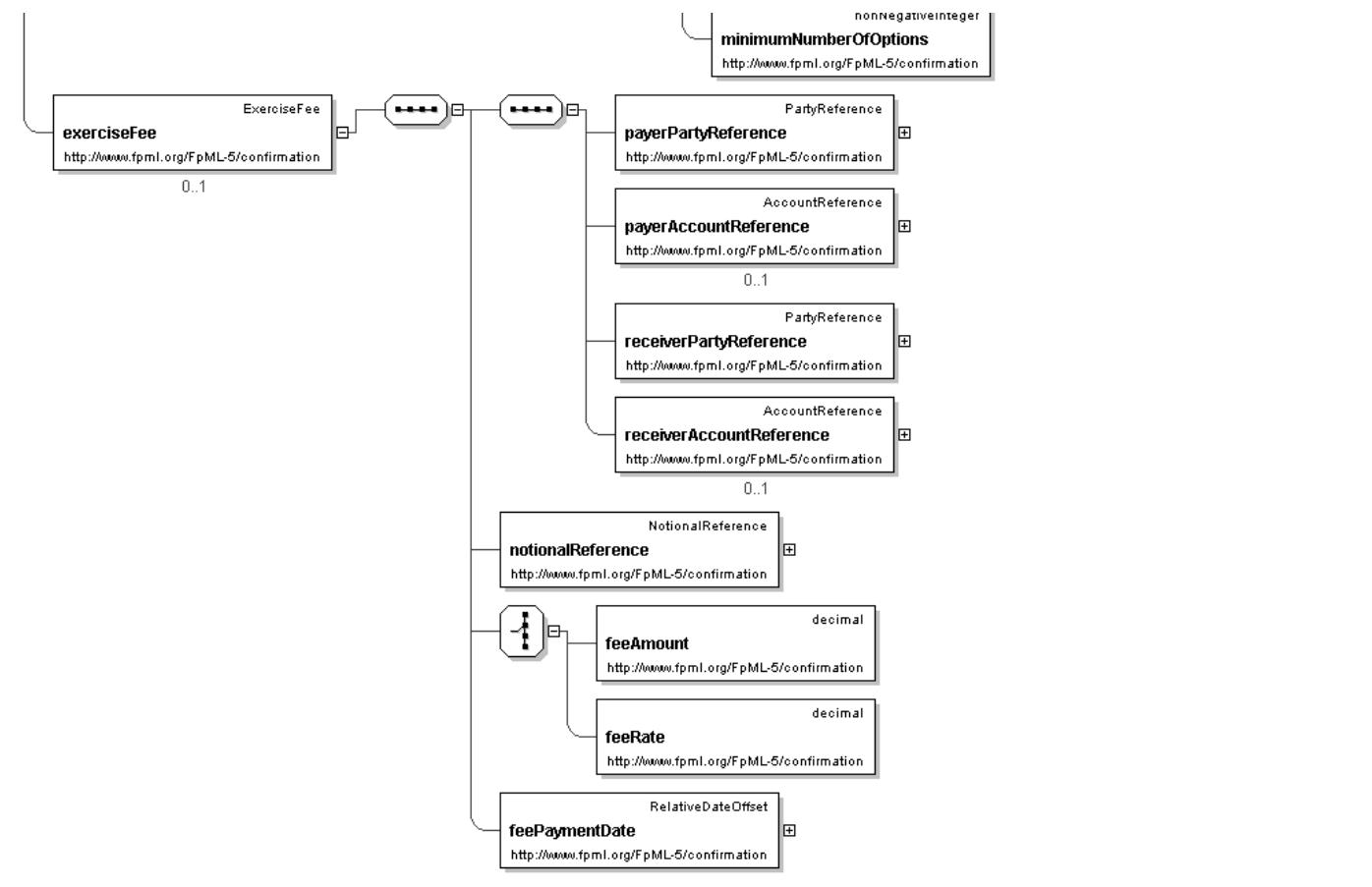
top

**Element: europeanExercise**

- This element can be used wherever the following element is referenced:
  - [exercise](#)

<b>Name</b>	europeanExercise
<b>Type</b>	<a href="#">EuropeanExercise</a>
<b>Nullable</b>	no
<b>Abstract</b>	no
<b>Documentation</b>	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.

**Logical Diagram**

**XML Instance Representation**

```

<europeanExercise
id=" xsd:ID [0..1]">
  <expirationDate> AdjustableOrRelativeDate </expirationDate> [1]
  'The last day within an exercise period for an American style option. For a European
  style option it is the only day within the exercise period.'

  <relevantUnderlyingDate> AdjustableOrRelativeDates </relevantUnderlyingDate> [0..1]
  'The day on the underlying set by the exercise of an option. What this date is depends on
  the option (e.g. in a swaption it is the effective date, in an extendible/cancelable
  provision it is the termination date).'

  <earliestExerciseTime> BusinessCenterTime </earliestExerciseTime> [1]
  '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 Bermuda
  style option the commencement date to, and including, the expiration date , in the case of
  an American option.'

  <expirationTime> BusinessCenterTime </expirationTime> [1]
  'The latest time for exercise on expirationDate.'

  <partialExercise> PartialExercise </partialExercise> [0..1]
  '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

```

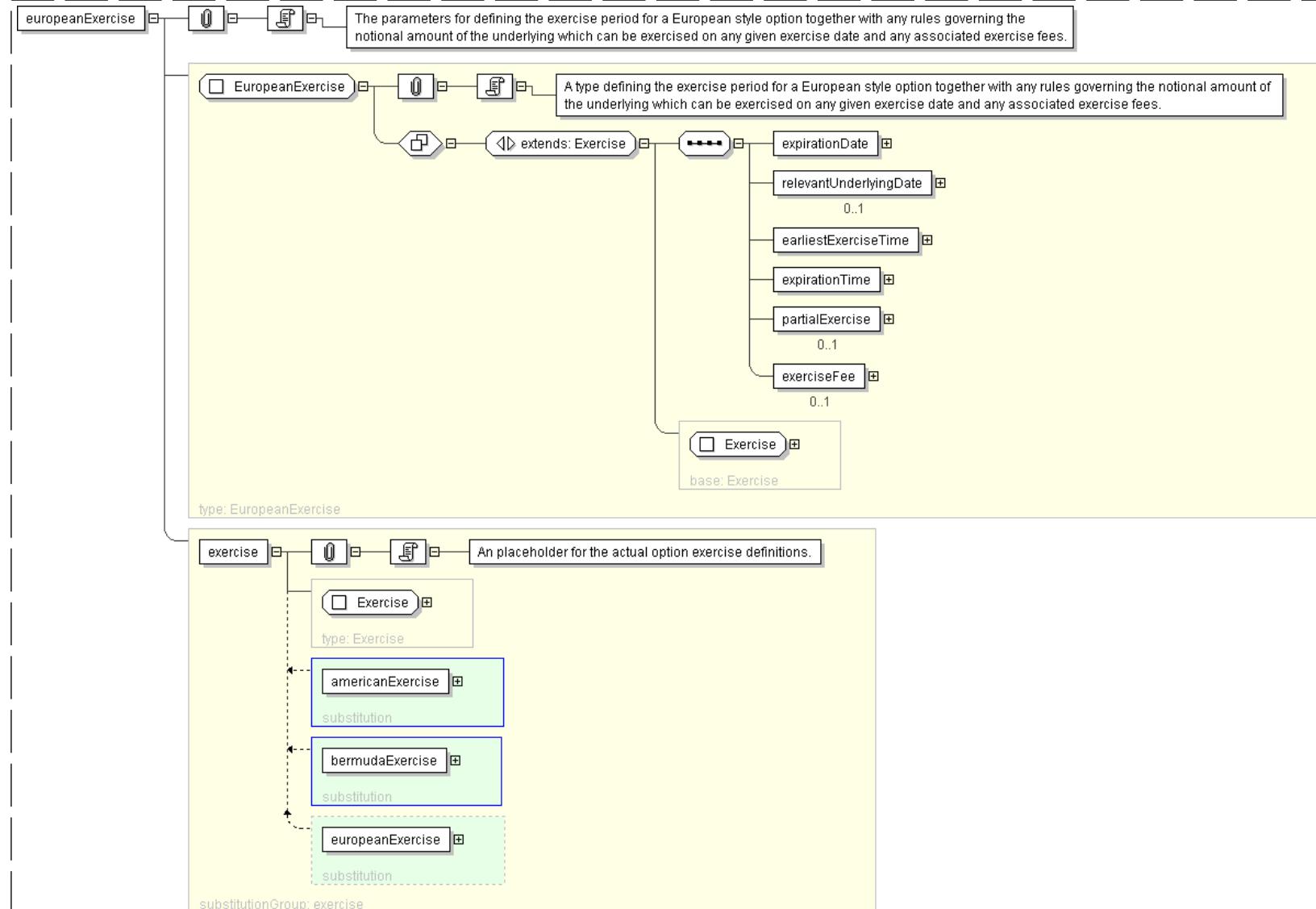
notional amount, and if an integral multiple amount is specified, the notional amount exercised must be equal to, or be an integral multiple of, the integral multiple amount.'

<exerciseFee> **ExerciseFee** </exerciseFee> [0..1]

'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.'

</europeanExercise>

#### Diagram



#### Schema Component Representation

```
<xsd:element name="europeanExercise" type=" EuropeanExercise " substitutionGroup="exercise"/>
```

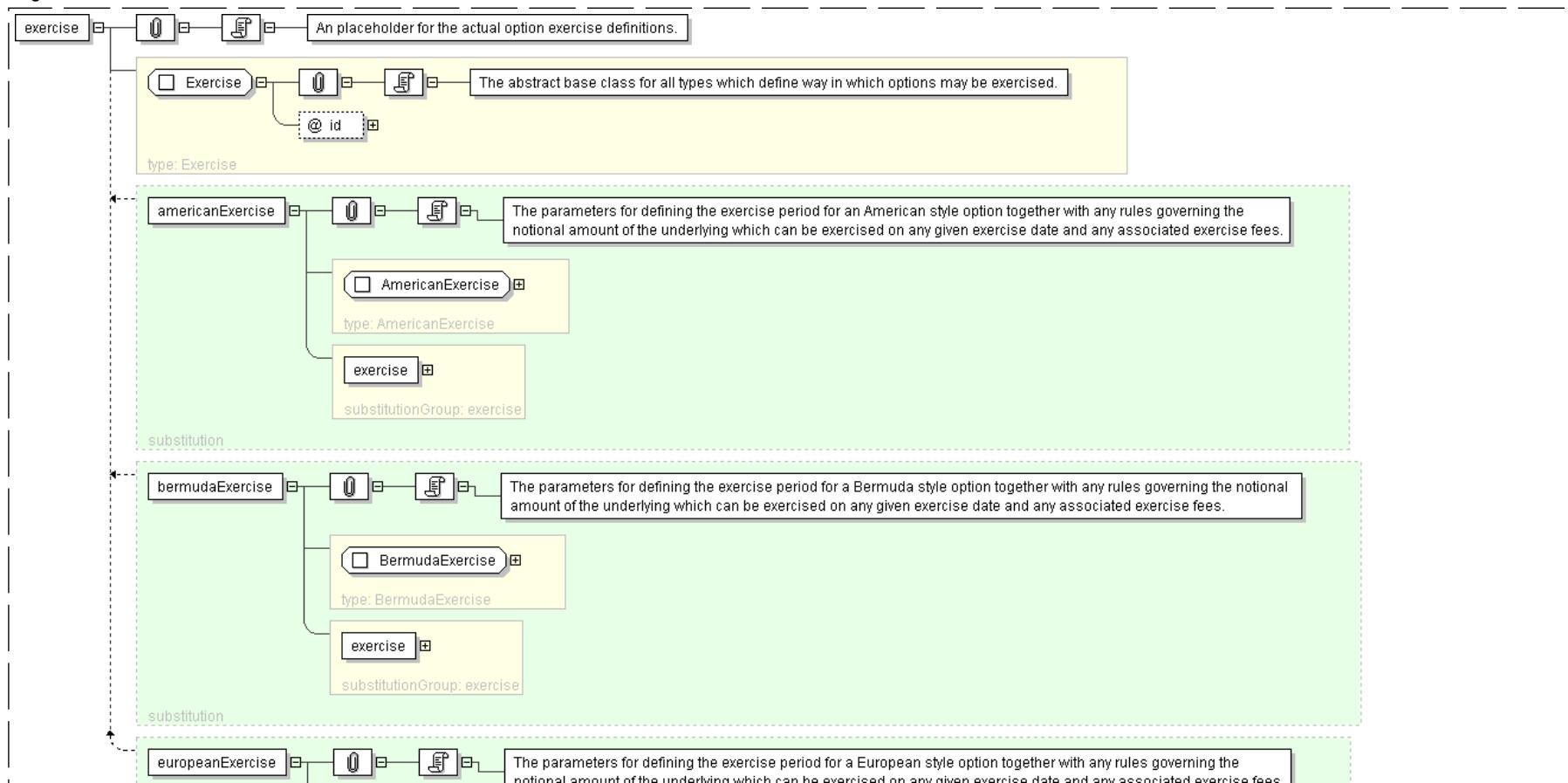
**Element: exercise**

- The following elements can be used wherever this element is referenced:
  - [americanExercise](#)
  - [bermudaExercise](#)
  - [europeanExercise](#)

Name	exercise
Type	<a href="#">Exercise</a>
Nillable	no
Abstract	yes
Documentation	An placeholder for the actual option exercise definitions.

**Logical Diagram****XML Instance Representation**

```
<exercise
id=" xsd:ID [0..1]" />
```

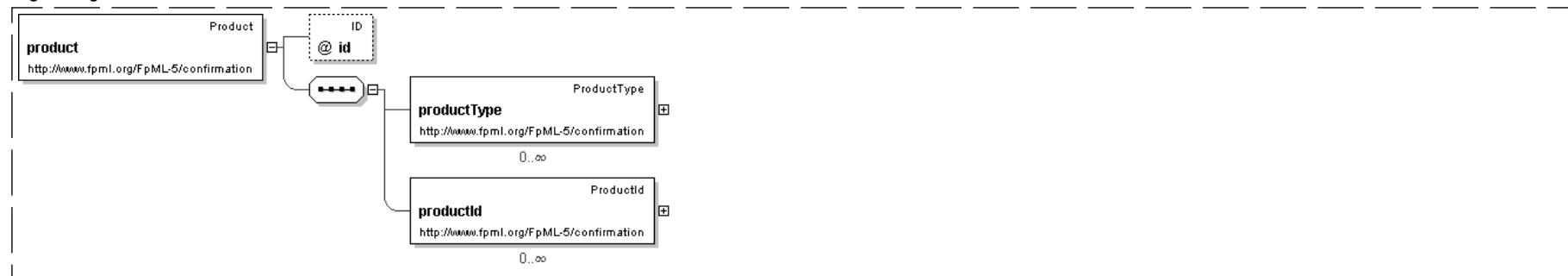
**Diagram**

**Schema Component Representation**

```
<xsd:element name="exercise" type="#Exercise" abstract="true"/>
```

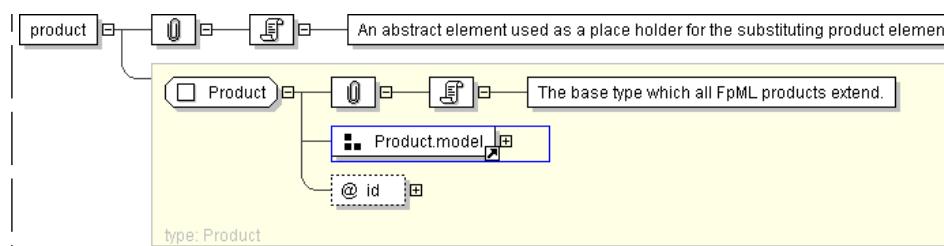
[top](#)**Element: product**

<b>Name</b>	product
<b>Type</b>	Product
<b>Nullable</b>	no
<b>Abstract</b>	yes
<b>Documentation</b>	An abstract element used as a place holder for the substituting product elements.

**Logical Diagram****XML Instance Representation**

```
<product
  id=" " xsd:ID [0..1]>
  <productType> ProductType </productType> [0..*]
  'A classification of the type of product. FpML defines a simple product categorization using
  a coding scheme.'
  <productId> ProductId </productId> [0..*]
  'A product 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.'
</product>
```

**Diagram**

**Schema Component Representation**

```
<xsd:element name="product" type=" Product " abstract="true"/>
```

top

**Global Definitions****Complex Type: Account**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None

<b>Name</b>	Account
<b>Used by (from the same schema document)</b>	Model Group <a href="#">PartiesAndAccounts.model</a>
<b>Abstract</b>	no
<b>Documentation</b>	A generic account that represents any party's account at another party. Parties may be identified by the account at another party.

**XML Instance Representation**

```
<...>
  id=" xsd:ID [1]
  'The unique identifier for the account within the document.'

  ">
  Start Sequence [1..*]
    <accountId> AccountId </accountId> [1]
    'An account identifier. For example an Account number.'

    <accountName> AccountName </accountName> [0..1]
    'The name by which the account is known.

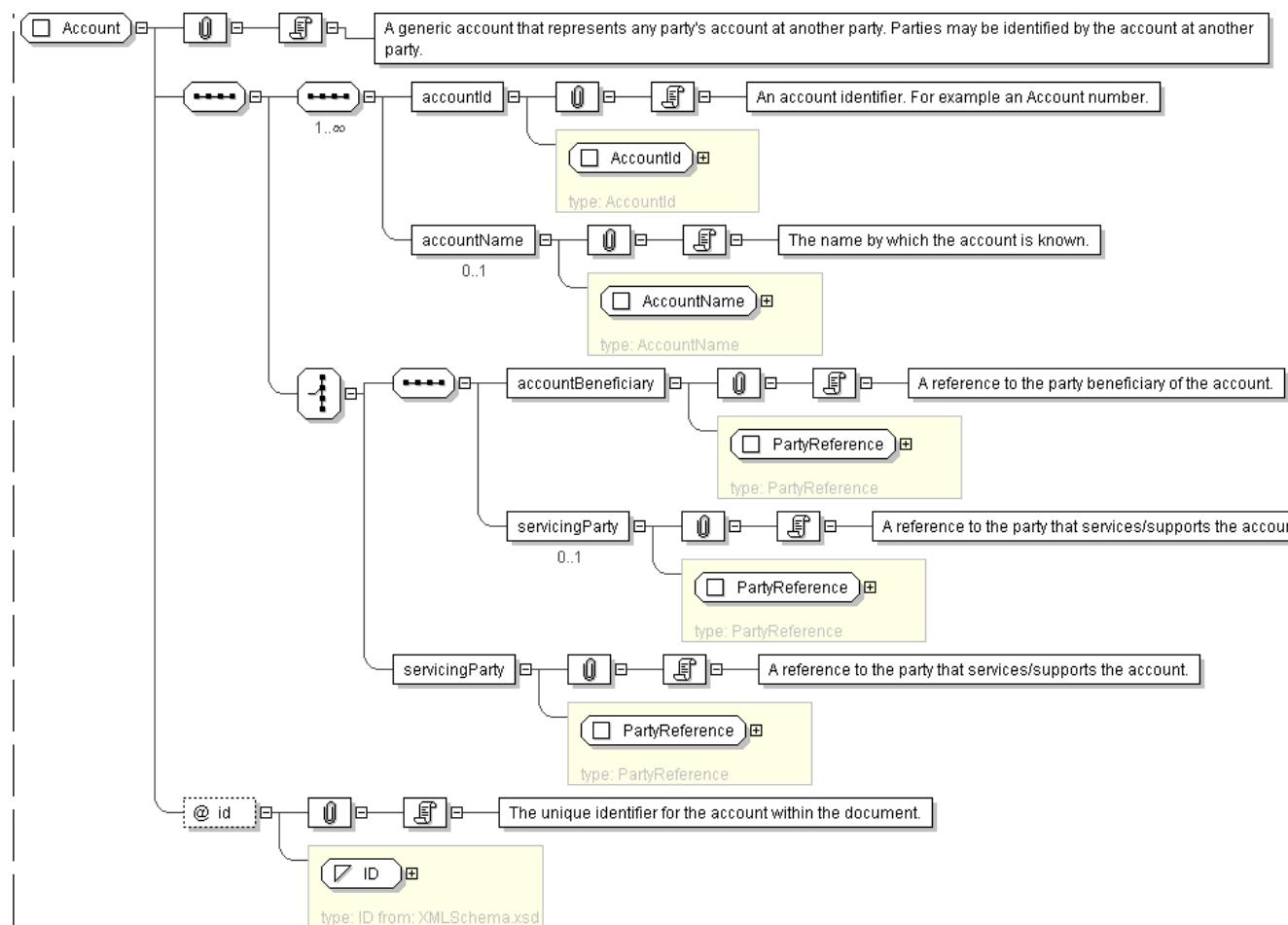
  End Sequence
  Start Choice [1]
    <accountBeneficiary> PartyReference </accountBeneficiary> [1]
    'A reference to the party beneficiary of the account.'

    <servicingParty> PartyReference </servicingParty> [0..1]
    'A reference to the party that services/supports the account.

    <servicingParty> PartyReference </servicingParty> [1]
    'A reference to the party that services/supports the account.

  End Choice
</...>
```

**Diagram**

**Schema Component Representation**

```

<xsd:complexType name="Account">
  <xsd:sequence>
    <xsd:sequence maxOccurs="unbounded">
      <xsd:element name="accountId" type=" AccountId " />
      <xsd:element name="accountName" type=" AccountName " minOccurs="0" />
    </xsd:sequence>
    <xsd:choice>
      <xsd:sequence>
        <xsd:element name="accountBeneficiary" type=" PartyReference " />
        <xsd:element name="servicingParty" type=" PartyReference " minOccurs="0" />
      </xsd:sequence>
      <xsd:element name="servicingParty" type=" PartyReference " />
    </xsd:choice>
  </xsd:sequence>
  <xsd:attribute name="id" type=" xsd:ID " use="required"/>
</xsd:complexType>

```

Super-types:	<a href="#">xsd:normalizedString &lt; Scheme</a> (by restriction) < <b>AccountId</b> (by extension)
Sub-types:	None

Name	AccountId
Used by (from the same schema document)	Complex Type <a href="#">Account</a>
Abstract	no
Documentation	The data type used for account identifiers.

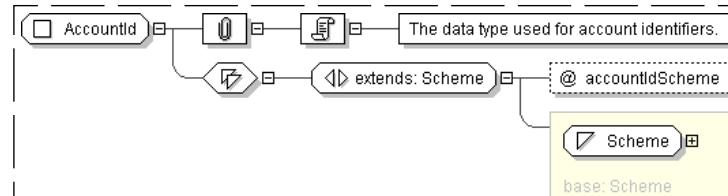
**XML Instance Representation**

```
<...>
<accountIdScheme=" xsd:anyURI [0..1]>
'The identifier scheme used with this accountId. A unique URI to determine the
authoritative issuer of these identifiers.'
```

&lt;"&gt;

[Scheme](#)

&lt;/...&gt;

**Diagram****Schema Component Representation**

```
<xsd:complexType name="AccountId">
  <xsd:simpleContent>
    <xsd:extension base=" Scheme ">
      <xsd:attribute name="accountIdScheme" type=" xsd:anyURI " />
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
```

[top](#)**Complex Type: AccountName**

Super-types:	<a href="#">xsd:normalizedString &lt; Scheme</a> (by restriction) < <b>AccountName</b> (by extension)
Sub-types:	None

Name	AccountName
Used by (from the same schema document)	Complex Type <a href="#">Account</a>
Abstract	no
Documentation	The data type used for the name of the account.

**XML Instance Representation**

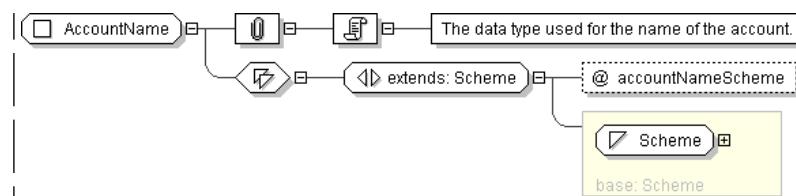
```
<...>
<accountNameScheme=" xsd:anyURI [0..1]>
'The identifier scheme used with this accountName. A unique URI to determine the source of
the account name.'
```

&lt;"&gt;

[Scheme](#)

&lt;/...&gt;

**Diagram**

**Schema Component Representation**

```

<xsd:complexType name="AccountName">
  <xsd:simpleContent>
    <xsd:extension base=" Scheme ">
      <xsd:attribute name="accountNameScheme" type=" xsd:anyURI " use="optional "/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
  
```

[top](#)**Complex Type: AccountReference**

**Super-types:** [Reference](#) < **AccountReference** (by extension)  
**Sub-types:** None

**Name** AccountReference  
**Used by (from the same schema document)** Model Group [BuyerSeller.model](#), Model Group [BuyerSeller.model](#), Model Group [PartyAndAccountReferences.model](#), Model Group [PayerReceiver.model](#), Model Group [PayerReceiver.model](#)

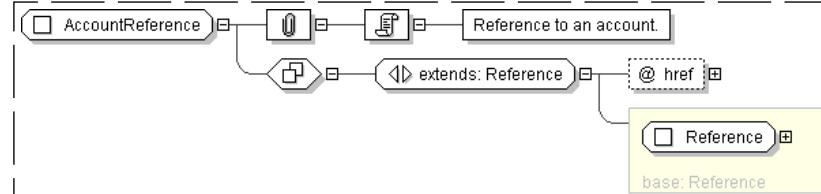
**Abstract** no

**Documentation** Reference to an account.

**XML Instance Representation**

```

<...
  href=" xsd:IDREF [1]" />
  
```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="AccountReference">
  <xsd:complexContent>
    <xsd:extension base=" Reference ">
      <xsd:attribute name="href" type=" xsd:IDREF " use="required" reference="Account"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
  
```

[top](#)**Complex Type: Address**

**Super-types:** None

Sub-types:

None

Name	Address
Used by (from the same schema document)	Model Group <a href="#">RoutingExplicitDetails.model</a>
Abstract	no
Documentation	A type that represents a physical postal address.

**XML Instance Representation**

```
<...>
<streetAddress> StreetAddress </streetAddress> [0..1]
'The set of street and building number information that identifies a postal address within
a city.'

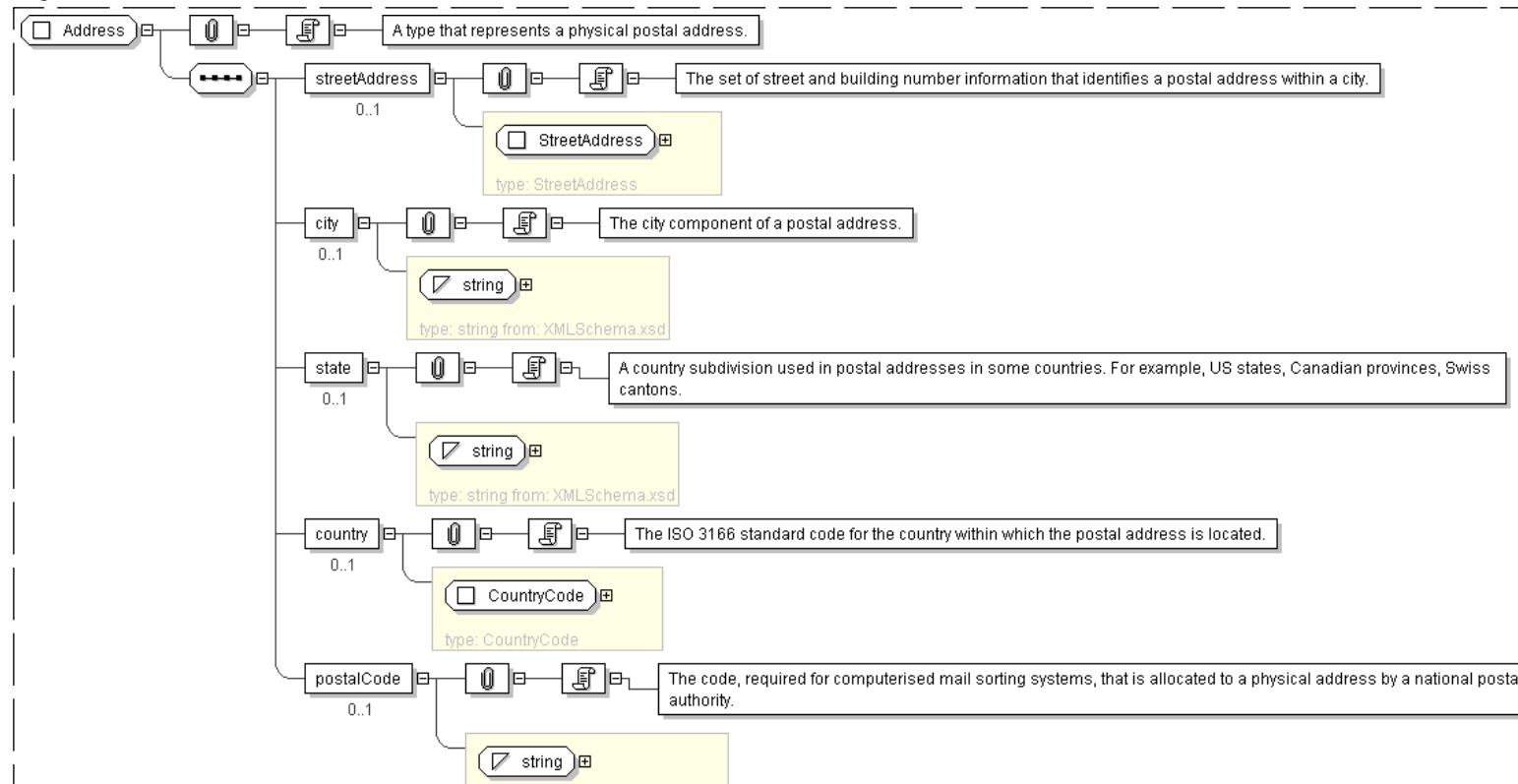
<city> xsd:string </city> [0..1]
'The city component of a postal address.'

<state> xsd:string </state> [0..1]
'A country subdivision used in postal addresses in some countries. For example, US
states, Canadian provinces, Swiss cantons.'

<country> CountryCode </country> [0..1]
'The ISO 3166 standard code for the country within which the postal address is located.'

<postalCode> xsd:string </postalCode> [0..1]
'The code, required for computerised mail sorting systems, that is allocated to a
physical address by a national postal authority.'

</...>
```

**Diagram**

type: string from: XMLSchema.xsd

**Schema Component Representation**

```
<xsd:complexType name="Address">
  <xsd:sequence>
    <xsd:element name="streetAddress" type=" StreetAddress " minOccurs="0"/>
    <xsd:element name="city" type=" xsd:string " minOccurs="0"/>
    <xsd:element name="state" type=" xsd:string " minOccurs="0"/>
    <xsd:element name="country" type=" CountryCode " minOccurs="0"/>
    <xsd:element name="postalCode" type=" xsd:string " minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>
```

top

**Complex Type: AdjustableDate**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None

<b>Name</b>	AdjustableDate
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">AdjustableOrRelativeDate</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type 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.

**XML Instance Representation**

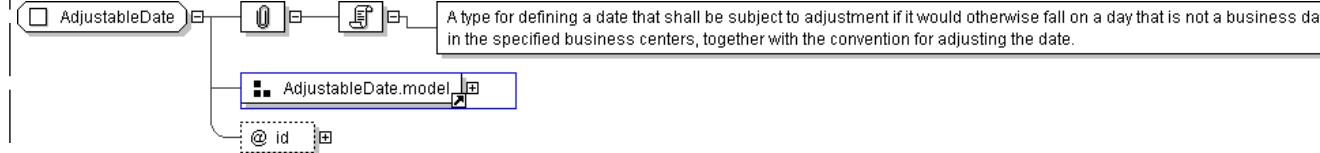
```
<...>
  id=" xsd:ID [0..1]">
  <unadjustedDate> IdentifiedDate </unadjustedDate> [1]
    'A date subject to adjustment.'
```

<dateAdjustments> [BusinessDayAdjustments](#) </dateAdjustments> [1]

'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 date in the specified business centers.'

<adjustedDate> [IdentifiedDate](#) </adjustedDate> [0..1]

'The date once the adjustment has been performed. (Note that this date may change if the business center holidays change.)'

**Diagram****Schema Component Representation**

```
<xsd:complexType name="AdjustableDate">
  <xsd:group ref="#AdjustableDate.model" />
  <xsd:attribute name="id" type=" xsd:ID " />
</xsd:complexType>
```

top

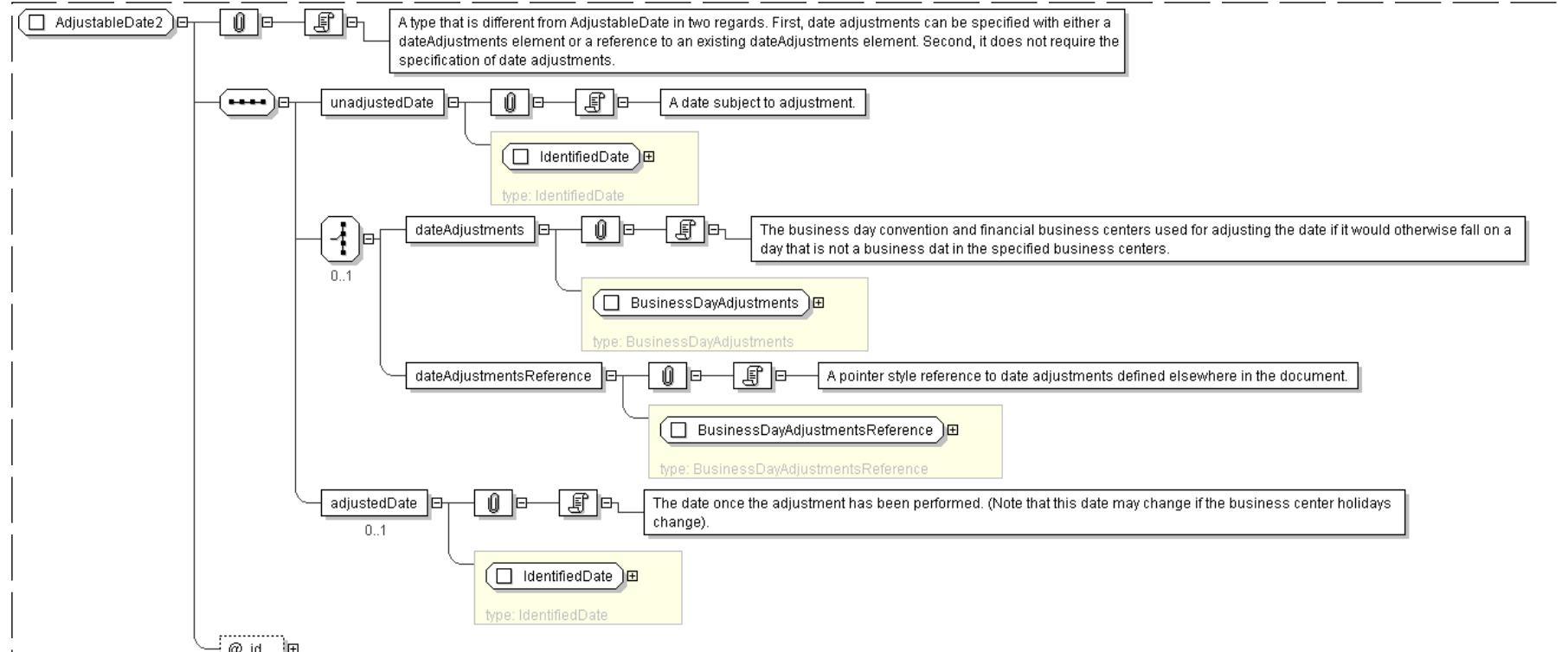
**Complex Type: AdjustableDate2**

**Super-types:** None  
**Sub-types:** None

<b>Name</b>	AdjustableDate2
<b>Abstract</b>	no
<b>Documentation</b>	A type that is different from AdjustableDate in two regards. First, date adjustments can be specified with either a dateAdjustments element or a reference to an existing dateAdjustments element. Second, it does not require the specification of date adjustments.

**XML Instance Representation**

```
<...>
  <id=" xsd:ID [0..1]">
    <unadjustedDate> IdentifiedDate </unadjustedDate> [1]
      'A date subject to adjustment.'
    Start Choice [0..1]
      <dateAdjustments> BusinessDayAdjustments </dateAdjustments> [1]
        '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 dat in the specified business centers.'
      <dateAdjustmentsReference> BusinessDayAdjustmentsReference </dateAdjustmentsReference> [1]
        'A pointer style reference to date adjustments defined elsewhere in the document.'
    End Choice
    <adjustedDate> IdentifiedDate </adjustedDate> [0..1]
      'The date once the adjustment has been performed. (Note that this date may change if the business center holidays change.).'
  </...>
```

**Diagram**

**Schema Component Representation**

```

<xsd:complexType name="AdjustableDate2">
  <xsd:sequence>
    <xsd:element name="unadjustedDate" type="#IdentifiedDate" />
    <xsd:choice minOccurs="0">
      <xsd:element name="dateAdjustments" type="BusinessDayAdjustments" />
      <xsd:element name="dateAdjustmentsReference" type="BusinessDayAdjustmentsReference" />
    </xsd:choice>
    <xsd:element name="adjustedDate" type="#IdentifiedDate" minOccurs="0"/>
  </xsd:sequence>
  <xsd:attribute name="id" type="xsd:ID" />
</xsd:complexType>

```

[top](#)**Complex Type: AdjustableDates**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None

<b>Name</b>	AdjustableDates
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">AdjustableDatesOrRelativeDateOffset</a> , Complex Type <a href="#">AdjustableOrRelativeDates</a> , Complex Type <a href="#">AdjustableRelativeOrPeriodicDates</a> , Complex Type <a href="#">AdjustableRelativeOrPeriodicDates2</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type 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.

**XML Instance Representation**

```

<...
  id="xsd:ID [0..1]">
  <unadjustedDate> IdentifiedDate </unadjustedDate> [1..*]
  'A date subject to adjustment.'

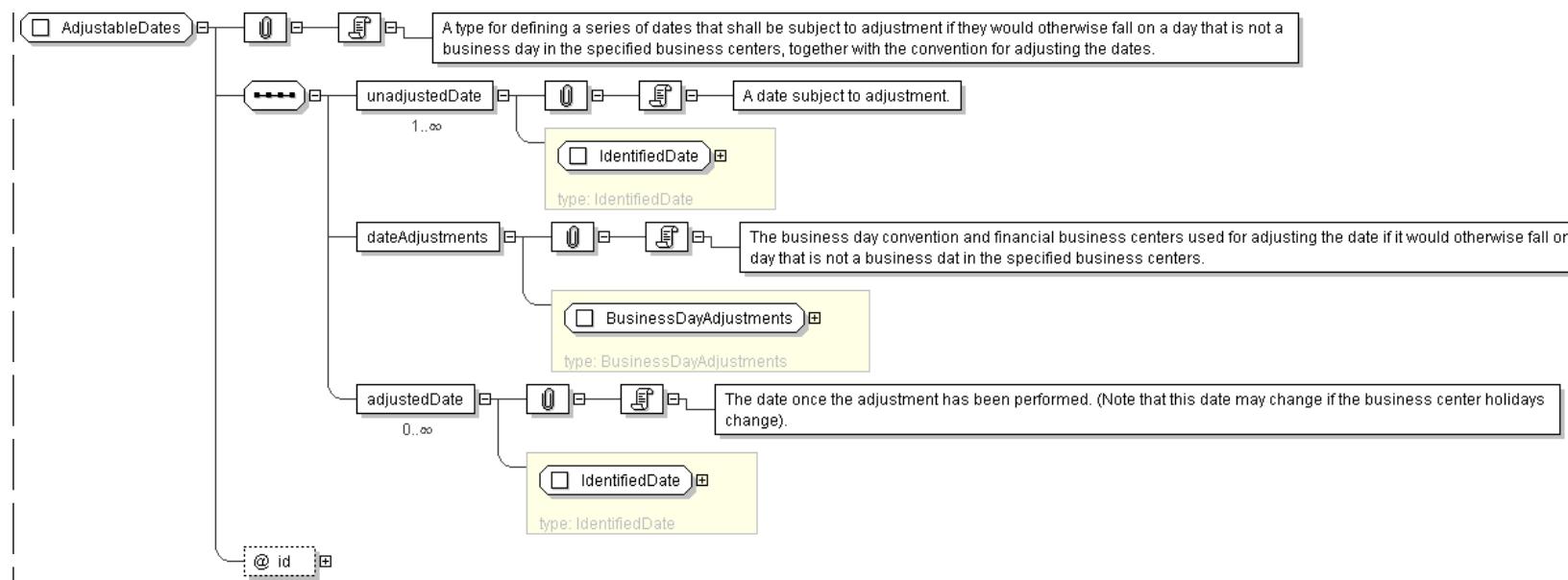
  <dateAdjustments> BusinessDayAdjustments </dateAdjustments> [1]
  'The business day convention and financial business centers used for adjusting the date if
  it would otherwise fall on a day that is not a business dat in the specified business centers.'

  <adjustedDate> IdentifiedDate </adjustedDate> [0..*]
  'The date once the adjustment has been performed. (Note that this date may change if
  the business center holidays change.).'

</...>

```

**Diagram**

**Schema Component Representation**

```

<xsd:complexType name="AdjustableDates">
  <xsd:sequence>
    <xsd:element name="unadjustedDate" type=" IdentifiedDate " maxOccurs="unbounded"/>
    <xsd:element name="dateAdjustments" type=" BusinessDayAdjustments " />
    <xsd:element name="adjustedDate" type=" IdentifiedDate " minOccurs="0" maxOccurs="unbounded"/>
  </xsd:sequence>
  <xsd:attribute name="id" type=" xsd:ID " use="optional"/>
</xsd:complexType>

```

top

**Complex Type: AdjustableDatesOrRelativeDateOffset**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None

<b>Name</b>	AdjustableDatesOrRelativeDateOffset
<b>Abstract</b>	no
<b>Documentation</b>	A type for defining a series of dates, either as a list of adjustable dates, or a as a repeating sequence from a base date

**XML Instance Representation**

```

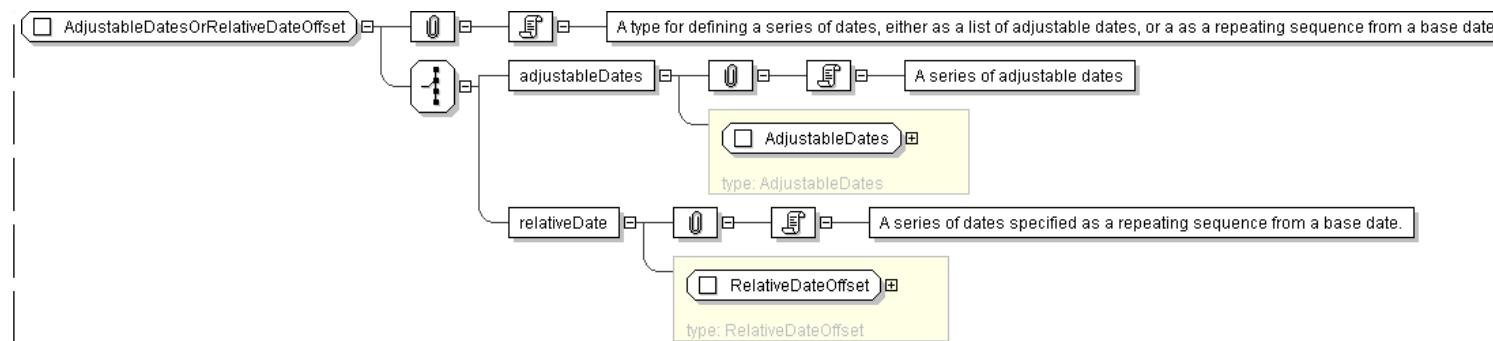
<...>
Start Choice [1]
  <adjustableDates> AdjustableDates </adjustableDates> [1]
    'A series of adjustable dates'

  <relativeDate> RelativeDateOffset </relativeDate> [1]
    'A series of dates specified as a repeating sequence from a base date.'

End Choice
</...>

```

**Diagram**

**Schema Component Representation**

```

<xsd:complexType name="AdjustableDatesOrRelativeDateOffset">
  <xsd:choice>
    <xsd:element name="adjustableDates" type="#AdjustableDates" />
    <xsd:element name="relativeDate" type="#RelativeDateOffset" />
  </xsd:choice>
</xsd:complexType>
  
```

top

**Complex Type: AdjustableOrAdjustedDate**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None
<b>Name</b>	AdjustableOrAdjustedDate
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">Payment</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type 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.

**XML Instance Representation**

```

<...
  id="#_ID [0..1]">
  Start Choice [1]
    <unadjustedDate> IdentifiedDate </unadjustedDate> [1]
    'A date subject to adjustment.'

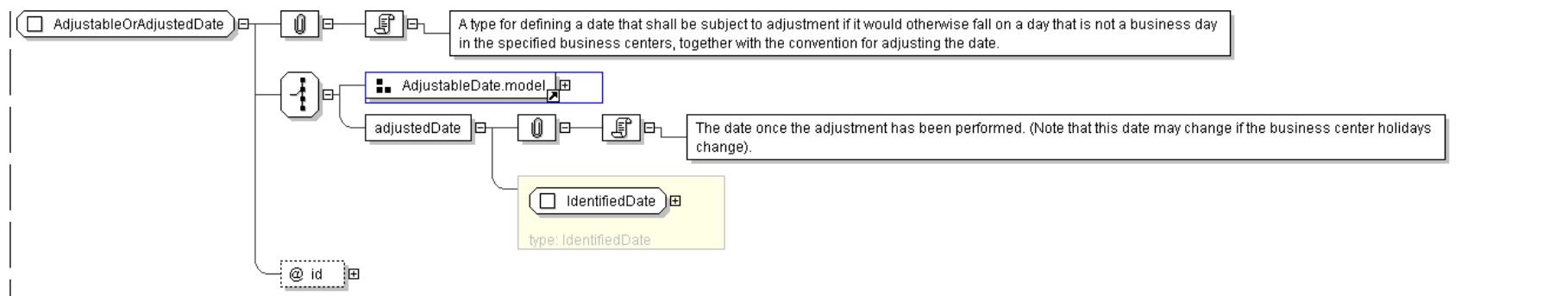
    <dateAdjustments> BusinessDayAdjustments </dateAdjustments> [1]
    'The business day convention and financial business centers used for adjusting the date if
    it would otherwise fall on a day that is not a business date in the specified business centers.'

    <adjustedDate> IdentifiedDate </adjustedDate> [0..1]
    'The date once the adjustment has been performed. (Note that this date may change if
    the business center holidays change).'

    <adjustedDate> IdentifiedDate </adjustedDate> [1]
    'The date once the adjustment has been performed. (Note that this date may change if
    the business center holidays change).'

  End Choice
</...>
  
```

**Diagram**

**Schema Component Representation**

```

<xsd:complexType name="AdjustableOrAdjustedDate">
  <xsd:choice>
    <xsd:group ref=" AdjustableDate.model " />
    <xsd:element name="adjustedDate" type=" IdentifiedDate " />
  </xsd:choice>
  <xsd:attribute name="id" type=" xsd:ID " />
</xsd:complexType>
  
```

[top](#)**Complex Type: AdjustableOrRelativeDate**

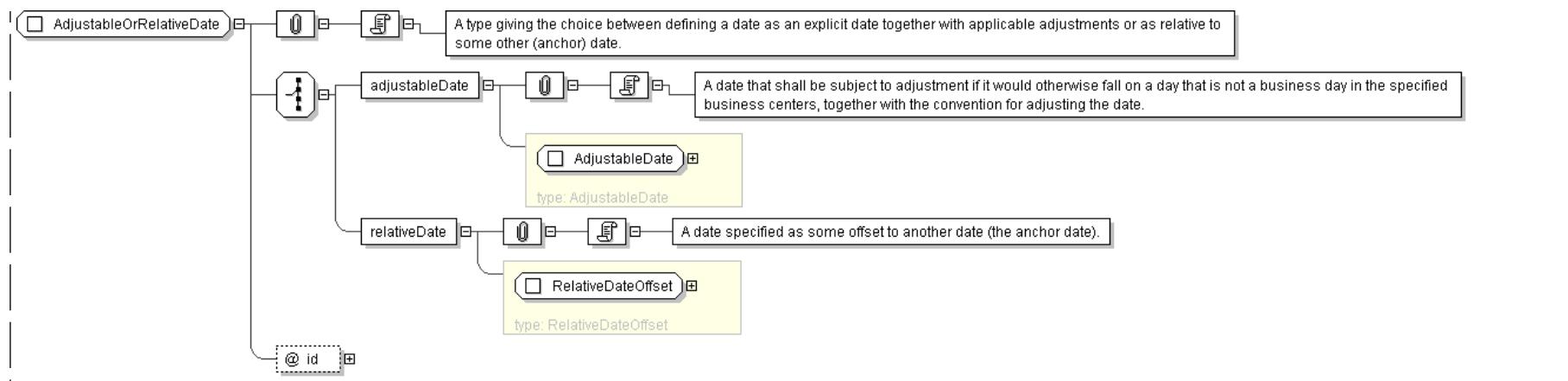
<b>Super-types:</b>	None
<b>Sub-types:</b>	None
<b>Name</b>	AdjustableOrRelativeDate
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">AmericanExercise</a> , Complex Type <a href="#">AmericanExercise</a> , Complex Type <a href="#">EuropeanExercise</a> , Complex Type <a href="#">PaymentBaseExtended</a> , Complex Type <a href="#">PeriodicDates</a> , Complex Type <a href="#">PeriodicDates</a> , Complex Type <a href="#">SharedAmericanExercise</a> , Complex Type <a href="#">SharedAmericanExercise</a> , Complex Type <a href="#">SimplePayment</a> , Complex Type <a href="#">Stub</a> , Complex Type <a href="#">Stub</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type giving the choice between defining a date as an explicit date together with applicable adjustments or as relative to some other (anchor) date.

**XML Instance Representation**

```

<...
  id=" xsd:ID [0..1]">
  Start Choice [1]
    <adjustableDate> AdjustableDate </adjustableDate> [1]
      '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.'
    <relativeDate> RelativeDateOffset </relativeDate> [1]
      'A date specified as some offset to another date (the anchor date).'
  End Choice
</...>
  
```

**Diagram**

**Schema Component Representation**

```

<xsd:complexType name="AdjustableOrRelativeDate">
  <xsd:choice>
    <xsd:element name="adjustableDate" type=" AdjustableDate " />
    <xsd:element name="relativeDate" type=" RelativeDateOffset " />
  </xsd:choice>
  <xsd:attribute name="id" type=" xsd:ID " />
</xsd:complexType>
  
```

top

**Complex Type: AdjustableOrRelativeDates**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None

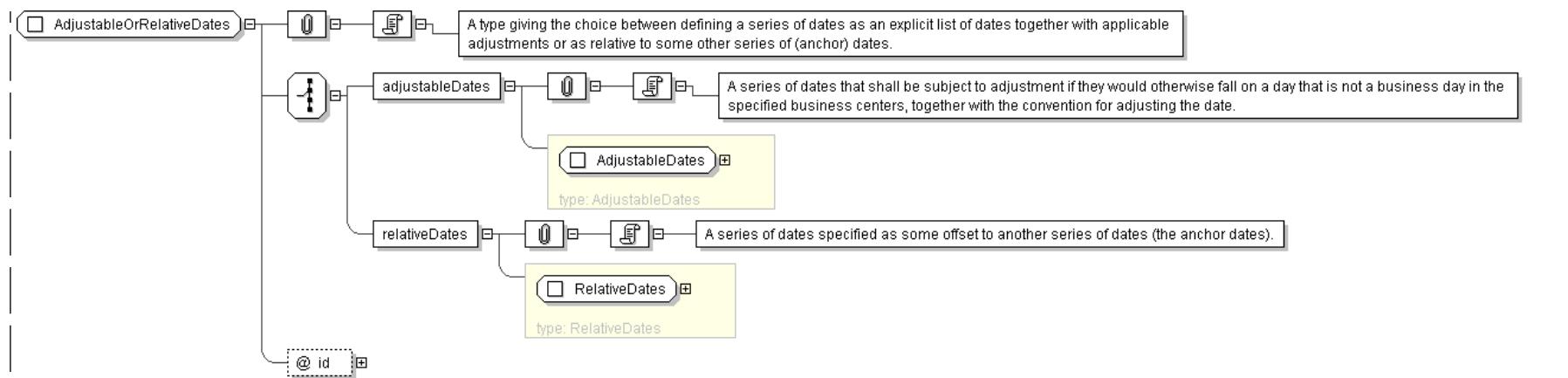
<b>Name</b>	AdjustableOrRelativeDates
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">AmericanExercise</a> , Complex Type <a href="#">BermudaExercise</a> , Complex Type <a href="#">BermudaExercise</a> , Complex Type <a href="#">EuropeanExercise</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type giving 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.

**XML Instance Representation**

```

<...>
  id=" xsd:ID [0..1]">
  Start Choice [1]
    <adjustableDates> AdjustableDates </adjustableDates> [1]
      '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.'
    <relativeDates> RelativeDates </relativeDates> [1]
      'A series of dates specified as some offset to another series of dates (the anchor dates).'
  End Choice
</...>
  
```

**Diagram**

**Schema Component Representation**

```

<xsd:complexType name="AdjustableOrRelativeDates">
  <xsd:choice>
    <xsd:element name="adjustableDates" type=" AdjustableDates " />
    <xsd:element name="relativeDates" type=" RelativeDates " />
  </xsd:choice>
  <xsd:attribute name="id" type=" xsd:ID " />
</xsd:complexType>
  
```

top

**Complex Type: AdjustableRelativeOrPeriodicDates**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None

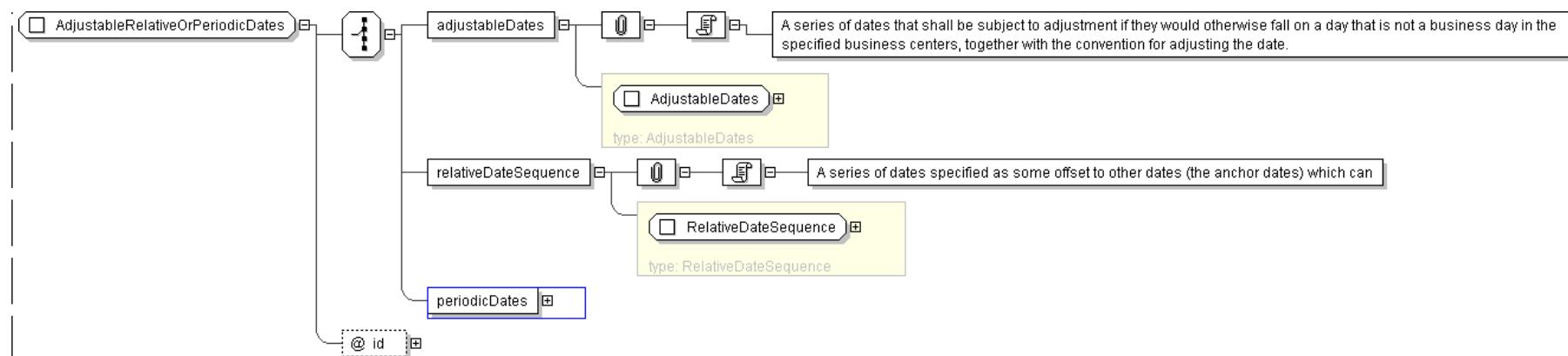
<b>Name</b>	AdjustableRelativeOrPeriodicDates
<b>Abstract</b>	no

**XML Instance Representation**

```

<...
  id=" xsd:ID [0..1]">
Start Choice [1]
  <adjustableDates> AdjustableDates </adjustableDates> [1]
    '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.'
  <relativeDateSequence> RelativeDateSequence </relativeDateSequence> [1]
    'A series of dates specified as some offset to other dates (the anchor dates) which can'
  <periodicDates> PeriodicDates </periodicDates> [1]
End Choice
</...>
  
```

**Diagram**

**Schema Component Representation**

```
<xsd:complexType name="AdjustableRelativeOrPeriodicDates">
  <xsd:choice>
    <xsd:element name="adjustableDates" type=" AdjustableDates " />
    <xsd:element name="relativeDateSequence" type=" RelativeDateSequence " />
    <xsd:element name="periodicDates" type=" PeriodicDates " />
  </xsd:choice>
  <xsd:attribute name="id" type=" xsd:ID " />
</xsd:complexType>
```

[top](#)**Complex Type: AdjustableRelativeOrPeriodicDates2**

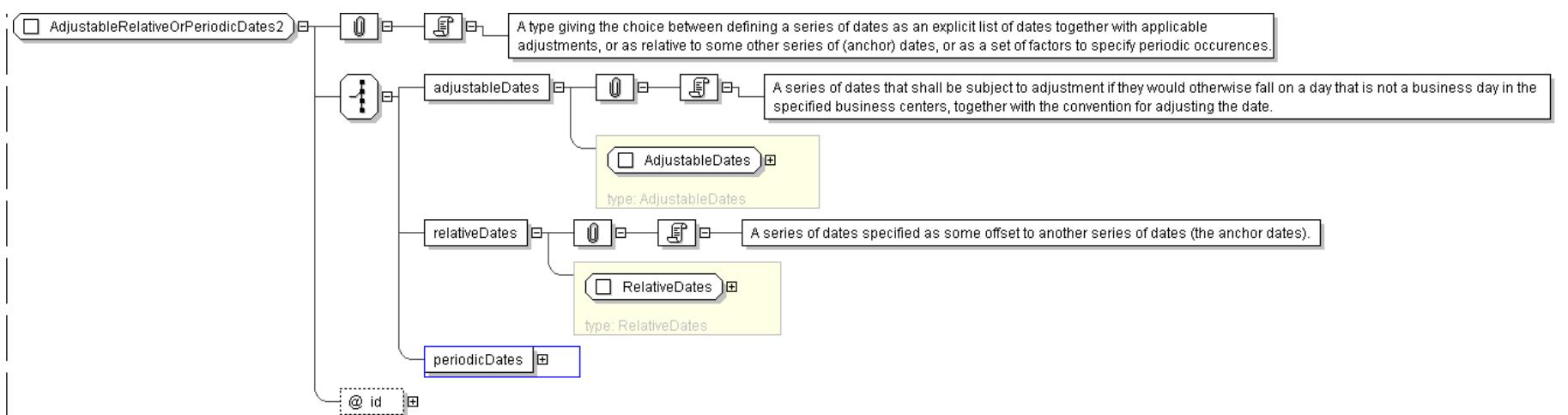
<b>Super-types:</b>	None
<b>Sub-types:</b>	None

<b>Name</b>	AdjustableRelativeOrPeriodicDates2
<b>Abstract</b>	no
<b>Documentation</b>	A type giving 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, or as a set of factors to specify periodic occurrences.

**XML Instance Representation**

```
<...
  id=" xsd:ID [0..1]">
Start Choice [1]
  <adjustableDates> AdjustableDates </adjustableDates> [1]
    '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.'
  <relativeDates> RelativeDates </relativeDates> [1]
    'A series of dates specified as some offset to another series of dates (the anchor dates).'
  <periodicDates> PeriodicDates </periodicDates> [1]
End Choice
</...>
```

**Diagram**



## Schema Component Representation

```
<xsd:complexType name="AdjustableRelativeOrPeriodicDates2">
  <xsd:choice>
    <xsd:element name="adjustableDates" type=" AdjustableDates " />
    <xsd:element name="relativeDates" type=" RelativeDates " />
    <xsd:element name="periodicDates" type=" PeriodicDates " />
  </xsd:choice>
  <xsd:attribute name="id" type=" xsd:ID " />
</xsd:complexType>
```

top

**Complex Type:** `AdjustedRelativeDateOffset`

**Super-types:** [Period](#) < [Offset](#) (by extension) < [RelativeDateOffset](#) (by extension) < **AdjustedRelativeDateOffset** (by extension)  
**Sub-types:** None

<b>Name</b>	AdjustedRelativeDateOffset
<b>Abstract</b>	no
<b>Documentation</b>	A type defining a date (referred to as the derived date) as a relative offset from another date (referred to as the anchor date) plus optional date adjustments.

## XML Instance Representation

```
<...  
id=" xsd:ID [0..1]">  
<periodMultipliers> xsd:integer </periodMultiplier> [1]  
'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.'  
  
<period> PeriodEnum </period> [1]  
'A time period, e.g. a day, week, month or year of the stream. If the periodMultiplier value  
is 0 (zero) then period must contain the value D (day).'  
  
<dayType> DayTypeEnum </dayType> [0..1]  
'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.'  
  
<businessDayConvention> BusinessDayConventionEnum </businessDayConvention> [1]
```

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

Start Group: [BusinessCentersOrReference.model](#) [0..1]

Start Choice [1]

<businessCentersReference> BusinessCentersReference </businessCentersReference> [1]

'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.'

<businessCenters> BusinessCenters </businessCenters> [1]

End Choice

End Group: [BusinessCentersOrReference.model](#)

<dateRelativeTo> DateReference </dateRelativeTo> [1]

'Specifies the anchor as an href attribute. The href attribute value is a pointer style reference to the element or component elsewhere in the document where the anchor date is defined.'

<adjustedDate> IdentifiedDate </adjustedDate> [0..1]

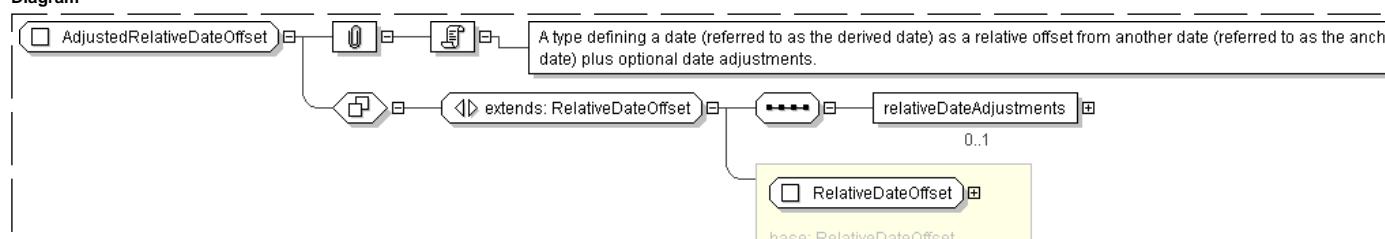
'The date once the adjustment has been performed. (Note that this date may change if the business center holidays change).'

<relativeDateAdjustments> BusinessDayAdjustments </relativeDateAdjustments> [0..1]

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

</...>

#### Diagram



#### Schema Component Representation

```

<xsd:complexType name="AdjustedRelativeDateOffset">
  <xsd:complexContent>
    <xsd:extension base=" RelativeDateOffset ">
      <xsd:sequence>
        <xsd:element name="relativeDateAdjustments" type=" BusinessDayAdjustments " minOccurs="0" />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
  
```

top

## Complex Type: [AgreementType](#)

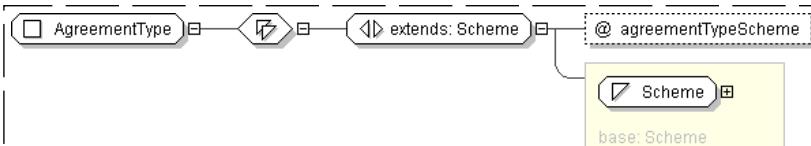
Super-types:	<a href="#">xsd:normalizedString</a> < <a href="#">Scheme</a> (by restriction) < <b>AgreementType</b> (by extension)
--------------	--

Sub-types:	None
------------	------

Name	AgreementType
Used by (from the same schema document)	Complex Type <a href="#">GenericAgreement</a>
Abstract	no

**XML Instance Representation**

```
<...  
  agreementTypeScheme=" xsd:anyURI [0..1]">  
  Scheme  
</...>
```

**Diagram****Schema Component Representation**

```
<xsd:complexType name="AgreementType">  
  <xsd:simpleContent>  
    <xsd:extension base=" Scheme ">  
      <xsd:attribute name="agreementTypeScheme" type=" xsd:anyURI " />  
    </xsd:extension>  
  </xsd:simpleContent>  
</xsd:complexType>
```

[top](#)**Complex Type: AgreementVersion**

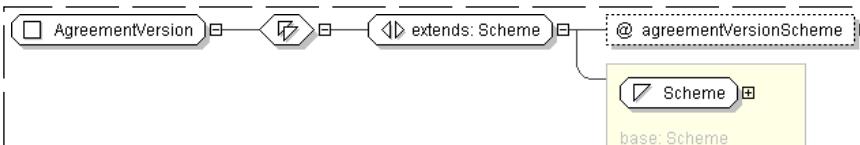
**Super-types:** xsd:normalizedString < [Scheme](#) (by restriction) < **AgreementVersion** (by extension)

**Sub-types:** None

<b>Name</b>	AgreementVersion
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">GenericAgreement</a>
<b>Abstract</b>	no

**XML Instance Representation**

```
<...  
  agreementVersionScheme=" xsd:anyURI [0..1]">  
  Scheme  
</...>
```

**Diagram****Schema Component Representation**

```
<xsd:complexType name="AgreementVersion">  
  <xsd:simpleContent>  
    <xsd:extension base=" Scheme ">  
      <xsd:attribute name="agreementVersionScheme" type=" xsd:anyURI " />  
    </xsd:extension>  
  </xsd:simpleContent>  
</xsd:complexType>
```

[top](#)

## Complex Type: AmericanExercise

Super-types: [Exercise](#) < **AmericanExercise** (by extension)

Sub-types:  
None

<b>Name</b>	AmericanExercise
<b>Used by (from the same schema document)</b>	Element <a href="#">americanExercise</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type 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.

## XML Instance Representation

```
<...>
<id=" xsd:ID [0..1]">
<commencementDate> AdjustableOrRelativeDate </commencementDate> [1]
'The first day of the exercise period for an American style option.'

<expirationDate> AdjustableOrRelativeDate </expirationDate> [1]
'The last day within an exercise period for an American style option. For a European
style option it is the only day within the exercise period.'

<relevantUnderlyingDate> AdjustableOrRelativeDates </relevantUnderlyingDate> [0..1]
'The day on the underlying set by the exercise of an option. What this date is depends on
the option (e.g. in a swaption it is the effective date, in an extendible/cancelable
provision it is the termination date).'

<earliestExerciseTime> BusinessCenterTime </earliestExerciseTime> [1]
'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 Bermuda
style option the commencement date to, and including, the expiration date , in the case of
an American option.'

<latestExerciseTime> BusinessCenterTime </latestExerciseTime> [0..1]
'For a Bermuda or American style option, the latest time on an exercise business day
(excluding the expiration date) within the exercise period that notice can be given by
the 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.'

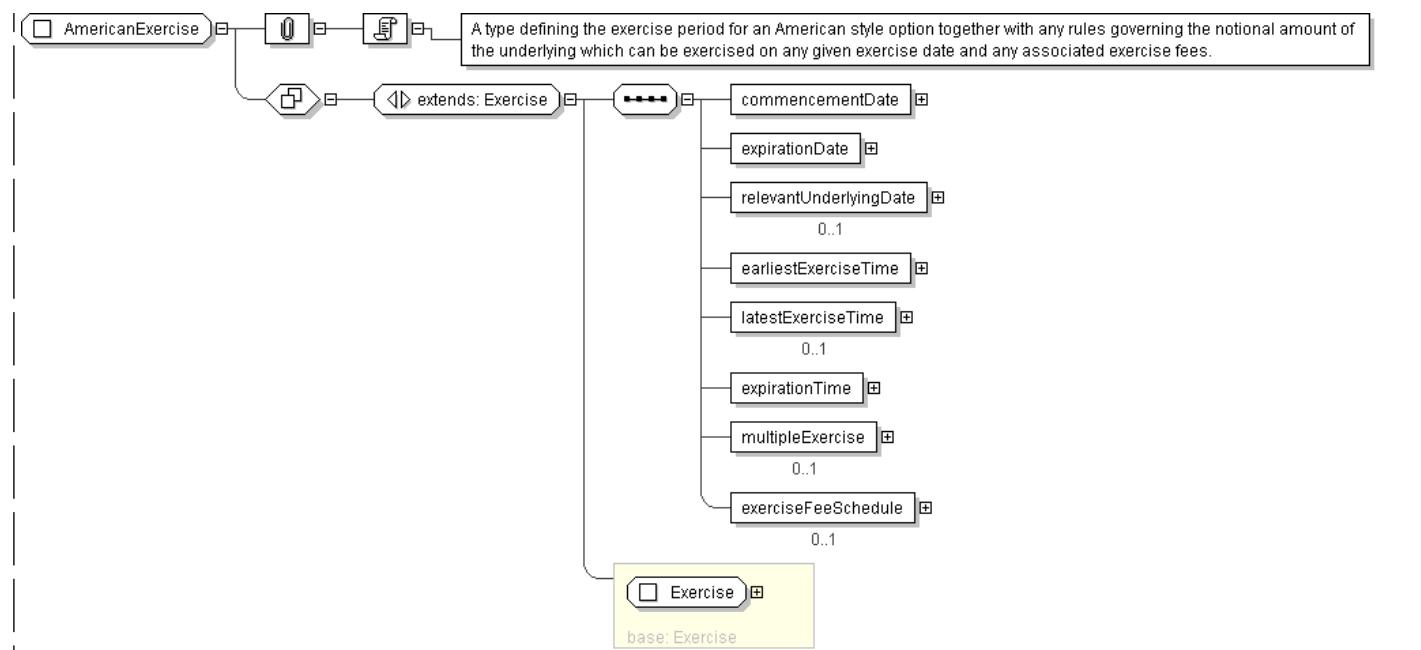
<expirationTime> BusinessCenterTime </expirationTime> [1]
'The latest time for exercise on expirationDate.'

<multipleExercise> MultipleExercise </multipleExercise> [0..1]
'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 intergral multiple of, the integral multiple amount.'

<exerciseFeeSchedule> ExerciseFeeSchedule </exerciseFeeSchedule> [0..1]
'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.'

</...>
```

## Diagram



#### Schema Component Representation

```

<xsd:complexType name="AmericanExercise">
  <xsd:complexContent>
    <xsd:extension base=" Exercise ">
      <xsd:sequence>
        <xsd:element name="commencementDate" type=" AdjustableOrRelativeDate " />
        <xsd:element name="expirationDate" type=" AdjustableOrRelativeDate " />
        <xsd:element name="relevantUnderlyingDate" type=" AdjustableOrRelativeDates " minOccurs="0"/>
        <xsd:element name="earliestExerciseTime" type=" BusinessCenterTime " />
        <xsd:element name="latestExerciseTime" type=" BusinessCenterTime " minOccurs="0"/>
        <xsd:element name="expirationTime" type=" BusinessCenterTime " />
        <xsd:element name="multipleExercise" type=" MultipleExercise " minOccurs="0"/>
        <xsd:element name="exerciseFeeSchedule" type=" ExerciseFeeSchedule " minOccurs="0"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
  
```

top

#### Complex Type: AmountReference

Super-types: [Reference](#) < **AmountReference** (by extension)

Sub-types: None

Name	AmountReference
------	-----------------

Abstract	no
----------	----

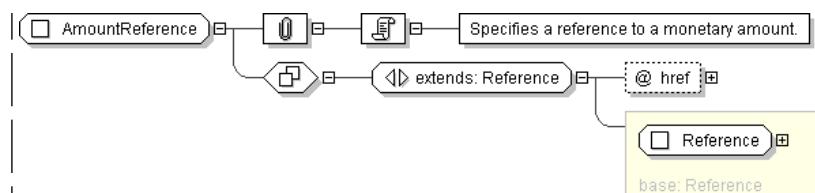
Documentation	Specifies a reference to a monetary amount.
---------------	---

#### XML Instance Representation

```

<...>
  <!-- href=" xsd:IDREF [1]" -->
  
```

#### Diagram

**Schema Component Representation**

```

<xsd:complexType name="AmountReference">
  <xsd:complexContent>
    <xsd:extension base=" Reference ">
      <xsd:attribute name="href" type=" xsd:IDREF " use="required"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
  
```

top

**Complex Type: AmountSchedule**

<b>Super-types:</b>	<a href="#">Schedule</a> < <b>AmountSchedule</b> (by extension)
<b>Sub-types:</b>	None

<b>Name</b>	AmountSchedule
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">ExerciseFeeSchedule</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type defining a currency amount or a currency amount schedule.

**XML Instance Representation**

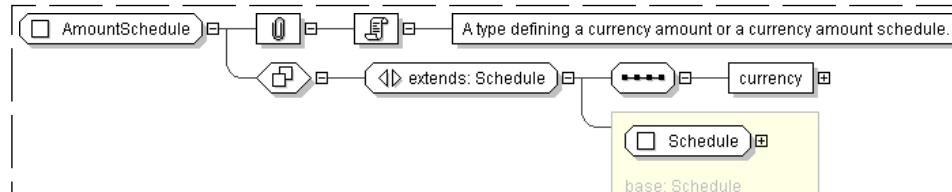
```

<...
  id=" xsd:ID [0..1]">
  <initialValue> xsd:decimal </initialValue> [1]
  'The initial rate or amount, as the case may be. An initial rate of 5% would be represented
  as 0.05.'

  <step> Step </step> [0..*]
  '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.'

  <currency> Currency </currency> [1]
  'The currency in which an amount is denominated.'

</...>
  
```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="AmountSchedule">
  <xsd:complexContent>
  
```

```

<xsd:extension base=" Schedule ">
  <xsd:sequence>
    <xsd:element name="currency" type=" Currency " />
  </xsd:sequence>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>

```

[top](#)

## Complex Type: AssetClass

**Super-types:** [xsd:normalizedString](#) < [Scheme](#) (by restriction) < **AssetClass** (by extension)  
**Sub-types:** None

Name	AssetClass
------	------------

Abstract	no
----------	----

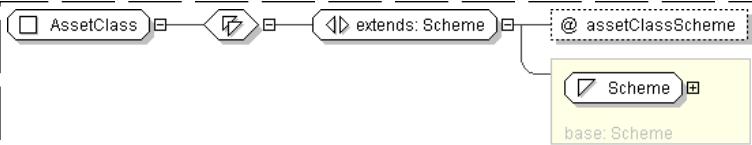
### XML Instance Representation

```

<...
  assetClassScheme=" xsd:anyURI [0..1]">
  Scheme
</...>

```

### Diagram



### Schema Component Representation

```

<xsd:complexType name="AssetClass">
  <xsd:simpleContent>
    <xsd:extension base=" Scheme ">
      <xsd:attribute name="assetClassScheme" type=" xsd:anyURI " default="http://www.fpml.org/
        coding-scheme/asset-class-simple"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>

```

[top](#)

## Complex Type: AutomaticExercise

**Super-types:** None  
**Sub-types:** None

Name	AutomaticExercise
------	-------------------

Used by (from the same schema document)	Complex Type <a href="#">ExerciseProcedure</a>
---	--

Abstract	no
----------	----

Documentation	A type to define automatic exercise of a swap option. With automatic exercise the option is deemed to have exercised if it is in the money by more than the threshold amount on the exercise date.
---------------	--

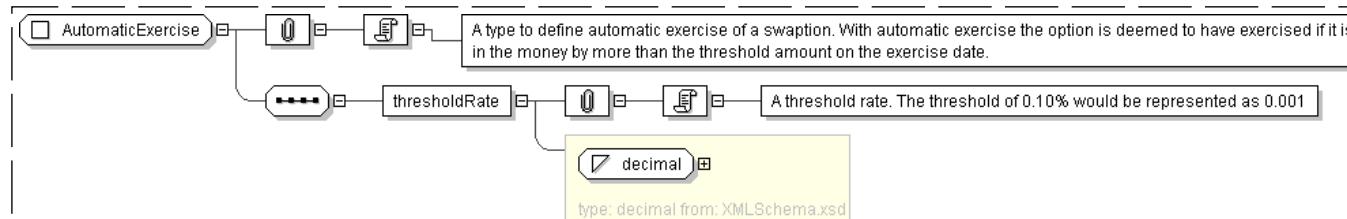
### XML Instance Representation

```

<...>
<thresholdRate> xsd:decimal </thresholdRate> [1]
' A threshold rate. The threshold of 0.10% would be represented as 0.001'

```

&lt; / . . . &gt;

**Diagram****Schema Component Representation**

```

<xsd:complexType name="AutomaticExercise">
  <xsd:sequence>
    <xsd:element name="thresholdRate" type="xsd:decimal" />
  </xsd:sequence>
</xsd:complexType>

```

[top](#)**Complex Type: AverageDailyTradingVolumeLimit**

Super-types:

None

Sub-types:

None

Name	AverageDailyTradingVolumeLimit
------	--------------------------------

Abstract	no
----------	----

Documentation	To indicate the limitation percentage and limitation period.
---------------	--

**XML Instance Representation**

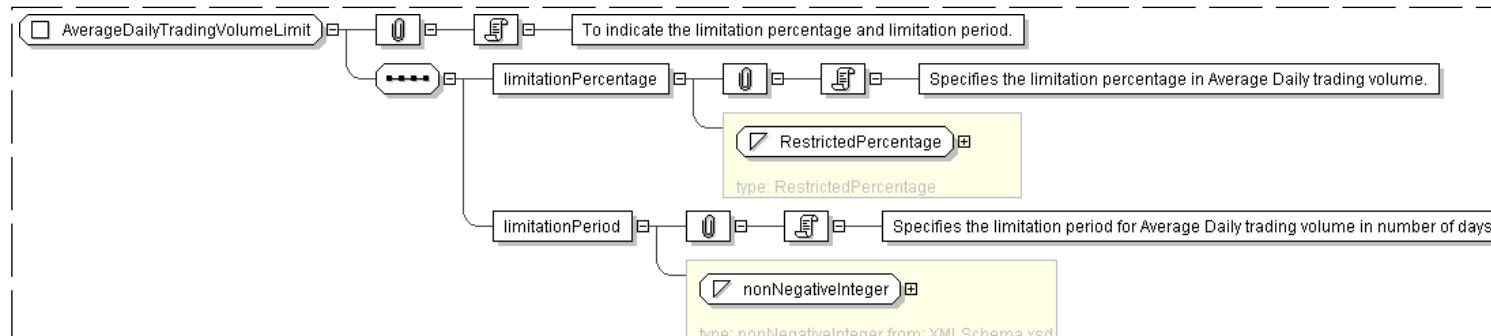
```

<...>
  <limitationPercentage> RestrictedPercentage </limitationPercentage> [1]
  'Specifies the limitation percentage in Average Daily trading volume.'

  <limitationPeriod> xsd:nonNegativeInteger </limitationPeriod> [1]
  'Specifies the limitation period for Average Daily trading volume in number of days.'

</...>

```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="AverageDailyTradingVolumeLimit">
  <xsd:sequence>

```

```

<xsd:element name="limitationPercentage" type="#RestrictedPercentage" />
<xsd:element name="limitationPeriod" type="xsd:nonNegativeInteger" />
</xsd:sequence>
</xsd:complexType>

```

## Complex Type: Beneficiary

**Super-types:** None  
**Sub-types:** None

<b>Name</b>	Beneficiary
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">SettlementInstruction</a> , Complex Type <a href="#">SettlementInstruction</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type defining the beneficiary of the funds.

### XML Instance Representation

```

<...>
Start Choice [1]
<routingIds> RoutingIds </routingIds> [1]
'A set of unique identifiers for a party, each one identifying the party within a
payment system. The assumption is that each party will not have more than one identifier
within the same payment system.'

<routingExplicitDetails> RoutingExplicitDetails </routingExplicitDetails> [1]
'A set of details that is used to identify a party involved in the routing of a payment
when the party does not have a code that identifies it within one of the recognized
payment systems.'

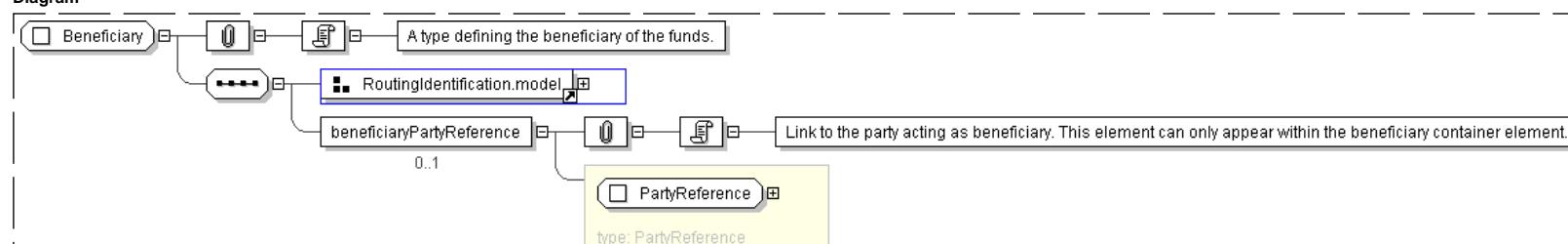
<routingIdsAndExplicitDetails> RoutingIdsAndExplicitDetails </routingIdsAndExplicitDetails> [1]
'A combination of coded payment system identifiers and details for physical addressing for
a party involved in the routing of a payment.'

End Choice
<beneficiaryPartyReference> PartyReference </beneficiaryPartyReference> [0..1]
'Link to the party acting as beneficiary. This element can only appear within the
beneficiary container element.'

</...>

```

### Diagram



### Schema Component Representation

```

<xsd:complexType name="Beneficiary">
<xsd:sequence>
  <xsd:group ref="#RoutingIdentification.model" />
  <xsd:element name="beneficiaryPartyReference" type="PartyReference" minOccurs="0"/>
</xsd:sequence>
</xsd:complexType>

```

**Complex Type: BermudaExercise**

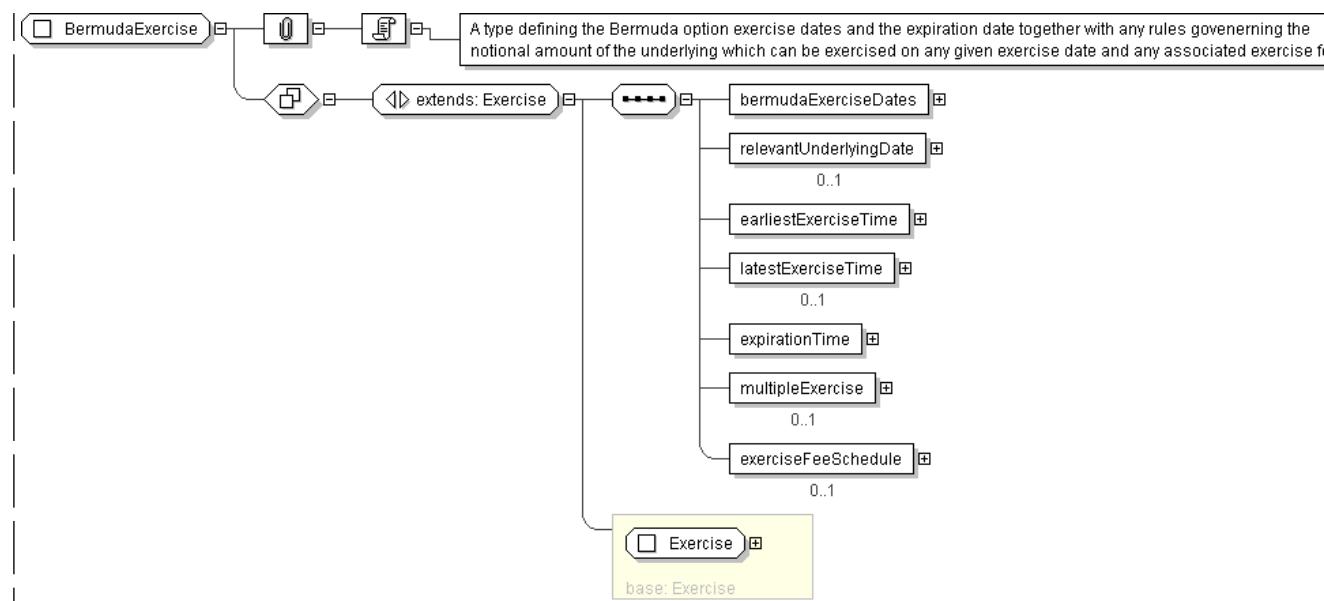
Super-types:	<a href="#">Exercise</a> < <b>BermudaExercise</b> (by extension)
Sub-types:	None
Name	BermudaExercise
Used by (from the same schema document)	Element <a href="#">bermudaExercise</a>
Abstract	no
Documentation	A type defining the Bermuda option exercise dates and the expiration date together with any rules governing the notional amount of the underlying which can be exercised on any given exercise date and any associated exercise fee.

**XML Instance Representation**

```

<...>
<id="xsd:ID [0..1]">
<bermudaExerciseDates> AdjustableOrRelativeDates </bermudaExerciseDates> [1]
'The dates that define the Bermuda option exercise dates and the expiration date. The last specified 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> AdjustableOrRelativeDates </relevantUnderlyingDate> [0..1]
'The day on the underlying set by the exercise of an option. What this date is depends on the option (e.g. in a swaption it is the effective date, in an extendible/cancelable provision it is the termination date).'
<earliestExerciseTime> BusinessCenterTime </earliestExerciseTime> [1]
'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 Bermuda style option the commencement date to, and including, the expiration date , in the case of an American option.'
<latestExerciseTime> BusinessCenterTime </latestExerciseTime> [0..1]
'For a Bermuda or American style option, the latest time on an exercise business day (excluding the expiration date) within the exercise period that notice can be given by the 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> BusinessCenterTime </expirationTime> [1]
'The latest time for exercise on expirationDate.'
<multipleExercise> MultipleExercise </multipleExercise> [0..1]
'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 intergral multiple of, the integral multiple amount.'
<exerciseFeeSchedule> ExerciseFeeSchedule </exerciseFeeSchedule> [0..1]
'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.'
</...>
```

**Diagram**



top

## Complex Type: BrokerConfirmation

Super-types: None  
 Sub-types: None

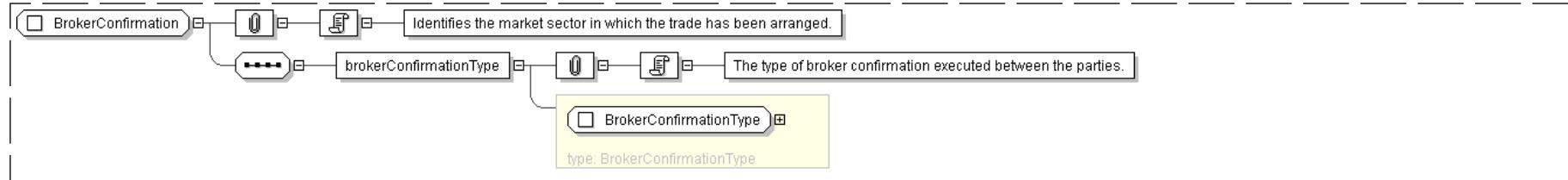
Name	BrokerConfirmation
Used by (from the same schema document)	Complex Type <a href="#">Documentation</a>
Abstract	no
Documentation	Identifies the market sector in which the trade has been arranged.

### XML Instance Representation

```

<...>
<brokerConfirmationType> BrokerConfirmationType </brokerConfirmationType> [1]
' The type of broker confirmation executed between the parties.'
</...>

```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="BrokerConfirmation">
  <xsd:sequence>
    <xsd:element name="brokerConfirmationType" type="<a href="#" type="BrokerConfirmationType">BrokerConfirmationType</a>" />
  </xsd:sequence>
</xsd:complexType>
  
```

top

**Complex Type: BrokerConfirmationType**

**Super-types:** [xsd:normalizedString](#) < [Scheme](#) (by restriction) < **BrokerConfirmationType** (by extension)

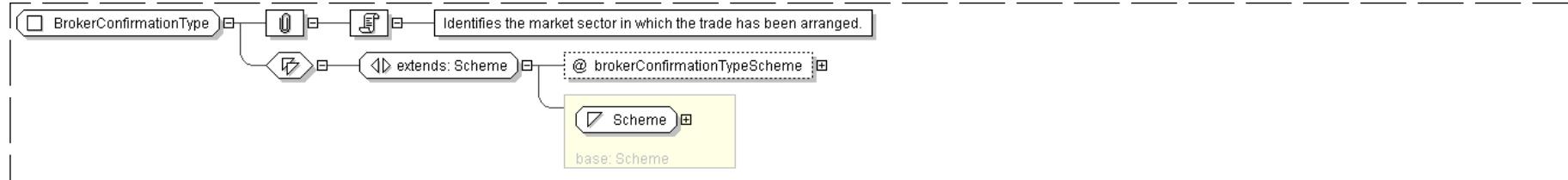
**Sub-types:** None

<b>Name</b>	BrokerConfirmationType
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">BrokerConfirmation</a>
<b>Abstract</b>	no
<b>Documentation</b>	Identifies the market sector in which the trade has been arranged.

**XML Instance Representation**

```

<...
  brokerConfirmationTypeScheme="<a href="#" type="xsd:anyURI [0..1]">
  Scheme
</a>
</...>
  
```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="BrokerConfirmationType">
  <xsd:simpleContent>
    <xsd:extension base="<a href="#" type="Scheme">Scheme</a>">
      <xsd:attribute name="brokerConfirmationTypeScheme" type="<a href="#" type="xsd:anyURI">xsd:anyURI</a>" default="http://www.
        fpml.org/coding-scheme/broker-confirmation-type"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
  
```

top

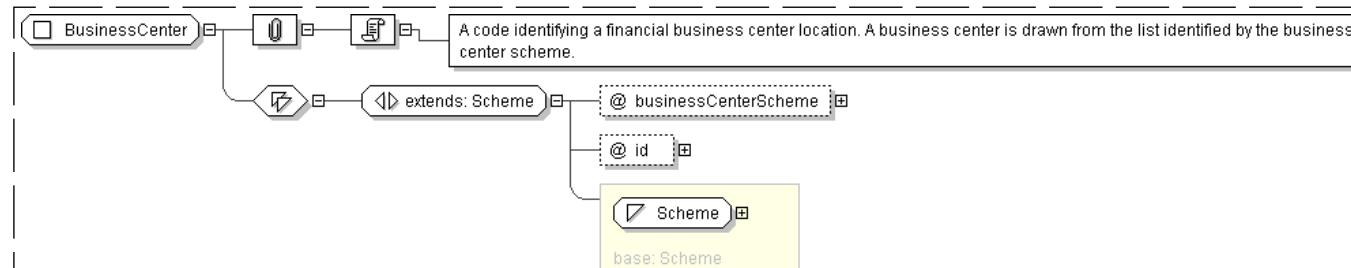
**Complex Type: BusinessCenter**

Super-types:	<a href="#">xsd:normalizedString</a> < <a href="#">Scheme</a> (by restriction) < <b>BusinessCenter</b> (by extension)
Sub-types:	None

Name	BusinessCenter
Used by (from the same schema document)	Complex Type <a href="#">BusinessCenters</a> , Complex Type <a href="#">BusinessCenterTime</a> , Complex Type <a href="#">ExerciseNotice</a>
Abstract	no
Documentation	A code identifying a financial business center location. A business center is drawn from the list identified by the business center scheme.

**XML Instance Representation**

```
<...>
  <@ businessCenterScheme=" xsd:anyURI [0..1]">
  <@ id=" xsd:ID [0..1]">
  <@ Scheme>
</...>
```

**Diagram****Schema Component Representation**

```
<xsd:complexType name="BusinessCenter">
  <xsd:simpleContent>
    <xsd:extension base=" Scheme ">
      <xsd:attribute name="businessCenterScheme" type=" xsd:anyURI " default="http://www.fpml.org/coding-scheme/business-center"/>
      <xsd:attribute name="id" type=" xsd:ID ">
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
```

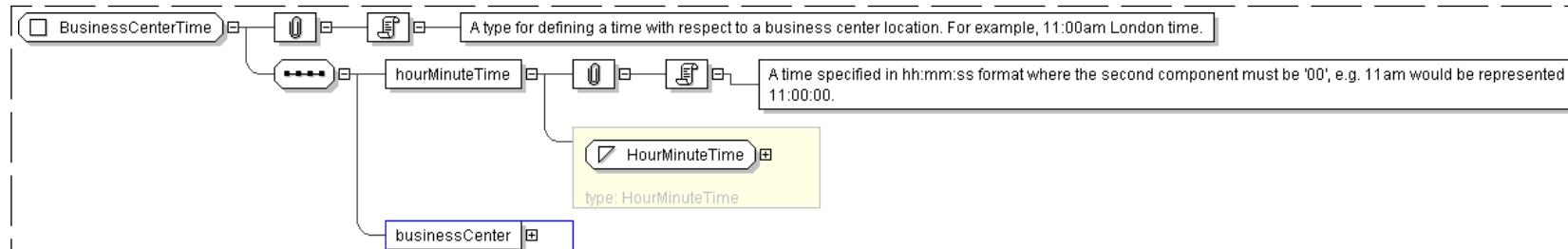
[top](#)**Complex Type: BusinessCenterTime**

Super-types:	None
Sub-types:	None

Name	BusinessCenterTime
Used by (from the same schema document)	Complex Type <a href="#">AmericanExercise</a> , Complex Type <a href="#">AmericanExercise</a> , Complex Type <a href="#">AmericanExercise</a> , Complex Type <a href="#">BermudaExercise</a> , Complex Type <a href="#">BermudaExercise</a> , Complex Type <a href="#">BermudaExercise</a> , Complex Type <a href="#">EuropeanExercise</a> , Complex Type <a href="#">EuropeanExercise</a> , Complex Type <a href="#">FxSpotRateSource</a> , Complex Type <a href="#">SharedAmericanExercise</a>
Abstract	no
Documentation	A type for defining a time with respect to a business center location. For example, 11:00am London time.

**XML Instance Representation**

```
<...>
  <hourMinuteTime> HourMinuteTime </hourMinuteTime> [1]
  '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> BusinessCenter </businessCenter> [1]
</...>
```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="BusinessCenterTime">
  <xsd:sequence>
    <xsd:element name="hourMinuteTime" type=" HourMinuteTime " />
    <xsd:element name="businessCenter" type=" BusinessCenter " />
  </xsd:sequence>
</xsd:complexType>
  
```

top

**Complex Type: BusinessCenters**

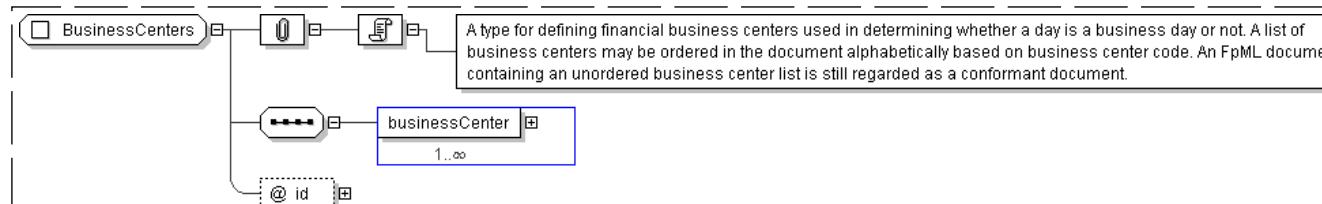
<b>Super-types:</b>	None
<b>Sub-types:</b>	None

<b>Name</b>	BusinessCenters
<b>Used by (from the same schema document)</b>	Model Group <a href="#">BusinessCentersOrReference.model</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type for defining financial business centers used in determining whether a day is a business day or not. 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.

**XML Instance Representation**

```

<...
  id=" xsd:ID [0..1]">
  <businessCenter> BusinessCenter </businessCenter> [1..*]
</...>
  
```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="BusinessCenters">
  <xsd:sequence>
    <xsd:element name="businessCenter" type=" BusinessCenter " maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:attribute name="id" type=" xsd:ID " />
</xsd:complexType>
  
```

## Complex Type: BusinessCentersReference

Super-types:

[Reference](#) < **BusinessCentersReference** (by extension)

Sub-types:

None

Name

BusinessCentersReference

Used by (from the same schema document)

Model Group [BusinessCentersOrReference.model](#)

Abstract

no

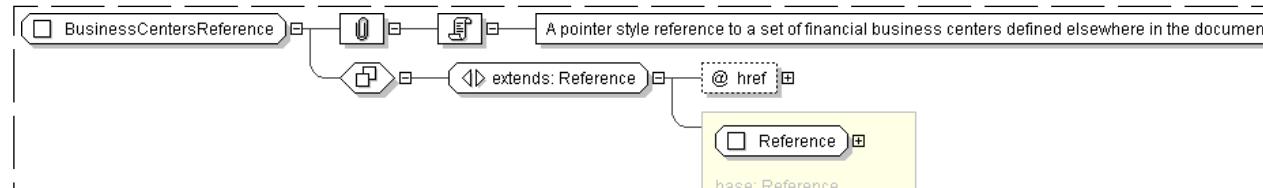
Documentation

A pointer style reference to a set of financial business centers defined elsewhere in the document.

### XML Instance Representation

```
<...>
  href=" xsd:IDREF [1]" />
```

### Diagram



### Schema Component Representation

```

<xsd:complexType name="BusinessCentersReference">
  <xsd:complexContent>
    <xsd:extension base=" Reference ">
      <xsd:attribute name="href" type=" xsd:IDREF " use="required" reference="BusinessCenters" />
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
  
```

## Complex Type: BusinessDateRange

Super-types:

[DateRange](#) < **BusinessDateRange** (by extension)

Sub-types:

None

Name

BusinessDateRange

Abstract

no

Documentation

A type 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 centers to be counted in the range.

### XML Instance Representation

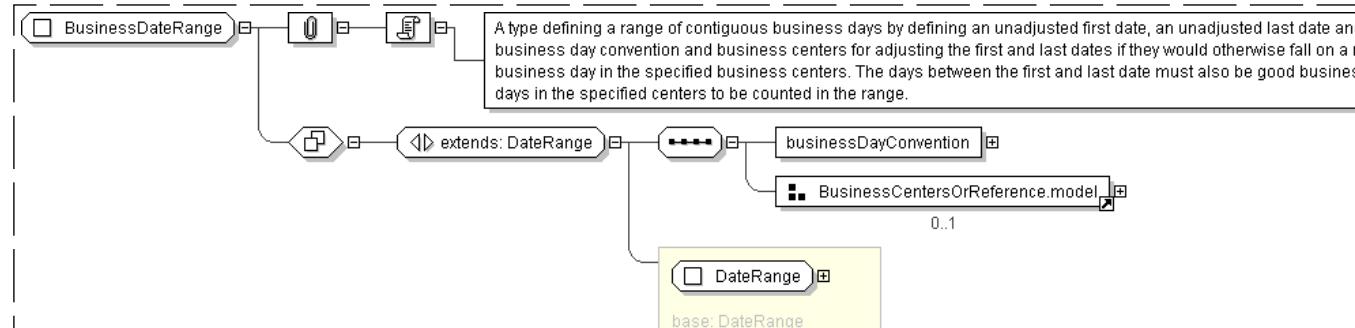
```

<...>
  <unadjustedFirstDate> xsd:date </unadjustedFirstDate> [1]
  'The first date of a date range.'
  <unadjustedLastDate> xsd:date </unadjustedLastDate> [1]
  'The last date of a date range.'
  <businessDayConvention> BusinessDayConventionEnum </businessDayConvention> [1]
  'The convention for adjusting a date if it would otherwise fall on a day that is not a business day.'
  
```

```

Start Group: BusinessCentersOrReference.model [0..1]
Start Choice [1]
  <businessCentersReference> BusinessCentersReference </businessCentersReference> [1]
    '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.'
  <businessCenters> BusinessCenters </businessCenters> [1]
End Choice
End Group: BusinessCentersOrReference.model
</...>

```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="BusinessDateRange">
  <xsd:complexContent>
    <xsd:extension base=" DateRange ">
      <xsd:sequence>
        <xsd;element name="businessDayConvention" type=" BusinessDayConventionEnum " />
        <xsd:group ref=" BusinessCentersOrReference.model " minOccurs="0" />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

[top](#)**Complex Type: BusinessDayAdjustments**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None

<b>Name</b>	BusinessDayAdjustments
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">AdjustableDate2</a> , Complex Type <a href="#">AdjustableDates</a> , Complex Type <a href="#">AdjustedRelativeDateOffset</a> , Complex Type <a href="#">PeriodicDates</a> , Model Group <a href="#">AdjustableDate.model</a>

<b>Abstract</b>	no
<b>Documentation</b>	A type 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.

**XML Instance Representation**

```

<...
  id=" xsd:ID [0..1]">
  <businessDayConvention> BusinessDayConventionEnum </businessDayConvention> [1]
  'The convention for adjusting a date if it would otherwise fall on a day that is not a
  business day.'

```

Start Group: [BusinessCentersOrReference.model](#) [0..1]

| Start Choice [1]  
 <businessCentersReference> BusinessCentersReference </businessCentersReference> [1]

'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.'

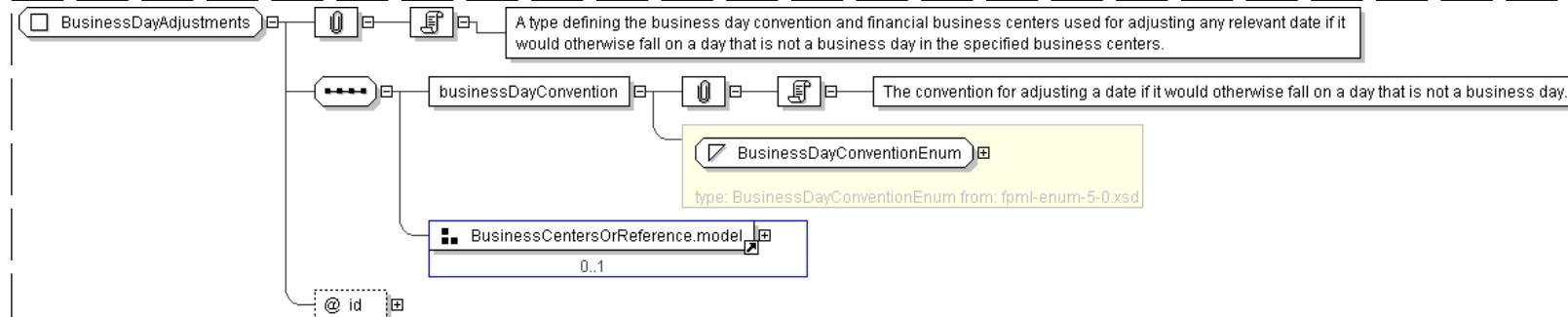
<businessCenters> BusinessCenters </businessCenters> [1]

End Choice

End Group: BusinessCentersOrReference.model

</...>

#### Diagram



#### Schema Component Representation

```

<xsd:complexType name="BusinessDayAdjustments">
  <xsd:sequence>
    <xsd:element name="businessDayConvention" type="BusinessDayConventionEnum" />
    <xsd:group ref=" BusinessCentersOrReference.model " minOccurs="0" />
  </xsd:sequence>
  <xsd:attribute name="id" type="xsd:ID" />
</xsd:complexType>
  
```

[top](#)

## Complex Type: `BusinessDayAdjustmentsReference`

Super-types: [Reference](#) < `BusinessDayAdjustmentsReference` (by extension)

Sub-types: None

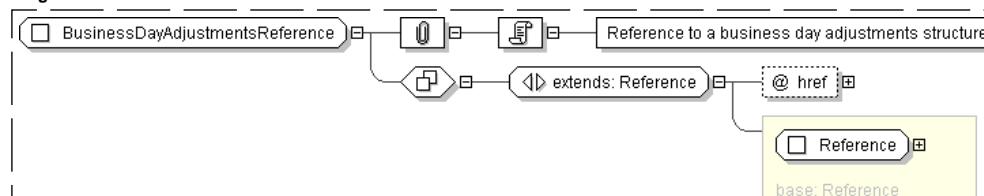
Name	BusinessDayAdjustmentsReference
Used by (from the same schema document)	Complex Type <a href="#">AdjustableDate2</a>
Abstract	no
Documentation	Reference to a business day adjustments structure.

#### XML Instance Representation

```

<...
  href="#" type="xsd:IDREF" />
  
```

#### Diagram



**Schema Component Representation**

```
<xsd:complexType name="BusinessDayAdjustmentsReference">
  <xsd:complexContent>
    <xsd:extension base=" Reference ">
      <xsd:attribute name="href" type=" xsd:IDREF " use="required"
        reference="BusinessDayAdjustments"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

[top](#)**Complex Type: CalculationAgent**

Super-types:	None
Sub-types:	None

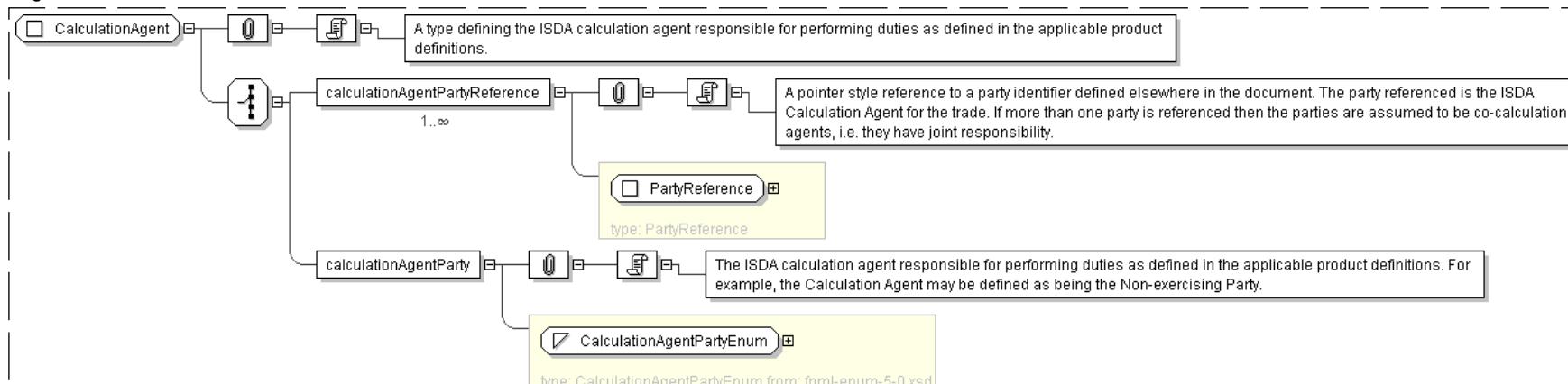
Name	CalculationAgent
Abstract	no
Documentation	A type defining the ISDA calculation agent responsible for performing duties as defined in the applicable product definitions.

**XML Instance Representation**

```
<...>
Start Choice [1]
<calculationAgentPartyReference> PartyReference </calculationAgentPartyReference> [1..*]
'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.'

<calculationAgentParty> CalculationAgentPartyEnum </calculationAgentParty> [1]
'The ISDA calculation agent responsible for performing duties as defined in the
applicable product definitions. For example, the Calculation Agent may be defined as being
the Non-exercising Party.'

End Choice
</...>
```

**Diagram****Schema Component Representation**

```
<xsd:complexType name="CalculationAgent">
  <xsd:choice>
```

```

<xsd:element name="calculationAgentPartyReference" type=" PartyReference
  " maxOccurs="unbounded"/>
<xsd:element name="calculationAgentParty" type=" CalculationAgentPartyEnum " />
</xsd:choice>
</xsd:complexType>

```

## Complex Type: CalculationPeriodFrequency

Super-types:	<a href="#">Frequency</a> < CalculationPeriodFrequency (by extension)
Sub-types:	None

Name	CalculationPeriodFrequency
Used by (from the same schema document)	Complex Type <a href="#">PeriodicDates</a>
Abstract	no
Documentation	A type defining the frequency at which calculation period end dates occur within the regular part of the calculation period schedule and their roll date convention. In case the calculation frequency is of value T (term), the period is defined by the swap\swapStream\calculationPerioDates\effectiveDate and the swap\swapStream\calculationPerioDates\terminationDate.

### XML Instance Representation

```

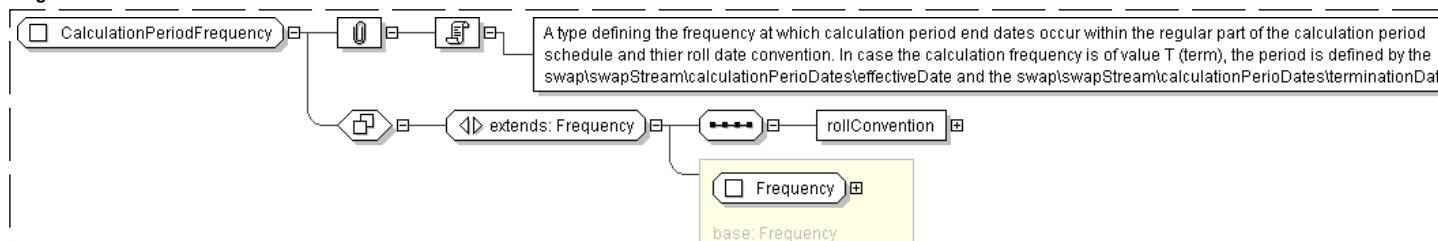
<...
  id=" xsd:ID [0..1]">
<periodMultiplier> xsd:integer </periodMultiplier> [1]
  'A time period multiplier, e.g. 1, 2 or 3 etc. A negative value can be used when specifying
  an offset relative to another date, e.g. -2 days. If the period value is T (Term)
  then periodMultiplier must contain the value 1.'

<period> PeriodExtendedEnum </period> [1]
  '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).'

<rollConvention> RollConventionEnum </rollConvention> [1]
  '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.'
</...>

```

### Diagram



### Schema Component Representation

```

<xsd:complexType name="CalculationPeriodFrequency">
  <xsd:complexContent>
    <xsd:extension base=" Frequency ">
      <xsd:sequence>
        <xsd:element name="rollConvention" type=" RollConventionEnum " />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

## Complex Type: CashSettlementReferenceBanks

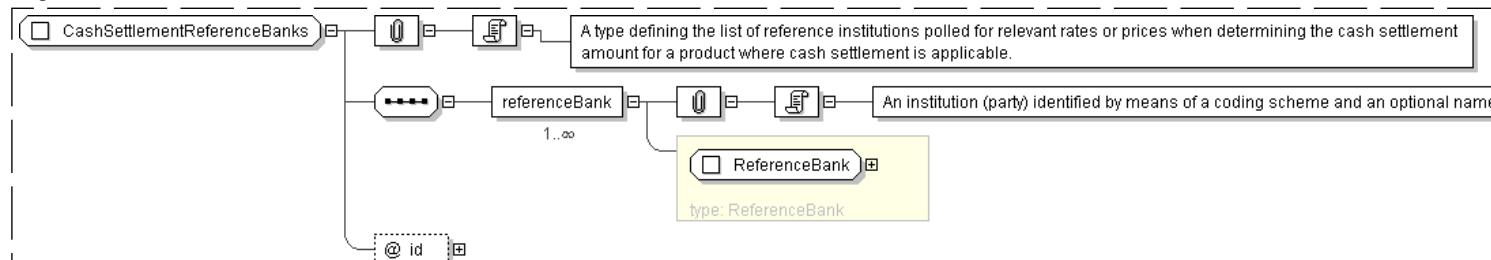
**Super-types:** None  
**Sub-types:** None

<b>Name</b>	CashSettlementReferenceBanks
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">SettlementRateSource</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type defining the list of reference institutions polled for relevant rates or prices when determining the cash settlement amount for a product where cash settlement is applicable.

### XML Instance Representation

```
<...>
  id="xsd:ID [0..1]">
    <referenceBank> ReferenceBank </referenceBank> [1..*]
      "An institution (party) identified by means of a coding scheme and an optional name."
    </...>
```

### Diagram



### Schema Component Representation

```

<xsd:complexType name="CashSettlementReferenceBanks">
  <xsd:sequence>
    <xsd:element name="referenceBank" type="ReferenceBank" maxOccurs="unbounded"/>
  </xsd:sequence>
  <xsd:attribute name="id" type="xsd:ID" />
</xsd:complexType>
  
```

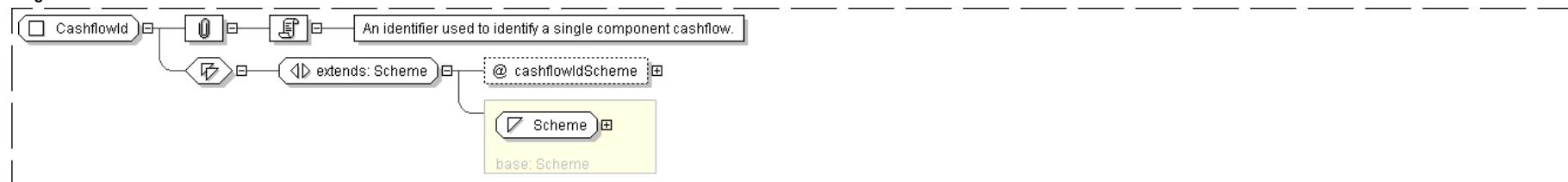
## Complex Type: CashflowId

**Super-types:** [xsd:normalizedString](#) < [Scheme](#) (by restriction) < **CashflowId** (by extension)  
**Sub-types:** None

<b>Name</b>	CashflowId
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">GrossCashflow</a>
<b>Abstract</b>	no
<b>Documentation</b>	An identifier used to identify a single component cashflow.

### XML Instance Representation

```
<...>
  cashflowIdScheme="xsd:anyURI [0..1]">
  Scheme
  </...>
```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="CashflowId">
  <xsd:simpleContent>
    <xsd:extension base=" Scheme ">
      <xsd:attribute name="cashflowIdScheme" type=" xsd:anyURI " />
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
  
```

top

**Complex Type: CashflowType**

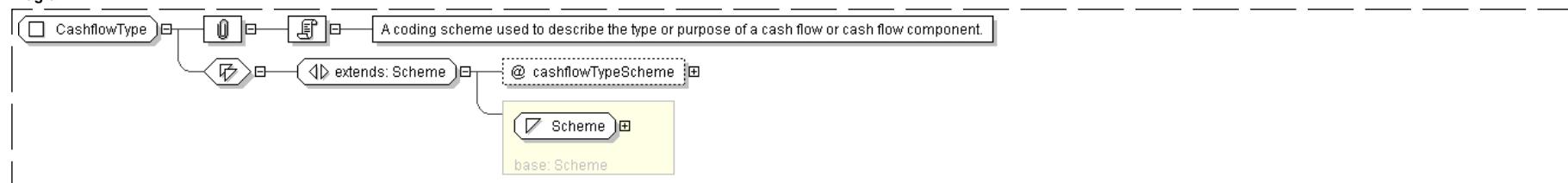
<b>Super-types:</b>	<a href="#">xsd:normalizedString</a> < <a href="#">Scheme</a> (by restriction) < <b>CashflowType</b> (by extension)
<b>Sub-types:</b>	None

<b>Name</b>	CashflowType
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">GrossCashflow</a>
<b>Abstract</b>	no
<b>Documentation</b>	A coding scheme used to describe the type or purpose of a cash flow or cash flow component.

**XML Instance Representation**

```

<...
  cashflowTypeScheme=" xsd:anyURI [ 0..1 ]">
  Scheme
</...>
  
```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="CashflowType">
  <xsd:simpleContent>
    <xsd:extension base=" Scheme ">
      <xsd:attribute name="cashflowTypeScheme" type=" xsd:anyURI " default="http://www.fpml.org/coding-scheme/cashflow-type" />
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
  
```

top

**Complex Type: ClearanceSystem**

**Super-types:** [xsd:normalizedString](#) < [Scheme](#) (by restriction) < **ClearanceSystem** (by extension)

**Sub-types:** None

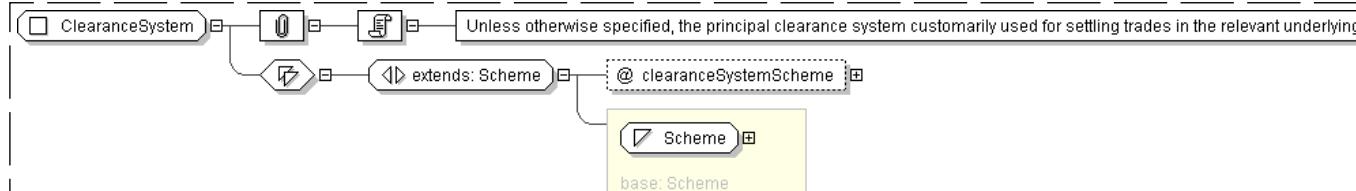
**Name** ClearanceSystem

**Abstract** no

**Documentation** Unless otherwise specified, the principal clearance system customarily used for settling trades in the relevant underlying.

**XML Instance Representation**

```
<...>
  clearanceSystemScheme=" xsd:anyURI [0..1]">
  Scheme
</...>
```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="ClearanceSystem">
  <xsd:simpleContent>
    <xsd:extension base=" Scheme ">
      <xsd:attribute name="clearanceSystemScheme" type=" xsd:anyURI " default="http://www.fpml.org/coding-scheme/clearance-system"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
  
```

top

**Complex Type: ContractualDefinitions**

**Super-types:** [xsd:normalizedString](#) < [Scheme](#) (by restriction) < **ContractualDefinitions** (by extension)

**Sub-types:** None

**Name** ContractualDefinitions

**Used by (from the same schema document)** Complex Type [Documentation](#)

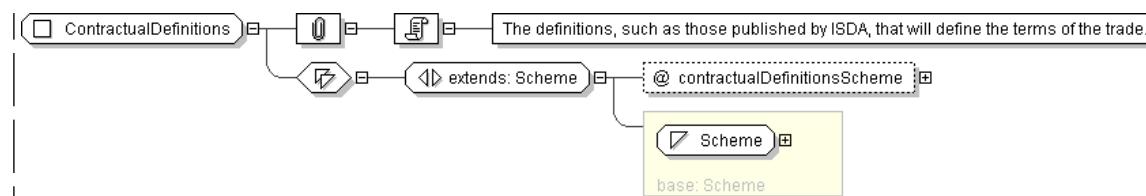
**Abstract** no

**Documentation** The definitions, such as those published by ISDA, that will define the terms of the trade.

**XML Instance Representation**

```
<...>
  contractualDefinitionsScheme=" xsd:anyURI [0..1]">
  Scheme
</...>
```

**Diagram**

**Schema Component Representation**

```

<xsd:complexType name="ContractualDefinitions">
  <xsd:simpleContent>
    <xsd:extension base=" Scheme ">
      <xsd:attribute name="contractualDefinitionsScheme" type=" xsd:anyURI " default="http://www.
        fpml.org/coding-scheme/contractual-definitions"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
  
```

top

**Complex Type: ContractualMatrix**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None

<b>Name</b>	ContractualMatrix
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">Documentation</a>
<b>Abstract</b>	no

**XML Instance Representation**

```

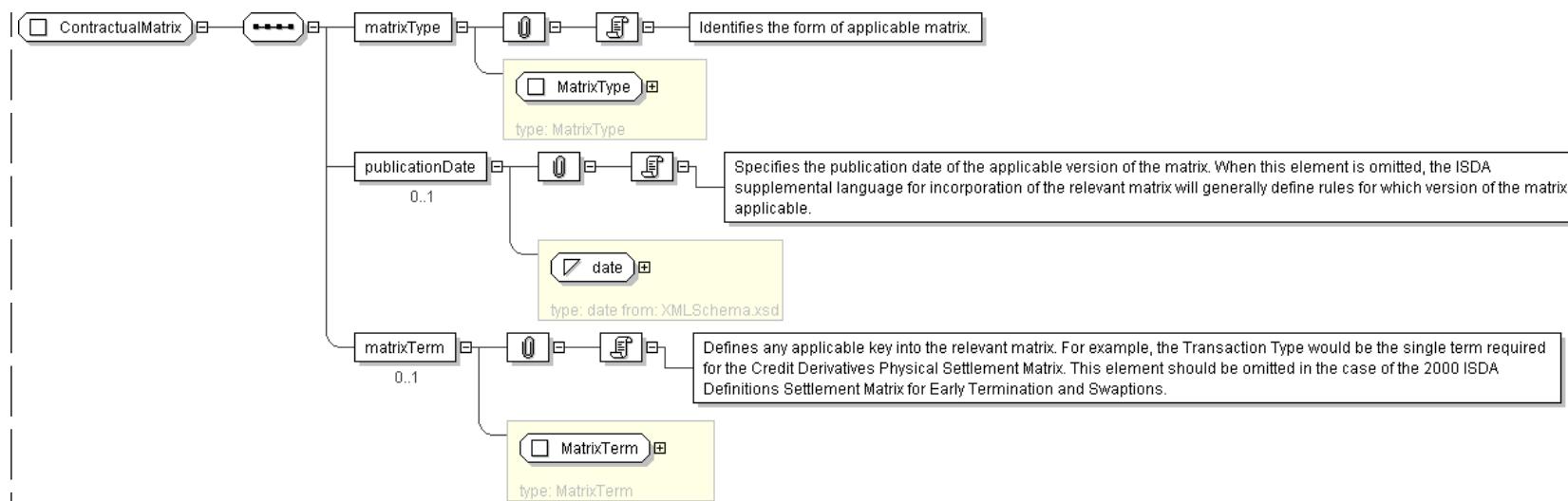
<...>
<matrixType> MatrixType </matrixType> [1]
'Identifies the form of applicable matrix.'

<publicationDate> xsd:date </publicationDate> [0..1]
'Specifies the publication date of the applicable version of the matrix. When this element
is omitted, the ISDA supplemental language for incorporation of the relevant matrix
will generally define rules for which version of the matrix is applicable.'

<matrixTerm> MatrixTerm </matrixTerm> [0..1]
'Defines any applicable key into the relevant matrix. For example, the Transaction Type
would be the single term required for the Credit Derivatives Physical Settlement Matrix.
This element should be omitted in the case of the 2000 ISDA Definitions Settlement Matrix
for Early Termination and Swaptions.'

</...>
  
```

**Diagram**

**Schema Component Representation**

```
<xsd:complexType name="ContractualMatrix">
  <xsd:sequence>
    <xsd:element name="matrixType" type="#MatrixType" />
    <xsd:element name="publicationDate" type="xsd:date" minOccurs="0"/>
    <xsd:element name="matrixTerm" type="#MatrixTerm" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>
```

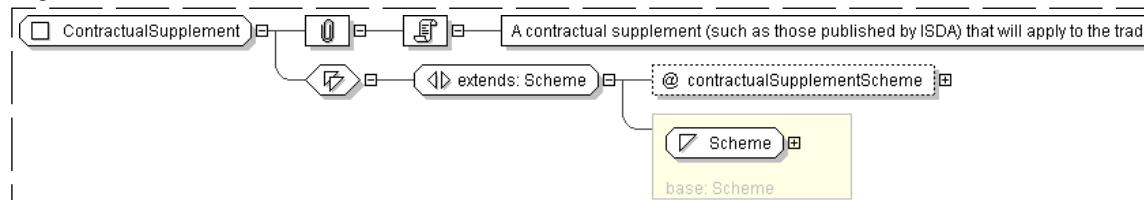
[top](#)**Complex Type: ContractualSupplement**

<b>Super-types:</b>	<code>xsd:normalizedString</code> < <a href="#">Scheme</a> (by restriction) < <b>ContractualSupplement</b> (by extension)
<b>Sub-types:</b>	None

<b>Name</b>	ContractualSupplement
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">ContractualTermsSupplement</a>
<b>Abstract</b>	no
<b>Documentation</b>	A contractual supplement (such as those published by ISDA) that will apply to the trade.

**XML Instance Representation**

```
<...
  contractualSupplementScheme=" xsd:anyURI [0..1]">
  Scheme
</...>
```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="ContractualSupplement">
  <xsd:simpleContent>
    <xsd:extension base=" Scheme ">
      <xsd:attribute name="contractualSupplementScheme" type=" xsd:anyURI " default="http://www.
        fpmi.org/coding-scheme/contractual-supplement "/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>

```

[top](#)

## Complex Type: ContractualTermsSupplement

Super-types:	None
Sub-types:	None

Name	ContractualTermsSupplement
Used by (from the same schema document)	Complex Type <a href="#">Documentation</a>
Abstract	no
Documentation	A contractual supplement (such as those published by ISDA) and its publication date that will apply to the trade.

### XML Instance Representation

```

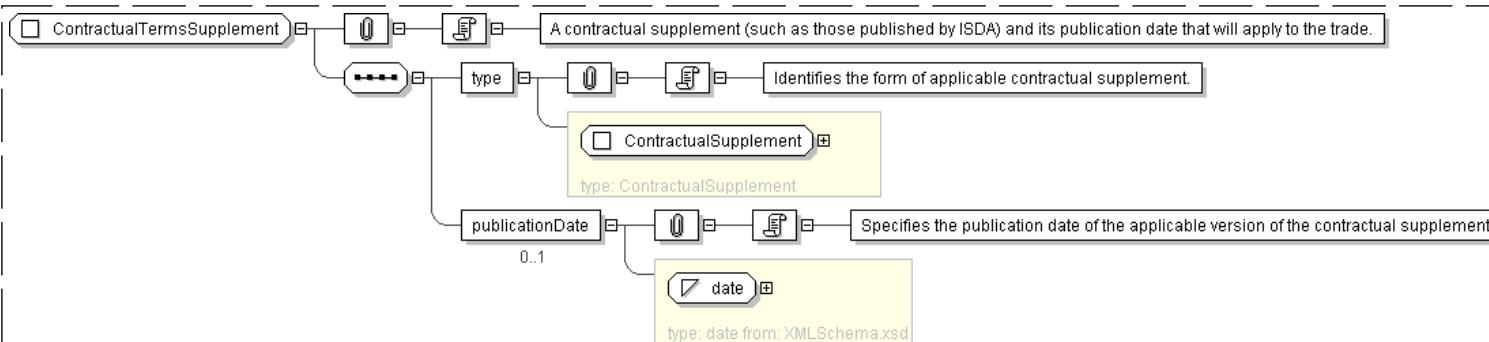
<...>
  <type> ContractualSupplement </type> [1]
  'Identifies the form of applicable contractual supplement.'

  <publicationDate> xsd:date </publicationDate> [0..1]
  'Specifies the publication date of the applicable version of the contractual supplement.'

</...>

```

### Diagram



### Schema Component Representation

```

<xsd:complexType name="ContractualTermsSupplement">
  <xsd:sequence>
    <xsd:element name="type" type=" ContractualSupplement "/>
    <xsd:element name="publicationDate" type=" xsd:date " minOccurs="0 "/>
  </xsd:sequence>
</xsd:complexType>

```

[top](#)

## Complex Type: CorrespondentInformation

Super-types:	None
--------------	------

Sub-types:

None

Name	CorrespondentInformation
Used by (from the same schema document)	Complex Type <a href="#">SettlementInstruction</a>
Abstract	no
Documentation	A type that describes the information to identify a correspondent bank that will make delivery of the funds on the paying bank's behalf in the country where the payment is to be made.

**XML Instance Representation**

```

<...>
Start Choice [1]
  <routingIds> RoutingIds </routingIds> [1]
    'A set of unique identifiers for a party, each one identifying the party within a
    payment system. The assumption is that each party will not have more than one identifier
    within the same payment system.'

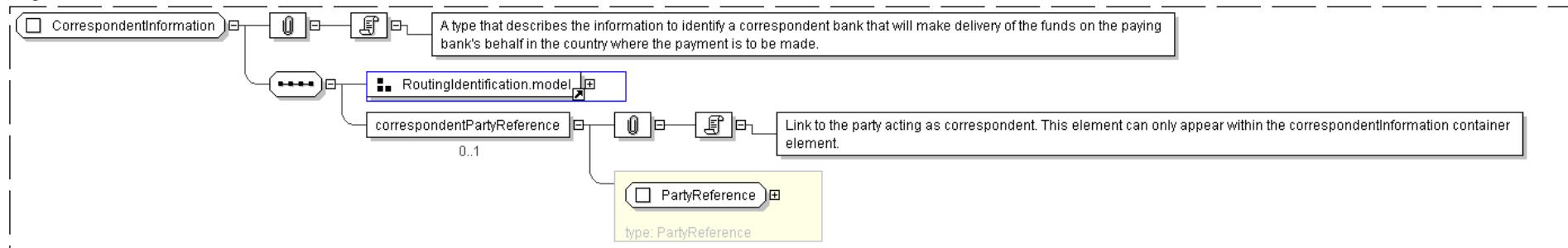
  <routingExplicitDetails> RoutingExplicitDetails </routingExplicitDetails> [1]
    'A set of details that is used to identify a party involved in the routing of a payment
    when the party does not have a code that identifies it within one of the recognized
    payment systems.'

  <routingIdsAndExplicitDetails> RoutingIdsAndExplicitDetails </routingIdsAndExplicitDetails> [1]
    'A combination of coded payment system identifiers and details for physical addressing for
    a party involved in the routing of a payment.'

End Choice
<correspondentPartyReference> PartyReference </correspondentPartyReference> [0..1]
  'Link to the party acting as correspondent. This element can only appear within
  the correspondentInformation container element.'

</...>

```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="CorrespondentInformation">
  <xsd:sequence>
    <xsd:group ref="#RoutingIdentification.model" />
    <xsd:element name="correspondentPartyReference" type="PartyReference" minOccurs="0" />
  </xsd:sequence>
</xsd:complexType>

```

[top](#)**Complex Type: CountryCode**

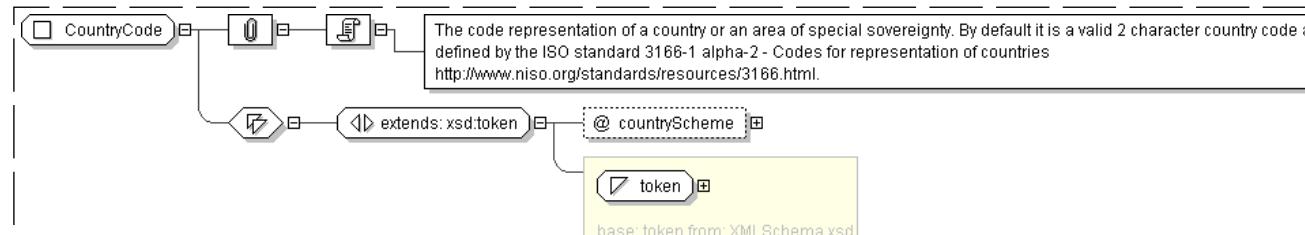
Super-types:	<a href="#">xsd:token</a> < <b>CountryCode</b> (by extension)
Sub-types:	None

Name	CountryCode
------	-------------

<b>Used by (from the same schema document)</b>	Complex Type <a href="#">Address</a> , Model Group <a href="#">PartyInformation.model</a>
<b>Abstract</b>	no
<b>Documentation</b>	The code representation of a country or an area of special sovereignty. By default it is a valid 2 character country code as defined by the ISO standard 3166-1 alpha-2 - Codes for representation of countries <a href="http://www.niso.org/standards/resources/3166.html">http://www.niso.org/standards/resources/3166.html</a> .

**XML Instance Representation**

```
<...>
<@ countryScheme=" xsd:anyURI [ 0..1 ] "xsd:token>
</...>
```

**Diagram****Schema Component Representation**

```
<xsd:complexType name="CountryCode">
  <xsd:simpleContent>
    <xsd:extension base=" xsd:token ">
      <xsd:attribute name="countryScheme" type=" xsd:anyURI " default="http://www.fpml.org/ext/iso3166"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
```

top

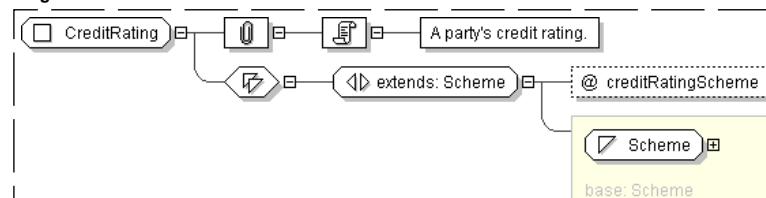
**Complex Type: CreditRating**

Super-types:	<a href="#">xsd:normalizedString</a> < <a href="#">Scheme</a> (by restriction) < <b>CreditRating</b> (by extension)
Sub-types:	None

<b>Name</b>	CreditRating
<b>Used by (from the same schema document)</b>	Model Group <a href="#">PartyInformation.model</a>
<b>Abstract</b>	no
<b>Documentation</b>	A party's credit rating.

**XML Instance Representation**

```
<...>
<@ creditRatingScheme=" xsd:anyURI [ 0..1 ] "xsd:Scheme>
</...>
```

**Diagram**

**Schema Component Representation**

```
<xsd:complexType name="CreditRating">
  <xsd:simpleContent>
    <xsd:extension base=" Scheme ">
      <xsd:attribute name="creditRatingScheme" type=" xsd:anyURI " default="http://www.fpml.org/
        ext/moodys"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
```

[top](#)**Complex Type: CreditSeniority**

**Super-types:** [xsd:normalizedString](#) < [Scheme](#) (by restriction) < **CreditSeniority** (by extension)

**Sub-types:** None

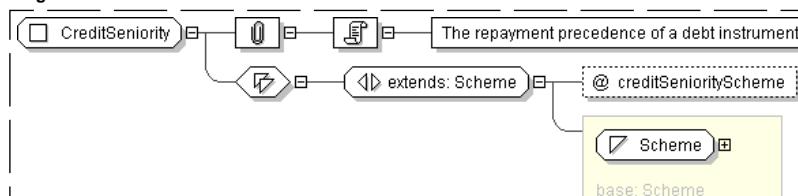
<b>Name</b>	CreditSeniority
-------------	-----------------

<b>Abstract</b>	no
-----------------	----

<b>Documentation</b>	The repayment precedence of a debt instrument.
----------------------	--

**XML Instance Representation**

```
<...
  creditSeniorityScheme=" xsd:anyURI [0..1]
  'creditSeniorityTradingScheme overrides creditSeniorityScheme when the underlyer defines
  the reference obligation used in a single name credit default swap trade.'>
  ...
  <*>
  <Scheme>
  </*>
  ...
  
```

**Diagram****Schema Component Representation**

```
<xsd:complexType name="CreditSeniority">
  <xsd:simpleContent>
    <xsd:extension base=" Scheme ">
      <xsd:attribute name="creditSeniorityScheme" type=" xsd:anyURI " default="http://www.fpml.
        org/coding-scheme/credit-seniority"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
```

[top](#)**Complex Type: CreditSupportAgreement**

**Super-types:** None

**Sub-types:** None

<b>Name</b>	CreditSupportAgreement
-------------	------------------------

<b>Used by (from the same schema document)</b>	Complex Type <a href="#">Documentation</a> , Complex Type <a href="#">PartyRelationshipDocumentation</a>
--	--

**Abstract**

no

**Documentation**

The agreement executed between the parties and intended to govern collateral arrangement for all OTC derivatives transactions between those parties.

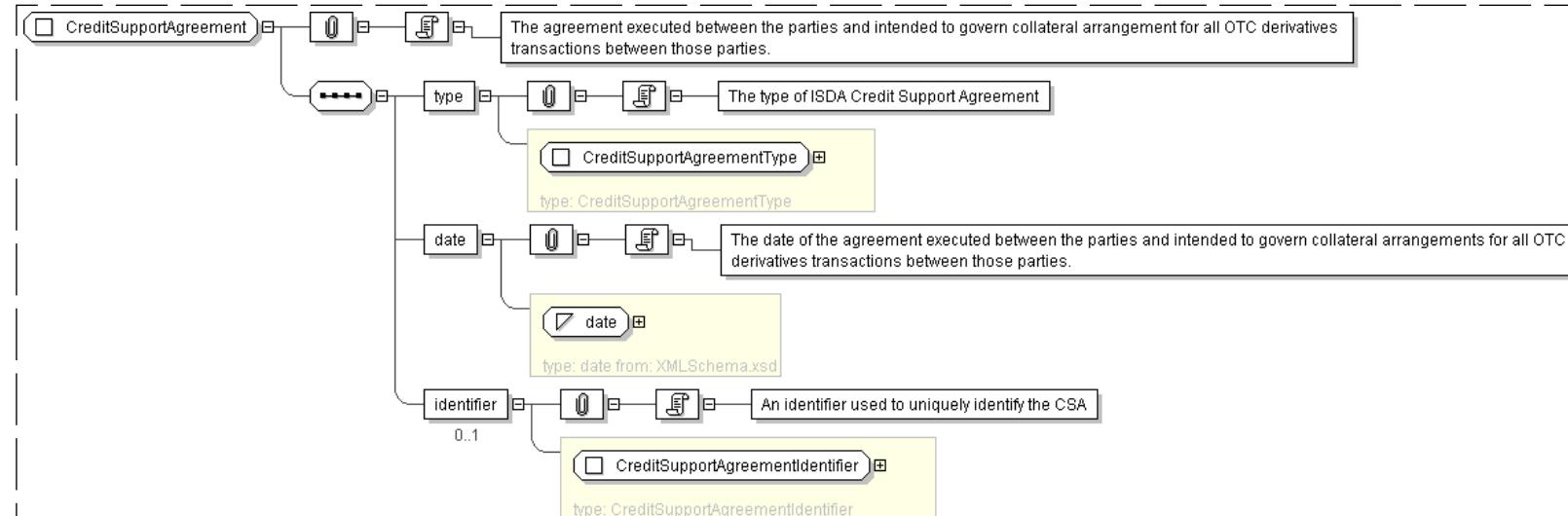
**XML Instance Representation**

```
<...>
<type> CreditSupportAgreementType </type> [1]
'The type of ISDA Credit Support Agreement'

<date> xsd:date </date> [1]
'The date of the agreement executed between the parties and intended to govern
collateral arrangements for all OTC derivatives transactions between those parties.'

<identifier> CreditSupportAgreementIdentifier </identifier> [0..1]
'An identifier used to uniquely identify the CSA'

</...>
```

**Diagram****Schema Component Representation**

```
<xsd:complexType name="CreditSupportAgreement">
  <xsd:sequence>
    <xsd:element name="type" type="#CreditSupportAgreementType" />
    <xsd:element name="date" type="xsd:date" />
    <!-- RPTWG 2009-09-04 begin-->
    <xsd:element name="identifier" type="#CreditSupportAgreementIdentifier" minOccurs="0" />
    <!-- RPTWG 2009-09-04 end -->
  </xsd:sequence>
</xsd:complexType>
```

top

**Complex Type: CreditSupportAgreementIdentifier**

Super-types:	<code>xsd:normalizedString</code> < <a href="#">Scheme</a> > (by restriction) < <b>CreditSupportAgreementIdentifier</b> > (by extension)
Sub-types:	None

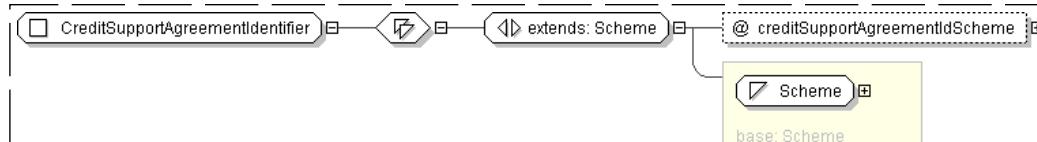
Name	<code>CreditSupportAgreementIdentifier</code>
Used by (from the same schema document)	Complex Type <b>CreditSupportAgreement</b>

**Abstract**

no

**XML Instance Representation**

```
<...>
<xsd:attribute name="creditSupportAgreementIdScheme" type="xsd:anyURI [0..1]">
  <!-- Scheme -->
</xsd:attribute>
```

**Diagram****Schema Component Representation**

```
<xsd:complexType name="CreditSupportAgreementIdentifier">
  <xsd:simpleContent>
    <xsd:extension base=" Scheme ">
      <xsd:attribute name="creditSupportAgreementIdScheme" type="xsd:anyURI "/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
```

top

**Complex Type: CreditSupportAgreementType**

**Super-types:** [xsd:normalizedString](#) < [Scheme](#) (by restriction) < **CreditSupportAgreementType** (by extension)  
**Sub-types:** None

**Name** CreditSupportAgreementType

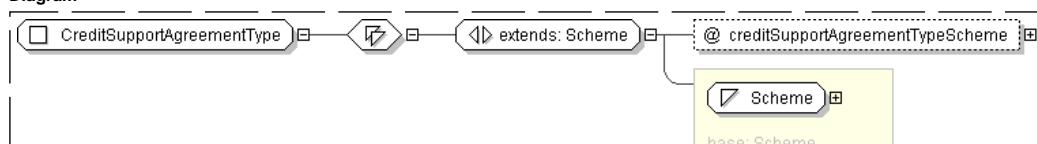
**Used by (from the same schema document)** Complex Type [CreditSupportAgreement](#)

**Abstract**

no

**XML Instance Representation**

```
<...>
<xsd:attribute name="creditSupportAgreementTypeScheme" type="xsd:anyURI [0..1]">
  <!-- Scheme -->
</xsd:attribute>
```

**Diagram****Schema Component Representation**

```
<xsd:complexType name="CreditSupportAgreementType">
  <xsd:simpleContent>
    <xsd:extension base=" Scheme ">
      <xsd:attribute name="creditSupportAgreementTypeScheme" type="xsd:anyURI " default="http://
        www.fpml.org/coding-scheme/credit-support-agreement-type"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
```

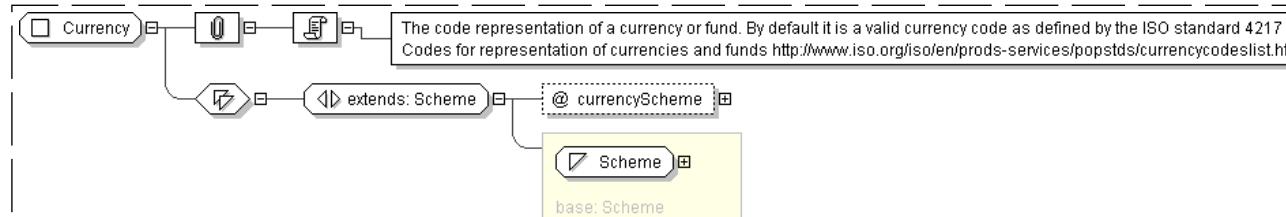
## Complex Type: Currency

Super-types:	<a href="#">xsd:normalizedString</a> < <a href="#">Scheme</a> (by restriction) < <b>Currency</b> (by extension)
Sub-types:	None
Name	Currency
Used by (from the same schema document)	Complex Type <a href="#">AmountSchedule</a> , Complex Type <a href="#">FxCashSettlement</a> , Complex Type <a href="#">MoneyBase</a> , Complex Type <a href="#">NonNegativeAmountSchedule</a> , Complex Type <a href="#">PositiveAmountSchedule</a> , Complex Type <a href="#">PricingStructure</a> , Complex Type <a href="#">QuotedCurrencyPair</a> , Complex Type <a href="#">QuotedCurrencyPair</a> , Model Group <a href="#">SettlementAmountOrCurrency.model</a>
Abstract	no
Documentation	The code representation of a currency or fund. By default it is a valid currency code as defined by the ISO standard 4217 - Codes for representation of currencies and funds <a href="http://www.iso.org/iso/en/prods-services/popstds/currencycodeslist.html">http://www.iso.org/iso/en/prods-services/popstds/currencycodeslist.html</a> .

### XML Instance Representation

```
<...>
<@currencyScheme=" xsd:anyURI [0..1]">
<Scheme>
</...>
```

### Diagram



### Schema Component Representation

```
<xsd:complexType name="Currency">
  <xsd:simpleContent>
    <xsd:extension base=" Scheme ">
      <xsd:attribute name="currencyScheme" type=" xsd:anyURI " default="http://www.fpml.org/ext/iso4217-2001-08-15">
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
```

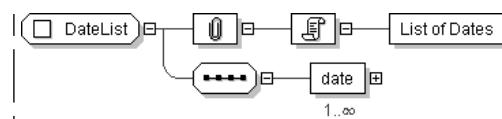
## Complex Type: DateList

Super-types:	None
Sub-types:	None
Name	DateList
Abstract	no
Documentation	List of Dates

### XML Instance Representation

```
<...>
<date> xsd:date </date> [1...*]
</...>
```

### Diagram

**Schema Component Representation**

```

<xsd:complexType name="DateList">
  <xsd:sequence>
    <xsd:element name="date" type="xsd:date" maxOccurs="unbounded" />
  </xsd:sequence>
</xsd:complexType>
  
```

top

**Complex Type: DateOffset**

<b>Super-types:</b>	<a href="#">Period</a> < <a href="#">Offset</a> (by extension) < <b>DateOffset</b> (by extension)
<b>Sub-types:</b>	None

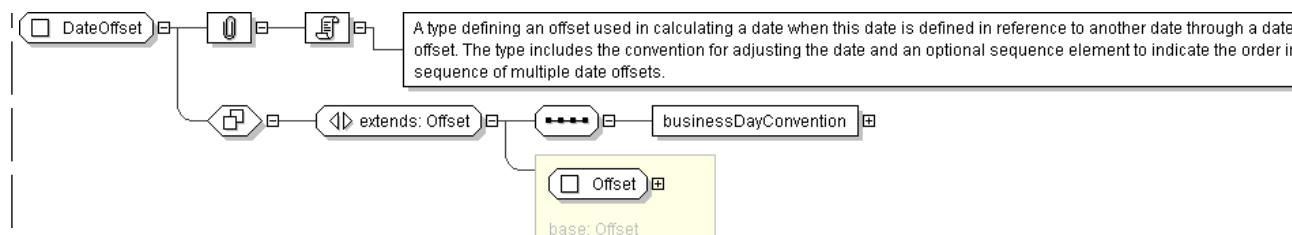
<b>Name</b>	DateOffset
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">RelativeDateSequence</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type defining an offset used in calculating a date when this date is defined in reference to another date through a date offset. The type includes the convention for adjusting the date and an optional sequence element to indicate the order in a sequence of multiple date offsets.

**XML Instance Representation**

```

<...
  id="#ID [0..1]">
  <periodMultiplier> xsd:integer </periodMultiplier> [1]
  '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.'
  <period> PeriodEnum </period> [1]
  'A time period, e.g. a day, week, month or year of the stream. If the periodMultiplier value
  is 0 (zero) then period must contain the value D (day).'
  <dayType> DayTypeEnum </dayType> [0..1]
  '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.'
  <businessDayConvention> BusinessDayConventionEnum </businessDayConvention> [1]
  'The convention for adjusting a date if it would otherwise fall on a day that is not a
  business day.'
</...>
  
```

**Diagram**

**Schema Component Representation**

```

<xsd:complexType name="DateOffset">
  <xsd:complexContent>
    <xsd:extension base=" Offset ">
      <xsd:sequence>
        <xsd:element name="businessDayConvention" type=" BusinessDayConventionEnum " />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
  
```

top

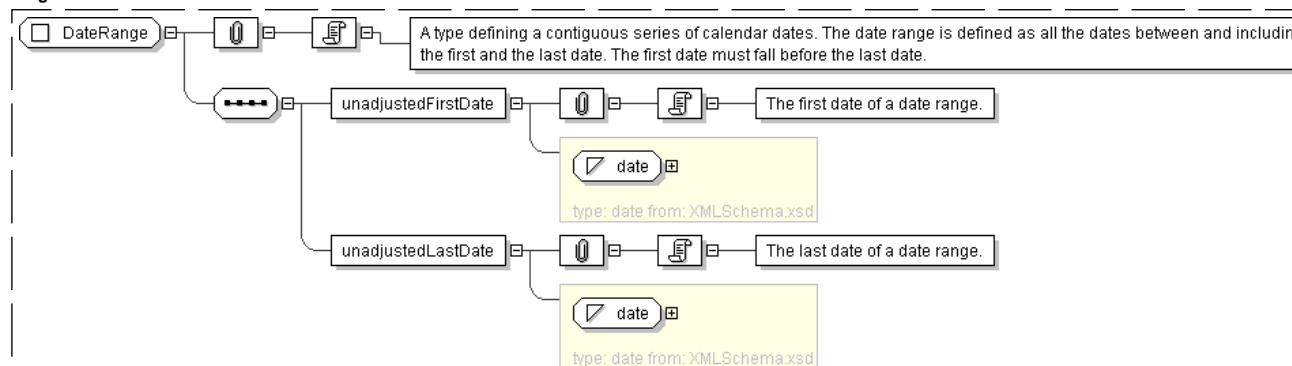
**Complex Type: DateRange**

<b>Super-types:</b>	None
<b>Sub-types:</b>	<ul style="list-style-type: none"> <li>• <a href="#">BusinessDateRange</a> (by extension)</li> </ul>
<b>Name</b>	DateRange
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">RelativeDates</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type defining a contiguous series of calendar dates. The date range is defined as all the dates between and including the first and the last date. The first date must fall before the last date.

**XML Instance Representation**

```

<...>
<unadjustedFirstDate> xsd:date </unadjustedFirstDate> [1]
  'The first date of a date range.'
<unadjustedLastDate> xsd:date </unadjustedLastDate> [1]
  'The last date of a date range.'
</...>
  
```

**Diagram**

**Schema Component Representation**

```
<xsd:complexType name="DateRange">
  <xsd:sequence>
    <xsd:element name="unadjustedFirstDate" type="xsd:date" />
    <xsd:element name="unadjustedLastDate" type="xsd:date" />
  </xsd:sequence>
</xsd:complexType>
```

top

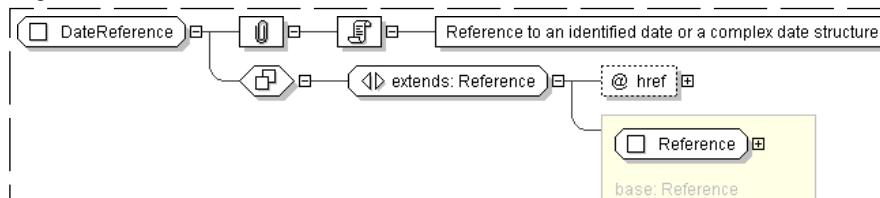
**Complex Type: DateReference**

Super-types:	<a href="#">Reference</a> < DateReference (by extension)
Sub-types:	None

Name	DateReference
Used by (from the same schema document)	Complex Type <a href="#">RelativeDateOffset</a> , Complex Type <a href="#">RelativeDateSequence</a>
Abstract	no
Documentation	Reference to an identified date or a complex date structure.

**XML Instance Representation**

```
<...>
  <@ href=" xsd:IDREF [1]" />
```

**Diagram****Schema Component Representation**

```
<xsd:complexType name="DateReference">
  <xsd:complexContent>
    <xsd:extension base=" Reference ">
      <xsd:attribute name="href" type="xsd:IDREF" use="required"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

top

**Complex Type: DateTimeList**

Super-types:	None
Sub-types:	None

Name	DateTimeList
Abstract	no
Documentation	List of DateTimes

**XML Instance Representation**

```
<...>
  <dateTime> xsd:dateTime </dateTime> [1..*]
</...>
```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="DateTimeList">
  <xsd:sequence>
    <xsd:element name="dateTime" type="xsd:dateTime" maxOccurs="unbounded" />
  </xsd:sequence>
</xsd:complexType>
  
```

[top](#)**Complex Type: DayCountFraction**

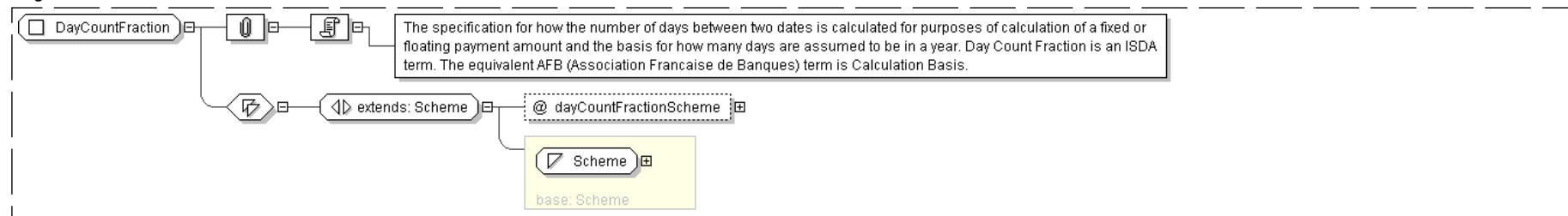
**Super-types:** [xsd:normalizedString](#) < [Scheme](#) (by restriction) < **DayCountFraction** (by extension)  
**Sub-types:** None

<b>Name</b>	DayCountFraction
<b>Abstract</b>	no
<b>Documentation</b>	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 Francaise de Banques) term is Calculation Basis.

**XML Instance Representation**

```

<...
  dayCountFractionScheme="xsd:anyURI [0..1]">
  Scheme
</...>
  
```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="DayCountFraction">
  <xsd:simpleContent>
    <xsd:extension base="Scheme">
      <xsd:attribute name="dayCountFractionScheme" type="xsd:anyURI" default="http://www.fpml.org/coding-scheme/day-count-fraction"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
  
```

[top](#)**Complex Type: DeterminationMethod**

**Super-types:** [xsd:normalizedString](#) < [Scheme](#) (by restriction) < **DeterminationMethod** (by extension)

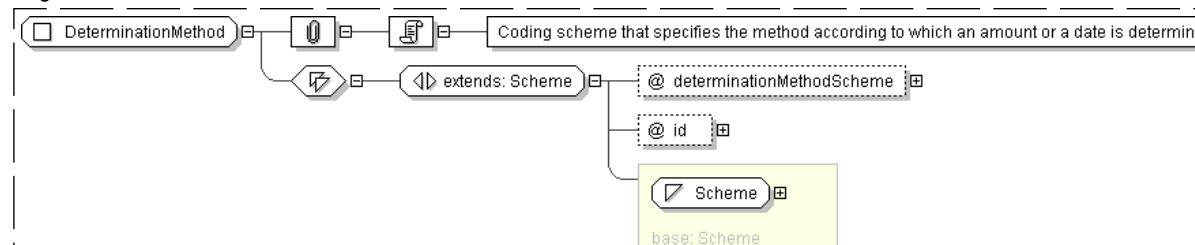
Sub-types:

None

Name	DeterminationMethod
Used by (from the same schema document)	Complex Type <a href="#">SharedAmericanExercise</a>
Abstract	no
Documentation	Coding scheme that specifies the method according to which an amount or a date is determined.

**XML Instance Representation**

```
<...>
<determinationMethodScheme=" xsd:anyURI [0..1]">
  id=" xsd:ID [0..1]">
  Scheme
</...>
```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="DeterminationMethod">
  <xsd:simpleContent>
    <xsd:extension base=" Scheme ">
      <xsd:attribute name="determinationMethodScheme" type=" xsd:anyURI " default="http://www.
        fpmml.org/coding-scheme/determination-method"/>
      <xsd:attribute name="id" type=" xsd:ID "/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
  
```

[top](#)**Complex Type: DeterminationMethodReference**

Super-types:

[Reference](#) < DeterminationMethodReference (by extension)

Sub-types:

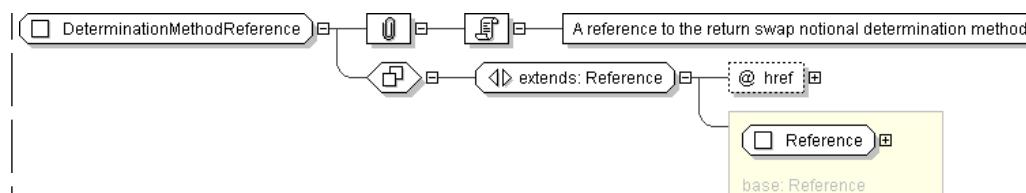
None

Name	DeterminationMethodReference
Abstract	no
Documentation	A reference to the return swap notional determination method.

**XML Instance Representation**

```
<...>
<ref href=" xsd:IDREF [1]">
```

**Diagram**

**Schema Component Representation**

```

<xsd:complexType name="DeterminationMethodReference">
  <xsd:complexContent>
    <xsd:extension base=" Reference ">
      <xsd:attribute name="href" type=" xsd:IDREF " use="required" reference="DeterminationMethod"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
  
```

[top](#)**Complex Type: Documentation**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None

<b>Name</b>	Documentation
<b>Abstract</b>	no
<b>Documentation</b>	An entity for defining the definitions that govern the document and should include the year and type of definitions referenced, along with any relevant documentation (such as master agreement) and the date it was signed.

**XML Instance Representation**

```

<...>
  <masterAgreement> MasterAgreement </masterAgreement> [0..1]
  'The agreement executed between the parties and intended to govern all OTC
  derivatives transactions between those parties.'

  Start Choice [0..1]
    <masterConfirmation> MasterConfirmation </masterConfirmation> [1]
    'The agreement executed between the parties and intended to govern all OTC
    derivatives transactions between those parties.'

    <brokerConfirmation> BrokerConfirmation </brokerConfirmation> [1]
    'Specifies the details for a broker confirm.'

  End Choice
  <contractualDefinitions> ContractualDefinitions </contractualDefinitions> [0..*]
  'The definitions such as those published by ISDA that will define the terms of the trade.'

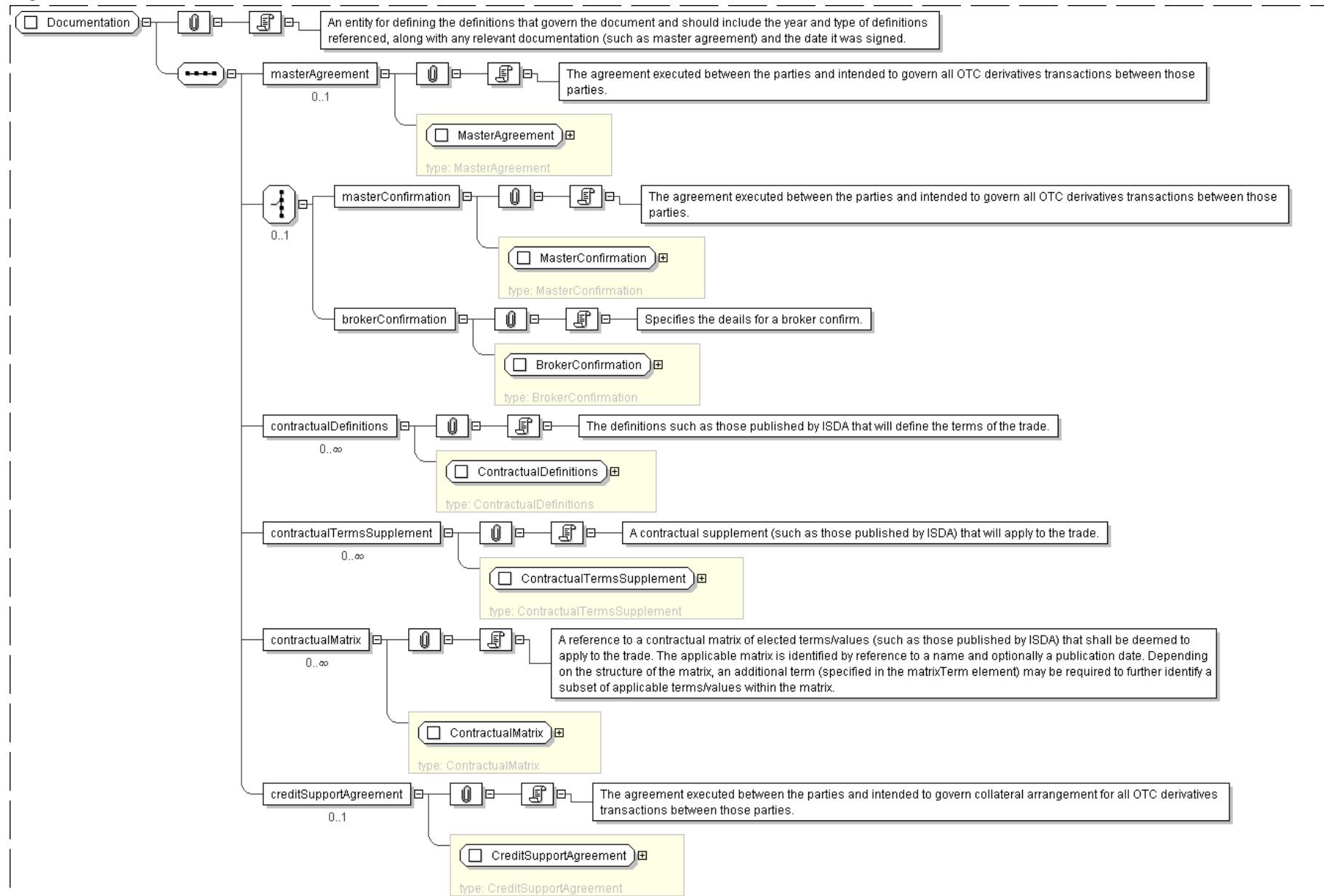
  <contractualTermsSupplement> ContractualTermsSupplement </contractualTermsSupplement> [0..*]
  'A contractual supplement (such as those published by ISDA) that will apply to the trade.'

  <contractualMatrix> ContractualMatrix </contractualMatrix> [0..*]
  'A reference to a contractual matrix of elected terms/values (such as those published by
  ISDA) that shall be deemed to apply to the trade. The applicable matrix is identified
  by reference to a name and optionally a publication date. Depending on the structure of
  the matrix, an additional term (specified in the matrixTerm element) may be required to
  further identify a subset of applicable terms/values within the matrix.'

  <creditSupportAgreement> CreditSupportAgreement </creditSupportAgreement> [0..1]
  'The agreement executed between the parties and intended to govern collateral arrangement
  for all OTC derivatives transactions between those parties.'

</...>
  
```

## Diagram



## Schema Component Representation

```

<xsd:complexType name="Documentation">
    <xsd:sequence>
        <xsd;element name="masterAgreement" type="#MasterAgreement" minOccurs="0"/>
        <xsd:choice minOccurs="0">
            <xsd;element name="masterConfirmation" type="#MasterConfirmation"/>
            <xsd;element name="brokerConfirmation" type="#BrokerConfirmation"/>
        </xsd:choice>
        <xsd;element name="contractualDefinitions" type="#ContractualDefinitions" minOccurs="0"/>
        <xsd;element name="contractualTermsSupplement" type="#ContractualTermsSupplement" minOccurs="0"/>
        <xsd;element name="contractualMatrix" type="#ContractualMatrix" minOccurs="0"/>
        <xsd;element name="creditSupportAgreement" type="#CreditSupportAgreement" minOccurs="1" maxOccurs="1"/>
    </xsd:sequence>
</xsd:complexType>

```

```

        minOccurs="0" maxOccurs="unbounded"/>
<xsd:element name="contractualTermsSupplement" type=" ContractualTermsSupplement"
        minOccurs="0" maxOccurs="unbounded"/>
<xsd:element name="contractualMatrix" type=" ContractualMatrix"
        minOccurs="0" maxOccurs="unbounded"/>
<xsd:element name="creditSupportAgreement" type=" CreditSupportAgreement" minOccurs="0"/>
</xsd:sequence>
</xsd:complexType>
```

top

## Complex Type: Empty

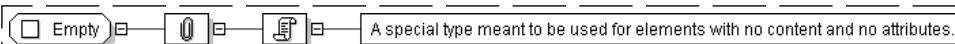
Super-types:	None
Sub-types:	None

Name	Empty
Abstract	no
Documentation	A special type meant to be used for elements with no content and no attributes.

### XML Instance Representation

&lt;.../&gt;

### Diagram



### Schema Component Representation

&lt;xsd:complexType name="Empty"/&gt;

top

## Complex Type: EntityId

Super-types:	xsd:normalizedString < Scheme (by restriction) < EntityId (by extension)
Sub-types:	None

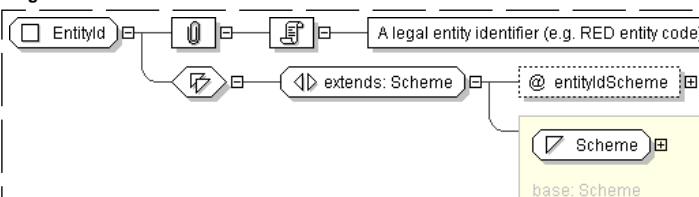
Name	EntityId
Used by (from the same schema document)	Complex Type LegalEntity , Complex Type LegalEntity
Abstract	no
Documentation	A legal entity identifier (e.g. RED entity code).

### XML Instance Representation

```

<...
entityIdScheme=" xsd:anyURI [0..1]">
Scheme
</...>
```

### Diagram



### Schema Component Representation

```

<xsd:complexType name="EntityId">
  <xsd:simpleContent>
    <xsd:extension base=" Scheme ">
      <xsd:attribute name="entityIdscheme" type=" xsd:anyURI " default="http://www.fpml.org/
        spec/2003/entity-id-RED-1-0"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>

```

[top](#)

## Complex Type: EntityName

**Super-types:** [xsd:normalizedString](#) < [Scheme](#) (by restriction) < **EntityName** (by extension)

**Sub-types:** None

<b>Name</b>	EntityName
-------------	------------

<b>Used by (from the same schema document)</b>	Complex Type <a href="#">LegalEntity</a>
--	--

<b>Abstract</b>	no
-----------------	----

<b>Documentation</b>	The name of the reference entity. A free format string. FpML does not define usage rules for this element.
----------------------	--

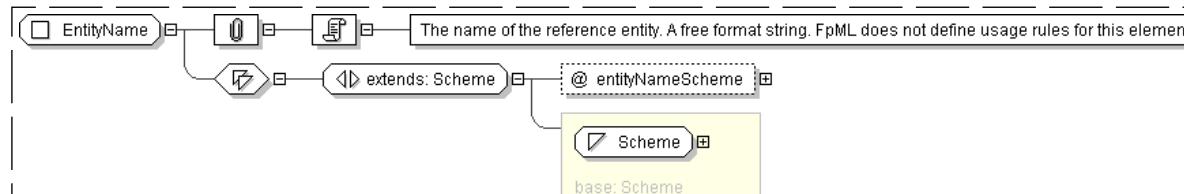
### XML Instance Representation

```

<...
  entityIdScheme=" xsd:anyURI [0..1]">
  Scheme
</...>

```

### Diagram



### Schema Component Representation

```

<xsd:complexType name="EntityName">
  <xsd:simpleContent>
    <xsd:extension base=" Scheme ">
      <xsd:attribute name="entityNameScheme" type=" xsd:anyURI " default="http://www.fpml.
        org/spec/2003/entity-name-RED-1-0"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>

```

[top](#)

## Complex Type: EuropeanExercise

**Super-types:** [Exercise](#) < **EuropeanExercise** (by extension)

**Sub-types:** None

<b>Name</b>	EuropeanExercise
-------------	------------------

<b>Used by (from the same schema document)</b>	Element <a href="#">europeanExercise</a>
--	--

<b>Abstract</b>	no
-----------------	----

<b>Documentation</b>	A type 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.
----------------------	---

**XML Instance Representation**

```
<...>
  id=" xsd:ID [0..1]">
  <expirationDate> AdjustableOrRelativeDate </expirationDate> [1]
  'The last day within an exercise period for an American style option. For a European
  style option it is the only day within the exercise period.'

  <relevantUnderlyingDate> AdjustableOrRelativeDates </relevantUnderlyingDate> [0..1]
  'The day on the underlying set by the exercise of an option. What this date is depends on
  the option (e.g. in a swaption it is the effective date, in an extendible/cancelable
  provision it is the termination date).'

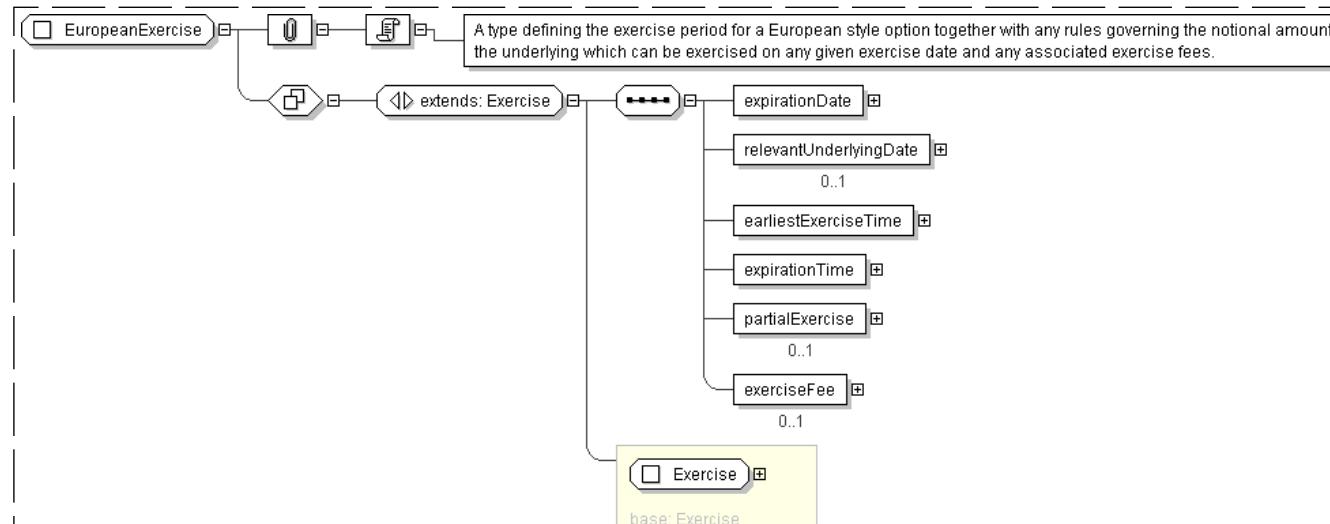
  <earliestExerciseTime> BusinessCenterTime </earliestExerciseTime> [1]
  'The earliest time at which notice of exercise can be given by the buyer to the seller
  (or seller's agent) i) on the expiration date, in the case of a European style option, (ii)
  on each bermuda option exercise date and the expiration date, in the case of a Bermuda
  style option the commencement date to, and including, the expiration date , in the case of
  an American option.'

  <expirationTime> BusinessCenterTime </expirationTime> [1]
  'The latest time for exercise on expirationDate.'

  <partialExercise> PartialExercise </partialExercise> [0..1]
  '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> ExerciseFee </exerciseFee> [0..1]
  '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.'

</...>
```

**Diagram****Schema Component Representation**

```
<xsd:complexType name="EuropeanExercise">
  <xsd:complexContent>
    <xsd:extension base=" Exercise ">
      <xsd:sequence>
```

```

<xsd:element name="expirationDate" type=" AdjustableOrRelativeDate " />
<xsd:element name="relevantUnderlyingDate" type=" AdjustableOrRelativeDates " minOccurs="0" />
<xsd:element name="earliestExerciseTime" type=" BusinessCenterTime " />
<xsd:element name="expirationTime" type=" BusinessCenterTime " />
<xsd:element name="partialExercise" type=" PartialExercise " minOccurs="0" />
<xsd:element name="exerciseFee" type=" ExerciseFee " minOccurs="0" />
</xsd:sequence>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>

```

[top](#)

## Complex Type: ExchangeId

**Super-types:** [xsd:normalizedString](#) < [Scheme](#) (by restriction) < **ExchangeId** (by extension)

**Sub-types:** None

<b>Name</b>	ExchangeId
-------------	------------

<b>Abstract</b>	no
-----------------	----

<b>Documentation</b>	A short form unique identifier for an exchange. If the element is not present then the exchange shall be the primary exchange on which the underlying is listed. The term "Exchange" is assumed to have the meaning as defined in the ISDA 2002 Equity Derivatives Definitions.
----------------------	---

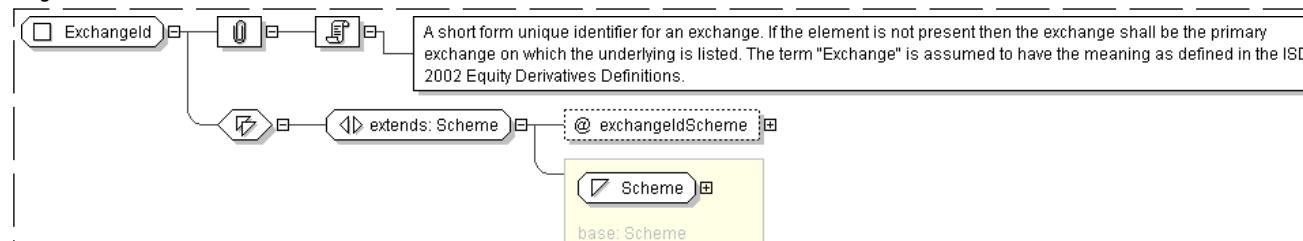
### XML Instance Representation

```

<...
  exchangeIdScheme=" xsd:anyURI [0..1]">
  Scheme
</...>

```

### Diagram



### Schema Component Representation

```

<xsd:complexType name="ExchangeId">
  <xsd:simpleContent>
    <xsd:extension base=" Scheme ">
      <xsd:attribute name="exchangeIdScheme" type=" xsd:anyURI " default="http://www.fpml.org/spec/2002/exchange-id-MIC-1-0"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>

```

[top](#)

## Complex Type: Exercise

**Super-types:** None

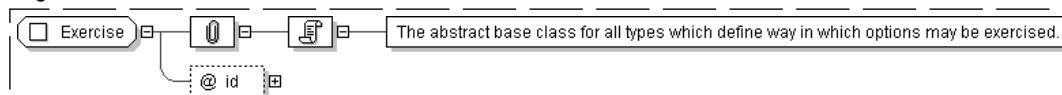
**Sub-types:**

- [AmericanExercise](#) (by extension)
- [BermudaExercise](#) (by extension)
- [EuropeanExercise](#) (by extension)
- [SharedAmericanExercise](#) (by extension)

<b>Name</b>	Exercise
<b>Used by (from the same schema document)</b>	Element <a href="#">exercise</a>
<b>Abstract</b>	no
<b>Documentation</b>	The abstract base class for all types which define way in which options may be exercised.

**XML Instance Representation**

```
<...>
  <@id> xsd:ID [0..1]</@id>
```

**Diagram****Schema Component Representation**

```
<xsd:complexType name="Exercise">
  <xsd:attribute name="id" type="xsd:ID" />
</xsd:complexType>
```

top

**Complex Type: ExerciseFee**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None

<b>Name</b>	ExerciseFee
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">EuropeanExercise</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type defining the fee payable on exercise of an option. This fee may be defined as an amount or a percentage of the notional exercised.

**XML Instance Representation**

```
<...>
  <payerPartyReference> PartyReference </payerPartyReference> [1]
  'A reference to the party responsible for making the payments defined by this structure.'

  <payerAccountReference> AccountReference </payerAccountReference> [0..1]
  'A reference to the account responsible for making the payments defined by this structure.'

  <receiverPartyReference> PartyReference </receiverPartyReference> [1]
  'A reference to the party that receives the payments corresponding to this structure.'

  <receiverAccountReference> AccountReference </receiverAccountReference> [0..1]
  'A reference to the account that receives the payments corresponding to this structure.'

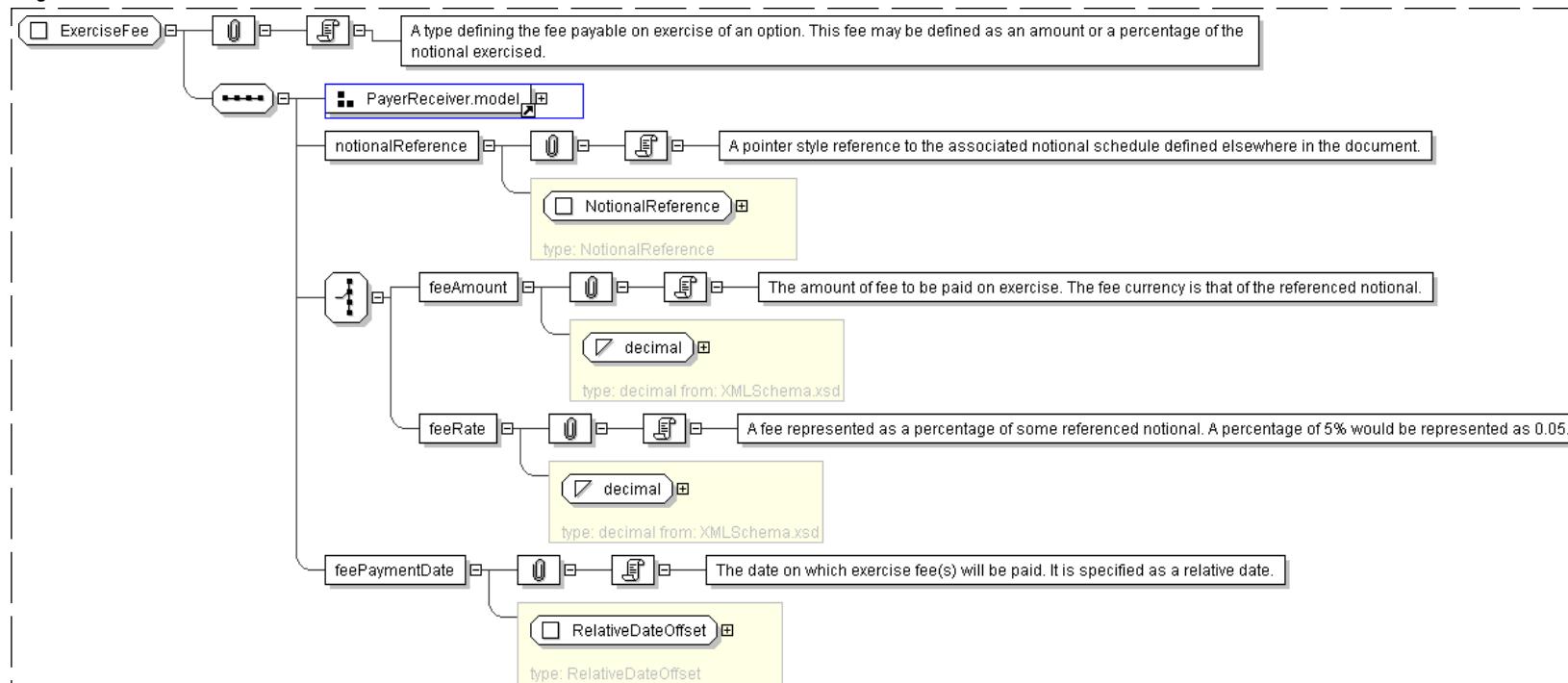
  <notionalReference> NotionalReference </notionalReference> [1]
  'A pointer style reference to the associated notional schedule defined elsewhere in
  the document.'

Start Choice [1]
  <feeAmount> xsd:decimal </feeAmount> [1]
  'The amount of fee to be paid on exercise. The fee currency is that of the referenced notional.'

  <feeRate> xsd:decimal </feeRate> [1]
  'A fee represented as a percentage of some referenced notional. A percentage of 5% would
  be represented as 0.05.'
End Choice
```

```
<feePaymentDate> RelativeDateOffset </feePaymentDate> [1]
'The date on which exercise fee(s) will be paid. It is specified as a relative date.'
```

&lt;...&gt;

**Diagram****Schema Component Representation**

```
<xsd:complexType name="ExerciseFee">
  <xsd:sequence>
    <xsd:group ref=" PayerReceiver.model "/>
    <xsd:element name="notionalReference" type=" NotionalReference "/>
    <xsd:choice>
      <xsd:element name="feeAmount" type=" xsd:decimal "/>
      <xsd:element name="feeRate" type=" xsd:decimal "/>
    </xsd:choice>
    <xsd:element name="feePaymentDate" type=" RelativeDateOffset "/>
  </xsd:sequence>
</xsd:complexType>
```

top

**Complex Type: ExerciseFeeSchedule**

Super-types:	None
Sub-types:	None

Name	ExerciseFeeSchedule
Used by (from the same schema document)	Complex Type <a href="#">AmericanExercise</a> , Complex Type <a href="#">BermudaExercise</a>
Abstract	no
Documentation	A type to define a fee or schedule of fees to be payable on the exercise of an option. This fee may be defined as an amount or a percentage of the notional exercised.

**XML Instance Representation**

```
<...>
<payerPartyReference> PartyReference </payerPartyReference> [1]
'A reference to the party responsible for making the payments defined by this structure.'

<payerAccountReference> AccountReference </payerAccountReference> [0..1]
'A reference to the account responsible for making the payments defined by this structure.'

<receiverPartyReference> PartyReference </receiverPartyReference> [1]
'A reference to the party that receives the payments corresponding to this structure.'

<receiverAccountReference> AccountReference </receiverAccountReference> [0..1]
'A reference to the account that receives the payments corresponding to this structure.'

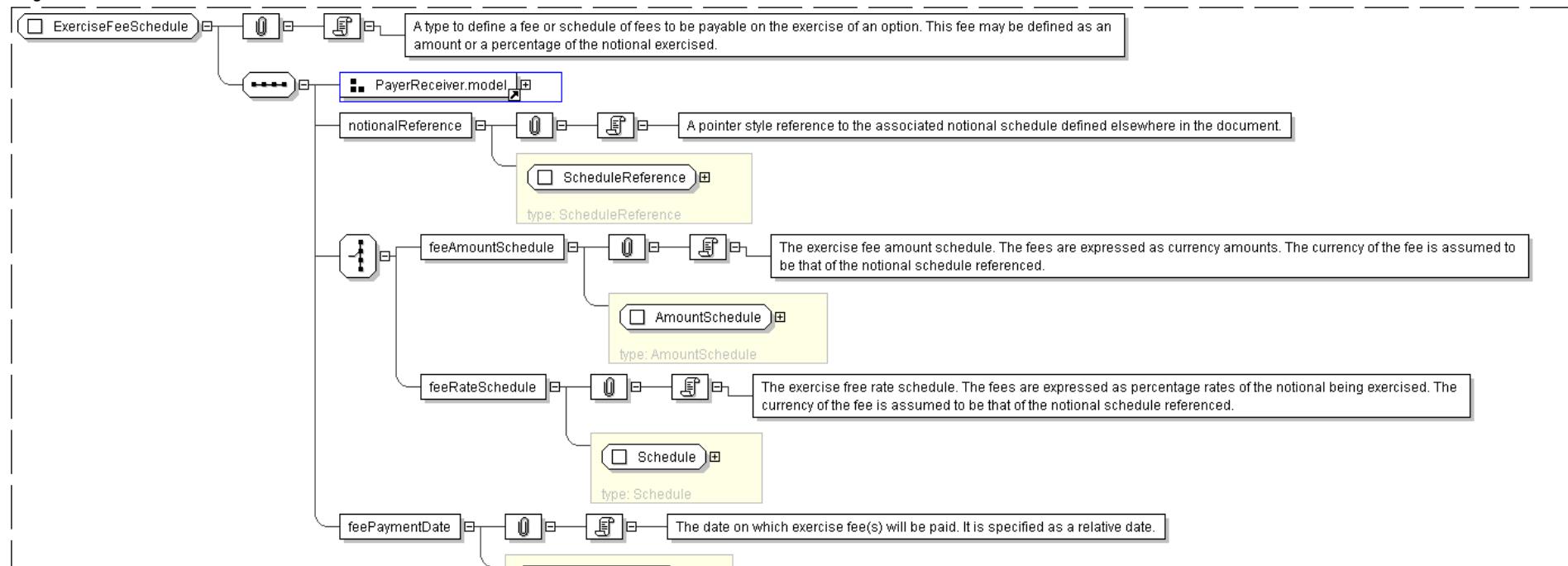
<notionalReference> ScheduleReference </notionalReference> [1]
'A pointer style reference to the associated notional schedule defined elsewhere in
the document.'

Start Choice [1]
  <feeAmountSchedule> AmountSchedule </feeAmountSchedule> [1]
    '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.'

  <feeRateSchedule> Schedule </feeRateSchedule> [1]
    'The exercise free 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.'

End Choice
<feePaymentDate> RelativeDateOffset </feePaymentDate> [1]
  'The date on which exercise fee(s) will be paid. It is specified as a relative date.'

</...>
```

**Diagram**

type: RelativeDateOffset

**Schema Component Representation**

```
<xsd:complexType name="ExerciseFeeSchedule">
  <xsd:sequence>
    <xsd:group ref=" PayerReceiver.model " />
    <xsd:element name="notionalReference" type=" ScheduleReference " />
    <xsd:choice>
      <xsd:element name="feeAmountSchedule" type=" AmountSchedule " />
      <xsd:element name="feeRateSchedule" type=" Schedule " />
    </xsd:choice>
    <xsd:element name="feePaymentDate" type=" RelativeDateOffset " />
  </xsd:sequence>
</xsd:complexType>
```

top

**Complex Type: ExerciseNotice**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None

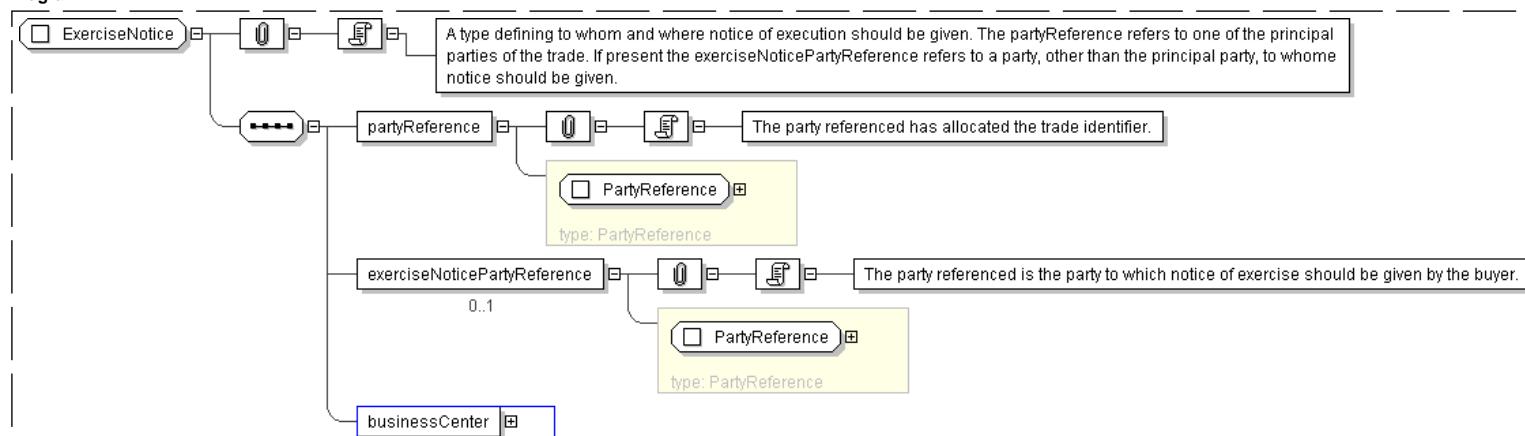
<b>Name</b>	ExerciseNotice
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">ManualExercise</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type defining to whom and where notice of execution should be given. The partyReference refers to one of the principal parties of the trade. If present the exerciseNoticePartyReference refers to a party, other than the principal party, to whom notice should be given.

**XML Instance Representation**

```
<...>
  <partyReference> PartyReference </partyReference> [1]
  'The party referenced has allocated the trade identifier.'

  <exerciseNoticePartyReference> PartyReference </exerciseNoticePartyReference> [0..1]
  'The party referenced is the party to which notice of exercise should be given by the buyer.'

  <businessCenter> BusinessCenter </businessCenter> [1]
</...>
```

**Diagram**

**Schema Component Representation**

```
<xsd:complexType name="ExerciseNotice">
  <xsd:sequence>
    <xsd:element name="partyReference" type="PartyReference" />
    <xsd:element name="exerciseNoticePartyReference" type="PartyReference" minOccurs="0" />
    <xsd:element name="businessCenter" type="BusinessCenter" />
  </xsd:sequence>
</xsd:complexType>
```

[top](#)**Complex Type: ExerciseProcedure**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None

<b>Name</b>	ExerciseProcedure
<b>Abstract</b>	no
<b>Documentation</b>	A type describing how notice of exercise should be given. This can be either manual or automatic.

**XML Instance Representation**

```
<...>
Start Choice [1]
<manualExercise> ManualExercise </manualExercise> [1]
'Specifies that the notice of exercise must be given by the buyer to the seller or seller
\'s agent.'

<automaticExercise> AutomaticExercise </automaticExercise> [1]
'If automatic 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 threshold rate. The term in-the-money is
assumed to have the meaning defining in the 2000 ISDA Definitions, Section 17.4 In-the-money.'
```

End Choice

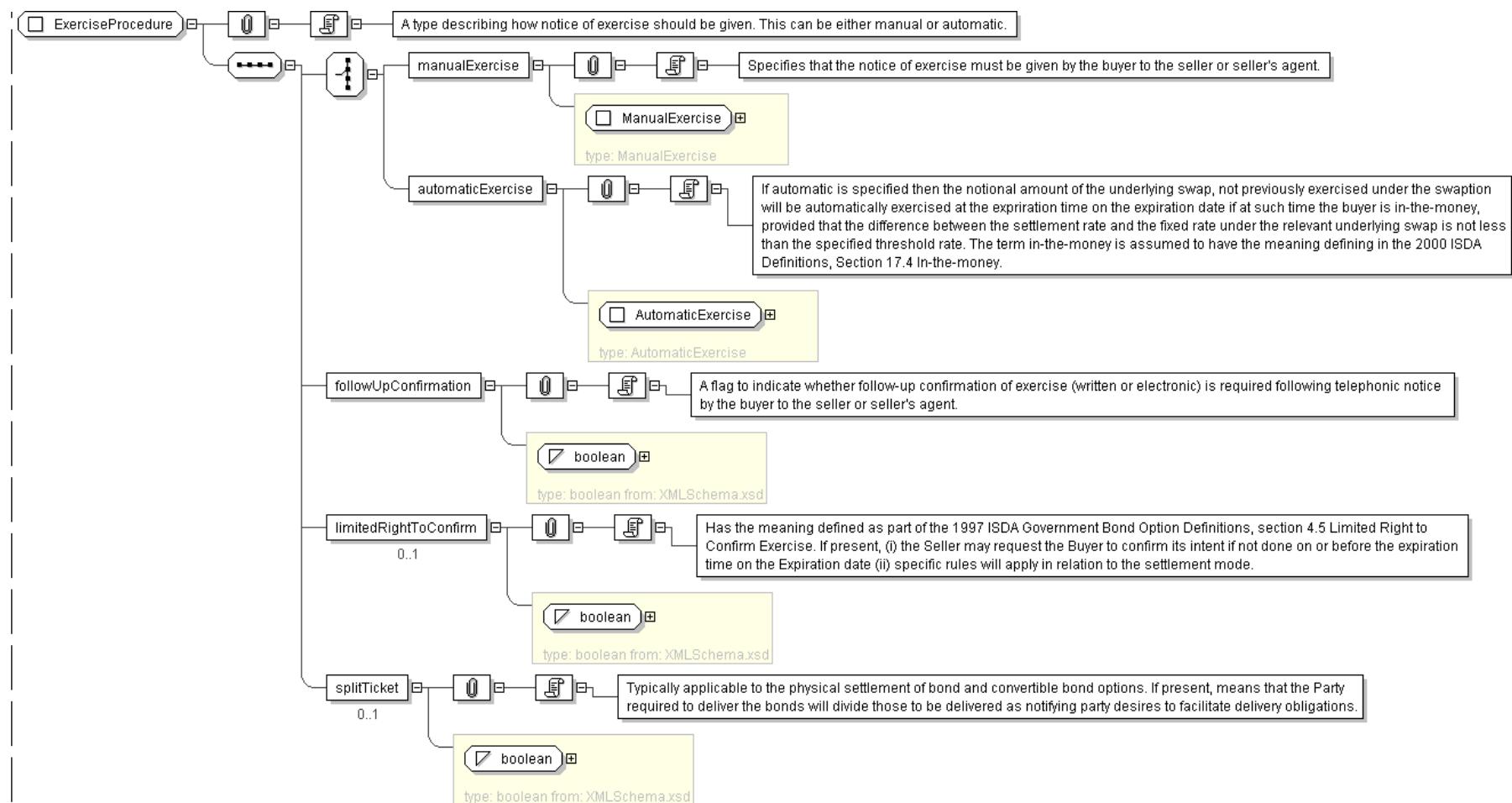
```
<followUpConfirmation> xsd:boolean </followUpConfirmation> [1]
'A flag to indicate whether follow-up confirmation of exercise (written or electronic)
is required following telephonic notice by the buyer to the seller or seller\'s agent.'
```

```
<limitedRightToConfirm> xsd:boolean </limitedRightToConfirm> [0..1]
'Has the meaning defined as part of the 1997 ISDA Government Bond Option Definitions,
section 4.5 Limited Right to Confirm Exercise. If present, (i) the Seller may request the
Buyer to confirm its intent if not done on or before the expiration time on the Expiration
date (ii) specific rules will apply in relation to the settlement mode.'
```

```
<splitTicket> xsd:boolean </splitTicket> [0..1]
'Typically applicable to the physical settlement of bond and convertible bond options.
If present, means that the Party required to deliver the bonds will divide those to
be delivered as notifying party desires to facilitate delivery obligations.'
```

&lt;/...&gt;

**Diagram**



#### Schema Component Representation

```

<xsd:complexType name="ExerciseProcedure">
  <xsd:sequence>
    <xsd:choice>
      <xsd:element name="manualExercise" type=" ManualExercise " />
      <xsd:element name="automaticExercise" type=" AutomaticExercise " />
    </xsd:choice>
    <xsd:element name="followUpConfirmation" type=" xsd:boolean " />
    <xsd:element name="limitedRightToConfirm" type=" xsd:boolean " minOccurs="0 " />
    <xsd:element name="splitTicket" type=" xsd:boolean " minOccurs="0 " />
  </xsd:sequence>
</xsd:complexType>

```

top

#### Complex Type: FloatingRate

Super-types:  
Sub-types:

[Rate < FloatingRate](#) (by extension)  
 • [FloatingRateCalculation](#) (by extension)

Name	FloatingRate
------	--------------

<b>Used by (from the same schema document)</b>	Complex Type <a href="#">StubValue</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type defining a floating rate.

**XML Instance Representation**

```

<...
  id=" xsd:ID [0..1]">
  <floatingRateIndex> FloatingRateIndex </floatingRateIndex> [1]
  <indexTenor> Period </indexTenor> [0..1]
  'The ISDA Designated Maturity, i.e. the tenor of the floating rate.'

  <floatingRateMultiplierschedule> Schedule </floatingRateMultiplierschedule> [0..1]
  'A rate multiplier or multiplier schedule to apply to the floating rate. A multiplier
  schedule is expressed as explicit multipliers and dates. In the case of a schedule, the
  step dates may be subject to adjustment in accordance with any adjustments specified in
  the calculationPeriodDatesAdjustments. The multiplier can be a positive or negative
  decimal. This element should only be included if the multiplier is not equal to 1 (one) for
  the term of the stream.'

  <spreadSchedule> SpreadSchedule </spreadSchedule> [0..*]
  '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> RateTreatmentEnum </rateTreatment> [0..1]
  '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.'

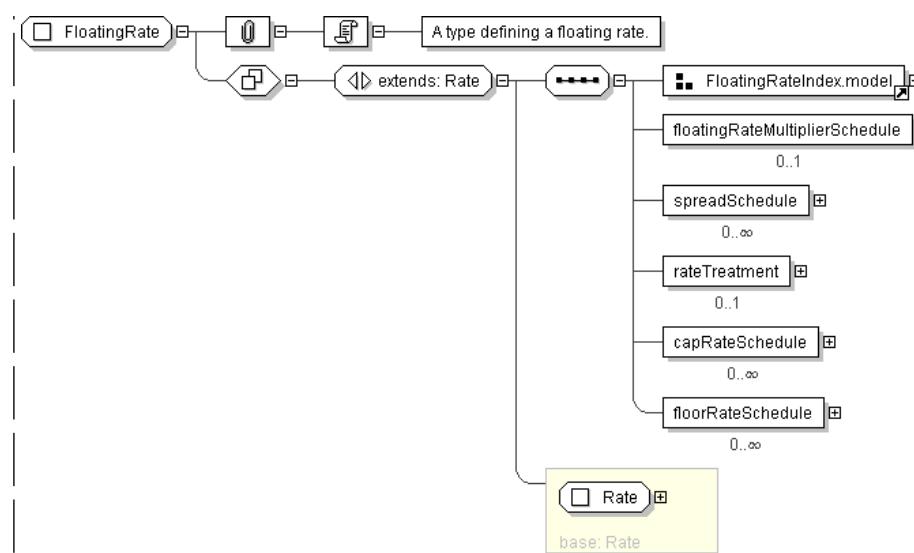
  <capRateSchedule> StrikeSchedule </capRateSchedule> [0..*]
  'The cap rate or cap rate schedule, if any, which applies to the floating rate. The cap
  rate (strike) is only required where the floating rate on a swap stream is capped at a
  certain level. A cap rate schedule is expressed as explicit cap 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> StrikeSchedule </floorRateSchedule> [0..*]
  '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.'

</...>

```

**Diagram**

**Schema Component Representation**

```

<xsd:complexType name="FloatingRate">
  <xsd:complexContent>
    <xsd:extension base=" Rate ">
      <xsd:sequence>
        <xsd:group ref=" FloatingRateIndex.model ">
          <xsd:element name="floatingRateMultiplierSchedule" type=" Schedule " minOccurs="0" />
          <xsd:element name="spreadSchedule" type=" SpreadSchedule " minOccurs="0" maxOccurs="unbounded" />
          <xsd:element name="rateTreatment" type=" RateTreatmentEnum " minOccurs="0" />
          <xsd:element name="capRateSchedule" type=" StrikeSchedule " minOccurs="0"
maxOccurs="unbounded" />
          <xsd:element name="floorRateSchedule" type=" StrikeSchedule "
minOccurs="0" maxOccurs="unbounded" />
        </xsd:sequence>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>
  
```

top

**Complex Type: FloatingRateCalculation**

<b>Super-types:</b>	<a href="#">Rate</a> < <a href="#">FloatingRate</a> (by extension) < <a href="#">FloatingRateCalculation</a> (by extension)
<b>Sub-types:</b>	None

<b>Name</b>	FloatingRateCalculation
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">InterestAccrualsMethod</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type defining the floating rate and definitions relating to the calculation of floating rate amounts.

**XML Instance Representation**

```

<...
  id=" xsd:ID [0..1]">
  <floatingRateIndex> FloatingRateIndex </floatingRateIndex> [1]
  <indexTenor> Period </indexTenor> [0..1]
  'The ISDA Designated Maturity, i.e. the tenor of the floating rate.'
  <floatingRateMultiplierSchedule> Schedule </floatingRateMultiplierSchedule> [0..1]
  
```

'A rate multiplier or multiplier schedule to apply to the floating rate. A multiplier schedule is expressed as explicit multipliers 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 multiplier can be a positive or negative decimal. This element should only be included if the multiplier is not equal to 1 (one) for the term of the stream.'

<spreadSchedule> SpreadSchedule </spreadSchedule> [0..\*]

'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> RateTreatmentEnum </rateTreatment> [0..1]

'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> StrikeSchedule </capRateSchedule> [0..\*]

'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 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> StrikeSchedule </floorRateSchedule> [0..\*]

'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.'

<initialRate> xsd:decimal </initialRate> [0..1]

'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> Rounding </finalRateRounding> [0..1]

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

<averagingMethod> AveragingMethodEnum </averagingMethod> [0..1]

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

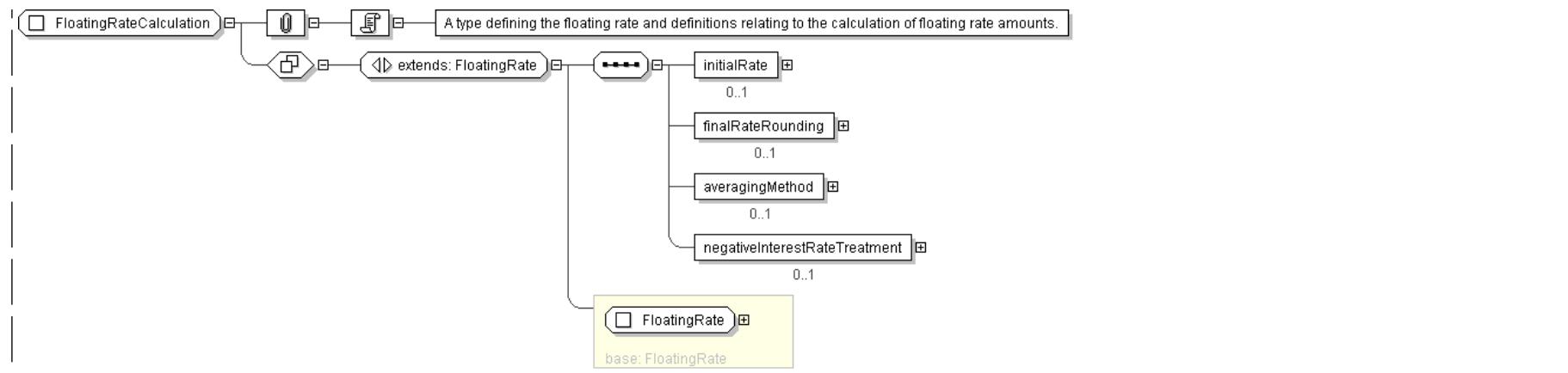
<negativeInterestRateTreatment> NegativeInterestRateTreatmentEnum

</negativeInterestRateTreatment> [0..1]

'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).'

</...>

**Diagram**

**Schema Component Representation**

```
<xsd:complexType name="FloatingRateCalculation">
  <xsd:complexContent>
    <xsd:extension base=" FloatingRate ">
      <xsd:sequence>
        <xsd:element name="initialRate" type=" xsd:decimal " minOccurs="0"/>
        <xsd:element name="finalRateRounding" type=" Rounding " minOccurs="0"/>
        <xsd:element name="averagingMethod" type=" AveragingMethodEnum " minOccurs="0"/>
        <xsd:element name="negativeInterestRateTreatment" type=" NegativeInterestRateTreatmentEnum
          " minOccurs="0"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

top

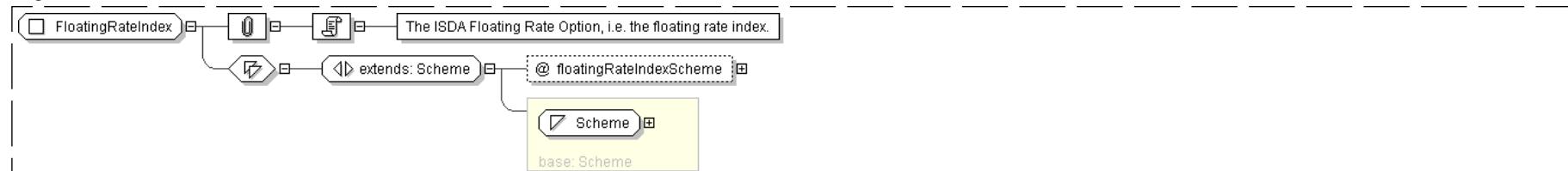
**Complex Type: [FloatingRateIndex](#)**

<b>Super-types:</b>	<a href="#">xsd:normalizedString</a> < <a href="#">Scheme</a> (by restriction) < <b>FloatingRateIndex</b> (by extension)
<b>Sub-types:</b>	None

<b>Name</b>	<b>FloatingRateIndex</b>
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">ForecastRateIndex</a> , Model Group <a href="#">FloatingRateIndex.model</a>
<b>Abstract</b>	no
<b>Documentation</b>	The ISDA Floating Rate Option, i.e. the floating rate index.

**XML Instance Representation**

```
<...
  floatingRateIndexScheme=" xsd:anyURI [0..1]>
  Scheme
</...>
```

**Diagram**

**Schema Component Representation**

```
<xsd:complexType name="FloatingRateIndex">
  <xsd:simpleContent>
    <xsd:extension base=" Scheme ">
      <xsd:attribute name="floatingRateIndexScheme" type=" xsd:anyURI " default="http://www.fpml.org/coding-scheme/floating-rate-index"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
```

[top](#)**Complex Type: ForecastRateIndex**

Super-types:	None
Sub-types:	None

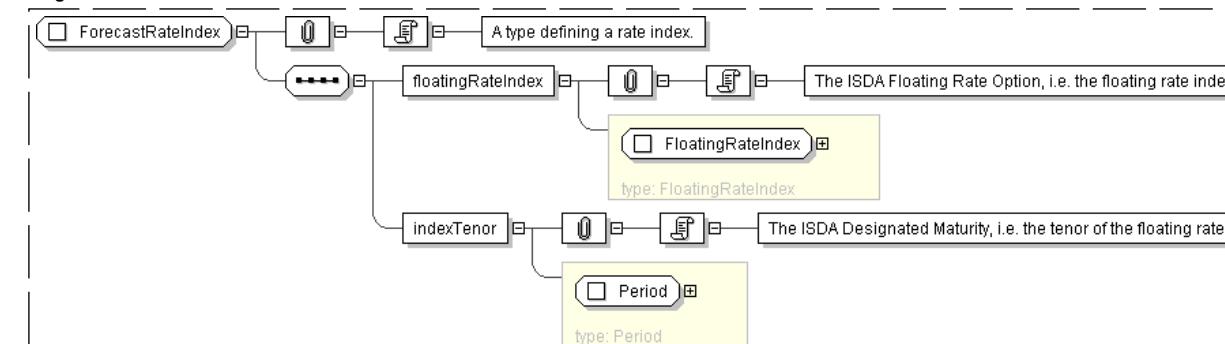
Name	ForecastRateIndex
Abstract	no
Documentation	A type defining a rate index.

**XML Instance Representation**

```
<...>
  <floatingRateIndex> FloatingRateIndex </floatingRateIndex> [1]
  'The ISDA Floating Rate Option, i.e. the floating rate index.'

  <indexTenor> Period </indexTenor> [1]
  'The ISDA Designated Maturity, i.e. the tenor of the floating rate.'

</...>
```

**Diagram****Schema Component Representation**

```
<xsd:complexType name="ForecastRateIndex">
  <xsd:sequence>
    <xsd:element name="floatingRateIndex" type=" FloatingRateIndex " />
    <xsd:element name="indexTenor" type=" Period " />
  </xsd:sequence>
</xsd:complexType>
```

[top](#)**Complex Type: Formula**

Super-types:	None
--------------	------

**Sub-types:** None

<b>Name</b>	Formula
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">FormulaComponent</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type describing a financial formula, with its description and components.

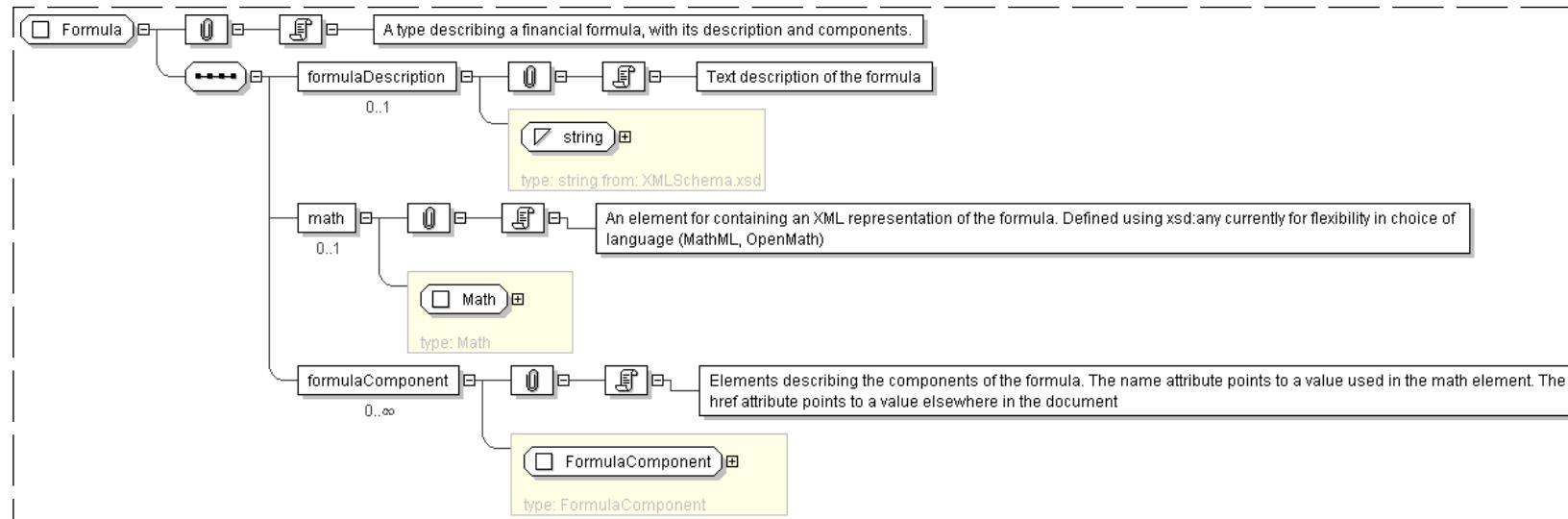
**XML Instance Representation**

```
<...>
<formulaDescription> xsd:string </formulaDescription> [0..1]
'Text description of the formula'

<math> Math </math> [0..1]
'An element for containing an XML representation of the formula. Defined using xsd: any currently for flexibility in choice of language (MathML, OpenMath)'

<formulaComponent> FormulaComponent </formulaComponent> [0..*]
'Elements describing the components of the formula. The name attribute points to a value used in the math element. The href attribute points to a value elsewhere in the document'

</...>
```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="Formula">
  <xsd:sequence>
    <xsd:element name="formulaDescription" type="xsd:string" minOccurs="0"/>
    <xsd:element name="math" type="Math" minOccurs="0"/>
    <xsd:element name="formulaComponent" type="FormulaComponent"
      minOccurs="0" maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>
  
```

top

**Complex Type: FormulaComponent**

**Super-types:** None

Sub-types:

None

Name	FormulaComponent
Used by (from the same schema document)	Complex Type <a href="#">Formula</a>
Abstract	no
Documentation	Elements describing the components of the formula. The name attribute points to a value used in the math element. The href attribute points to a numeric value defined elsewhere in the document that is used by the formula component.

**XML Instance Representation**

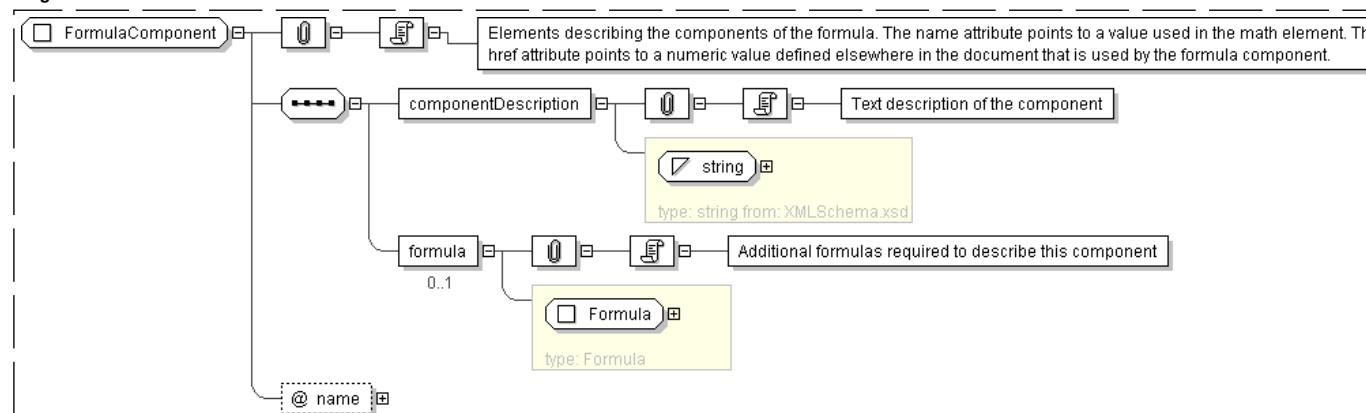
```

<...>
  name="xsd:normalizedString [0..1]">
    <componentDescription> xsd:string </componentDescription> [1]
      'Text description of the component'

    <formula> Formula </formula> [0..1]
      'Additional formulas required to describe this component'

  </...>

```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="FormulaComponent">
  <xsd:sequence>
    <xsd:element name="componentDescription" type="xsd:string" />
    <xsd:element name="formula" type="Formula" minOccurs="0"/>
  </xsd:sequence>
  <xsd:attribute name="name" type="xsd:normalizedString" />
</xsd:complexType>

```

[top](#)**Complex Type: Frequency**

Super-types:

None

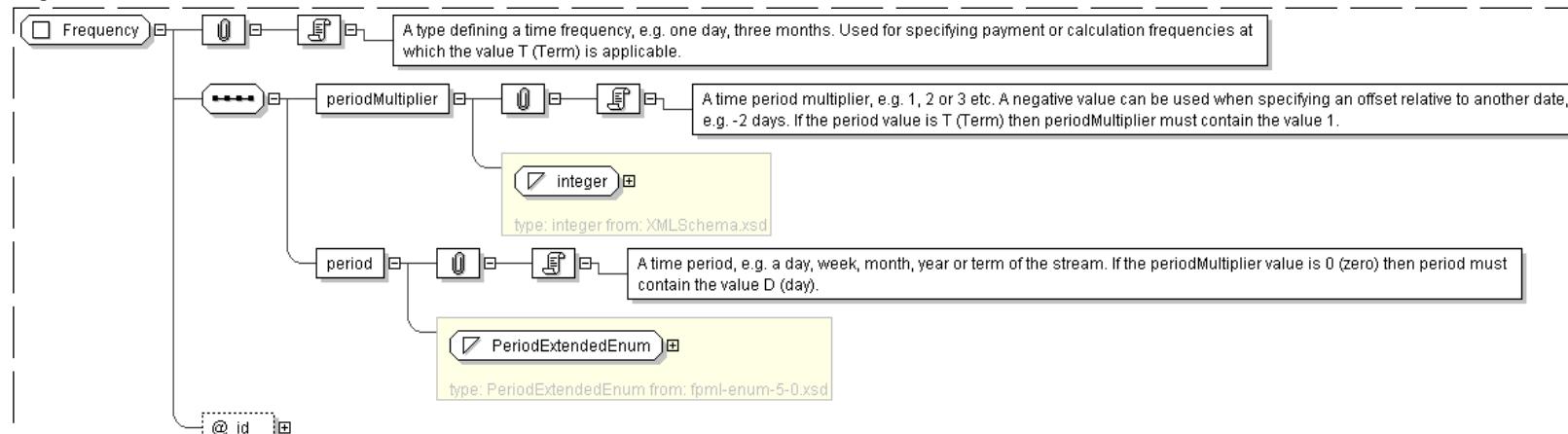
Sub-types:

- [CalculationPeriodFrequency](#) (by extension)
- [ResetFrequency](#) (by extension)

Name	Frequency
Abstract	no
Documentation	A type defining a time frequency, e.g. one day, three months. Used for specifying payment or calculation frequencies at which the value T (Term) is applicable.

**XML Instance Representation**

```
<...>
  id=" xsd:ID [0..1]">
    <periodMultiplier> xsd:integer </periodMultiplier> [1]
      '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> PeriodExtendedEnum </period> [1]
      '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).'
  </...>
```

**Diagram****Schema Component Representation**

```
<xsd:complexType name="Frequency">
  <xsd:sequence>
    <xsd:element name="periodMultiplier" type=" xsd:integer " />
    <xsd:element name="period" type=" PeriodExtendedEnum " />
  </xsd:sequence>
  <xsd:attribute name="id" type=" xsd:ID " />
</xsd:complexType>
```

top

**Complex Type: FutureValueAmount**

Super-types:	<a href="#">MoneyBase</a> < <a href="#">NonNegativeMoney</a> (by extension) < <b>FutureValueAmount</b> (by extension)
Sub-types:	None

Name	FutureValueAmount
Abstract	no
Documentation	A type defining a currency amount as at a future value date.

**XML Instance Representation**

```
<...>
  id=" xsd:ID [0..1]">
    <currency> Currency </currency> [1]
      'The currency in which an amount is denominated.'
    <amount> NonNegativeDecimal </amount> [1]
      'The non negative monetary quantity in currency units.'
```

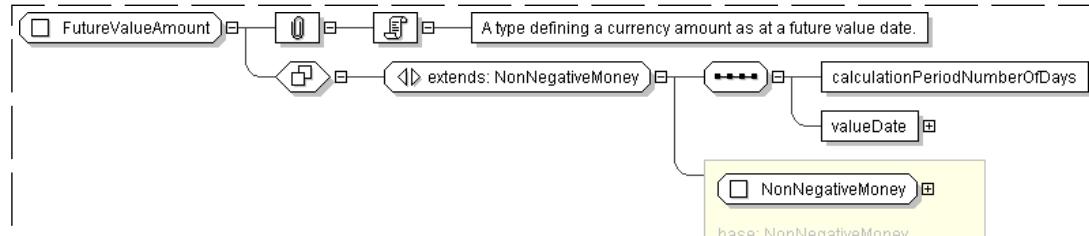
```

<calculationPeriodNumberOfDays> xsd:positiveInteger </calculationPeriodNumberOfDays> [1]
'The number of days from the adjusted calculation period start date to the adjusted value
date, calculated in accordance with the applicable day count fraction.'

<valueDate> xsd:date </valueDate> [1]
'Adjusted value date of the future value amount.'

</...>

```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="FutureValueAmount">
  <xsd:complexContent>
    <xsd:extension base=" NonNegativeMoney ">
      <xsd:sequence>
        <xsd:element name="calculationPeriodNumberOfDays" type=" xsd:positiveInteger ">
        <xsd:element name="valueDate" type=" xsd:date ">
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

top

**Complex Type: FxCashSettlement**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None

<b>Name</b>	FxCashSettlement
<b>Abstract</b>	no
<b>Documentation</b>	A type that is used for describing cash settlement of an option / non deliverable forward. It includes the currency to settle into together with the fixings required to calculate the currency amount.

**XML Instance Representation**

```

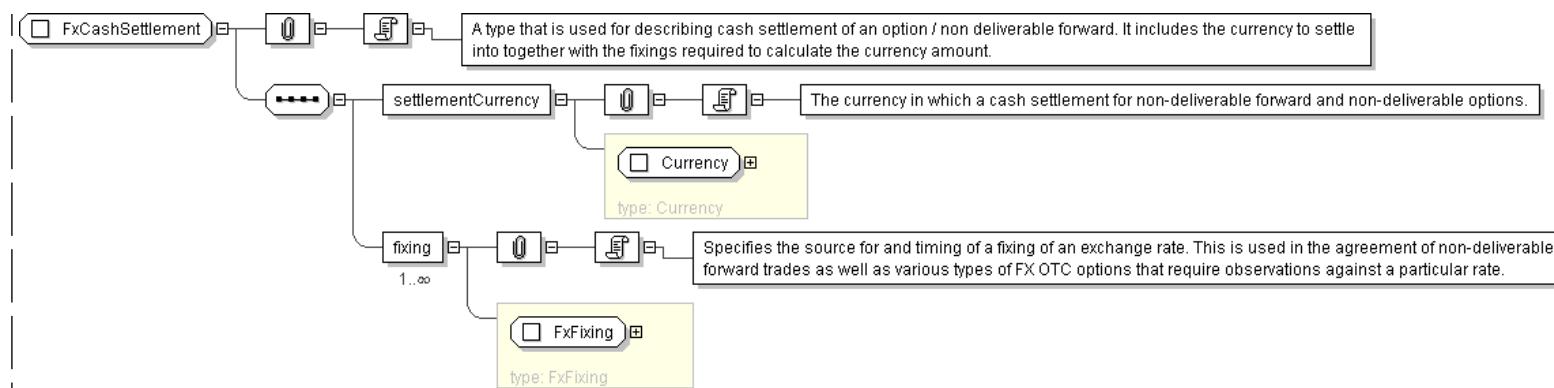
<...>
<settlementCurrency> Currency </settlementCurrency> [1]
'The currency in which a cash settlement for non-deliverable forward and non-
deliverable options.'

<fixing> FxFixing </fixing> [1..*]
'Specifies the source for and timing of a fixing of an exchange rate. This is used in
the agreement of non-deliverable forward trades as well as various types of FX OTC options
that require observations against a particular rate.'

</...>

```

**Diagram**



#### Schema Component Representation

```

<xsd:complexType name="FxCashSettlement">
  <xsd:sequence>
    <xsd:element name="settlementCurrency" type="Currency" />
    <xsd:element name="fixing" type="FxFixing" maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>
  
```

top

#### Complex Type: **FxFixing**

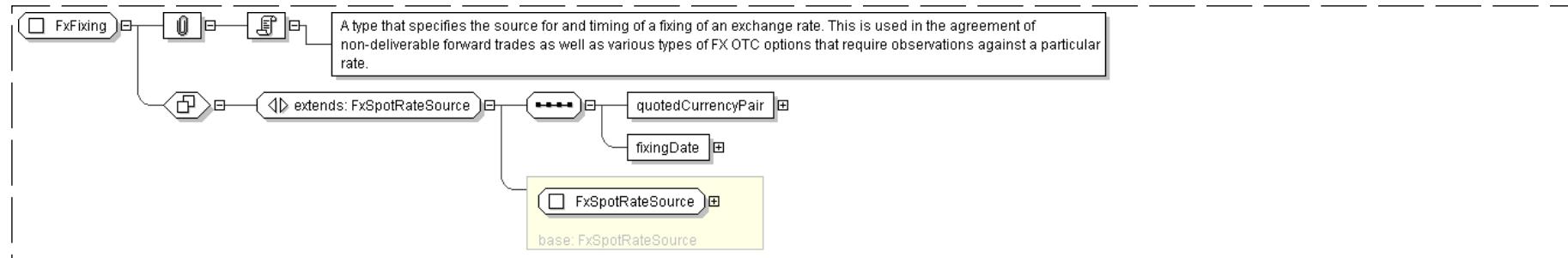
Super-types:	<a href="#">FxSpotRateSource</a> < <b>FxFixing</b> (by extension)
Sub-types:	None

Name	FxFixing
Used by (from the same schema document)	Complex Type <a href="#">FxCashSettlement</a>
Abstract	no
Documentation	A type that specifies the source for and timing of a fixing of an exchange rate. This is used in the agreement of non-deliverable forward trades as well as various types of FX OTC options that require observations against a particular rate.

#### XML Instance Representation

```

<...>
<primaryRateSource> InformationSource </primaryRateSource> [1]
  'The primary source for where the rate observation will occur. Will typically be either a page or a reference bank published rate.'
<secondaryRateSource> InformationSource </secondaryRateSource> [0..1]
  'An alternative, or secondary, source for where the rate observation will occur. Will typically be either a page or a reference bank published rate.'
<fixingTime> BusinessCenterTime </fixingTime> [1]
  '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. 11:00am London time.'
<quotedCurrencyPair> QuotedCurrencyPair </quotedCurrencyPair> [1]
  'Defines the two currencies for an FX trade and the quotation relationship between the two currencies.'
<fixingDate> xsd:date </fixingDate> [1]
  'Describes the specific date when a non-deliverable forward or non-deliverable option will \\"fix\\" against a particular rate, which will be used to compute the ultimate cash settlement.'
</...>
  
```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="FxFixing">
  <xsd:complexContent>
    <xsd:extension base=" FxSpotRateSource ">
      <xsd:sequence>
        <xsd:element name="quotedCurrencyPair" type=" QuotedCurrencyPair "/>
        <xsd:element name="fixingDate" type=" xsd:date "/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
  
```

[top](#)**Complex Type: FxRate**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None

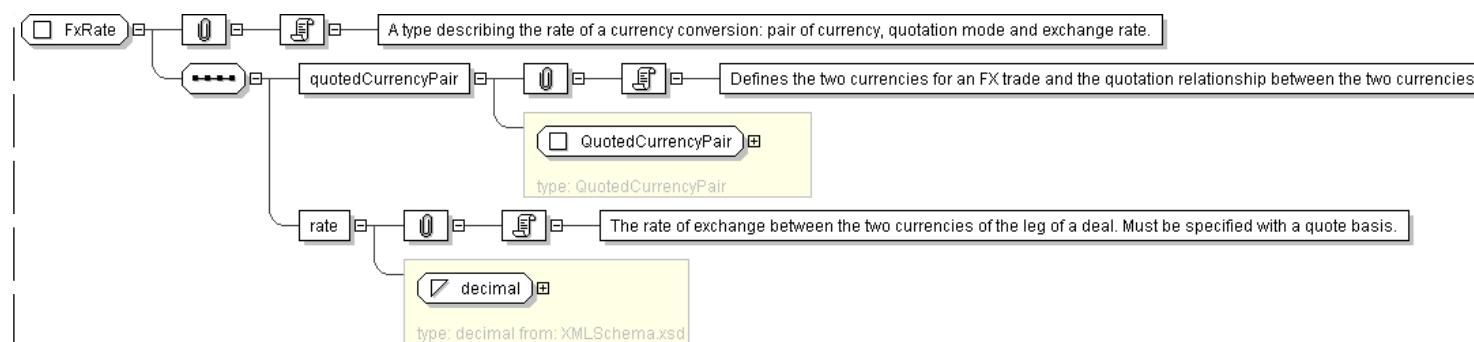
<b>Name</b>	FxRate
<b>Abstract</b>	no
<b>Documentation</b>	A type describing the rate of a currency conversion: pair of currency, quotation mode and exchange rate.

**XML Instance Representation**

```

<...>
  <quotedCurrencyPair> QuotedCurrencyPair </quotedCurrencyPair> [1]
  'Defines the two currencies for an FX trade and the quotation relationship between the
  two currencies.'
  <rate> xsd:decimal </rate> [1]
  'The rate of exchange between the two currencies of the leg of a deal. Must be specified with
  a quote basis.'
</...>
  
```

**Diagram**

**Schema Component Representation**

```

<xsd:complexType name="FxRate">
  <xsd:sequence>
    <xsd:element name="quotedCurrencyPair" type=" QuotedCurrencyPair " />
    <xsd:element name="rate" type=" xsd:decimal " />
  </xsd:sequence>
</xsd:complexType>
  
```

[top](#)**Complex Type: FxSpotRateSource**

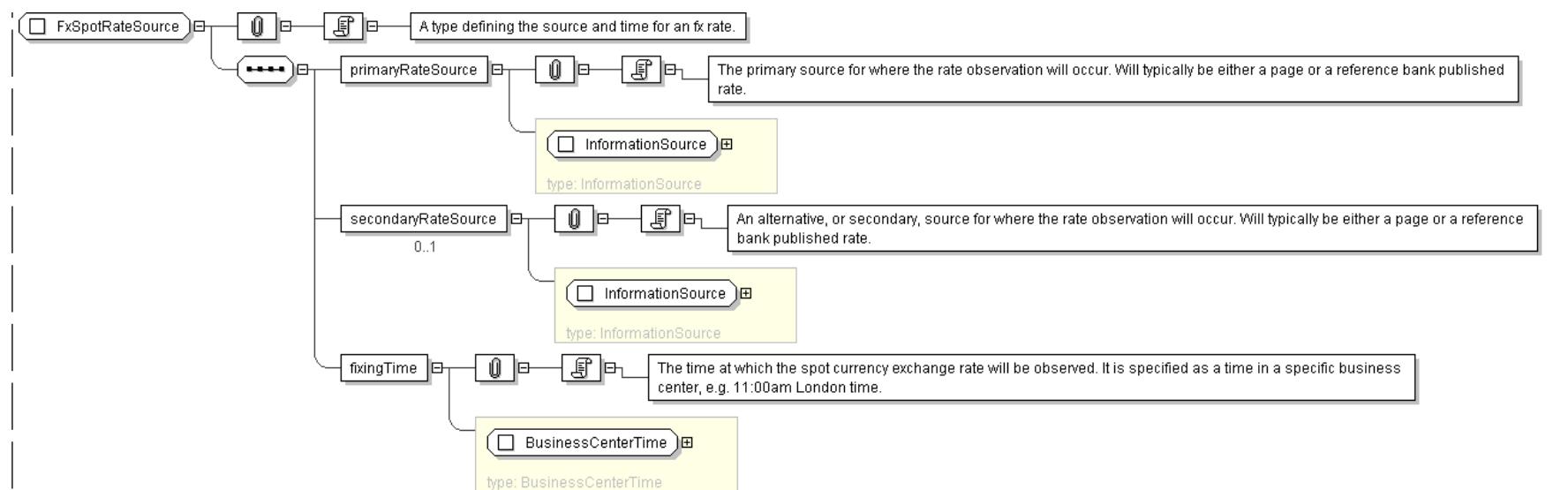
<b>Super-types:</b>	None
<b>Sub-types:</b>	<ul style="list-style-type: none"> <li>• <a href="#">FxFixing</a> (by extension)</li> </ul>
<b>Name</b>	FxSpotRateSource
<b>Abstract</b>	no
<b>Documentation</b>	A type defining the source and time for an fx rate.

**XML Instance Representation**

```

<...>
<primaryRateSource> InformationSource </primaryRateSource> [1]
  'The primary source for where the rate observation will occur. Will typically be either a page or a reference bank published rate.'
<secondaryRateSource> InformationSource </secondaryRateSource> [0..1]
  'An alternative, or secondary, source for where the rate observation will occur. Will typically be either a page or a reference bank published rate.'
<fixingTime> BusinessCenterTime </fixingTime> [1]
  '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. 11:00am London time.'
</...>
  
```

**Diagram**



#### Schema Component Representation

```

<xsd:complexType name="FxSpotRateSource">
  <xsd:sequence>
    <xsd:element name="primaryRateSource" type="InformationSource" />
    <xsd:element name="secondaryRateSource" type="InformationSource" minOccurs="0"/>
    <xsd:element name="fixingTime" type="BusinessCenterTime" />
  </xsd:sequence>
</xsd:complexType>
  
```

top

#### Complex Type: GenericAgreement

Super-types:	None
Sub-types:	None
Name	GenericAgreement
Used by (from the same schema document)	Complex Type <a href="#">PartyRelationshipDocumentation</a>
Abstract	no
Documentation	An entity for defining a generic agreement executed between two parties for any purpose.

#### XML Instance Representation

```

<...>
<type> AgreementType </type> [1]
'The type of agreement executed between the parties.'

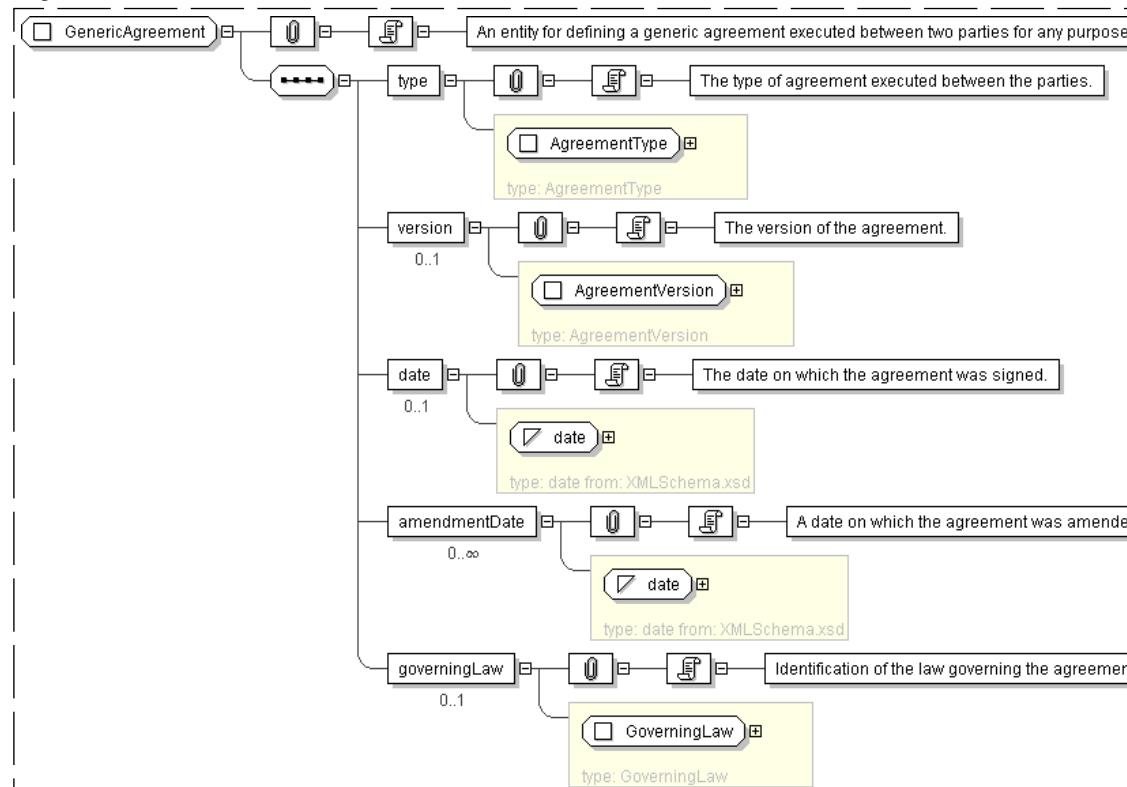
<version> AgreementVersion </version> [0..1]
'The version of the agreement.'

<date> xsd:date </date> [0..1]
'The date on which the agreement was signed.'

<amendmentDate> xsd:date </amendmentDate> [0..*]
'A date on which the agreement was amended.'

<governingLaw> GoverningLaw </governingLaw> [0..1]
'Identification of the law governing the agreement.'
  
```

&lt;/...&gt;

**Diagram****Schema Component Representation**

```

<xsd:complexType name="GenericAgreement">
  <xsd:sequence>
    <xsd:element name="type" type=" AgreementType "/>
    <xsd:element name="version" type=" AgreementVersion " minOccurs="0"/>
    <xsd:element name="date" type=" xsd:date " minOccurs="0"/>
    <xsd:element name="amendmentDate" type=" xsd:date " minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element name="governingLaw" type=" GoverningLaw " minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>
  
```

top

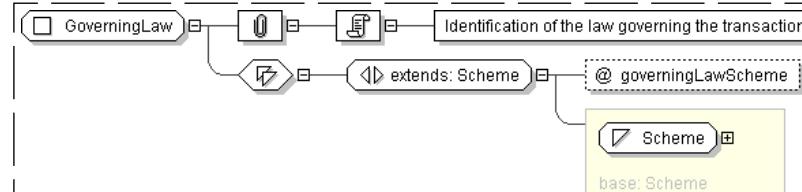
**Complex Type: GoverningLaw**

<b>Super-types:</b>	xsd:normalizedString < <a href="#">Scheme</a> (by restriction) < <b>GoverningLaw</b> (by extension)
<b>Sub-types:</b>	None

<b>Name</b>	GoverningLaw
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">GenericAgreement</a> , Model Group <a href="#">PartyInformation.model</a>
<b>Abstract</b>	no
<b>Documentation</b>	Identification of the law governing the transaction.

**XML Instance Representation**

```
<...
  governingLawScheme=" xsd:anyURI [0..1]">
  Scheme
</...>
```

**Diagram****Schema Component Representation**

```
<xsd:complexType name="GoverningLaw">
  <xsd:simpleContent>
    <xsd:extension base=" Scheme ">
      <xsd:attribute name="governingLawScheme" type=" xsd:anyURI " default="http://www.fpml.org/coding-scheme/governing-law"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
```

[top](#)**Complex Type: GrossCashflow**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None

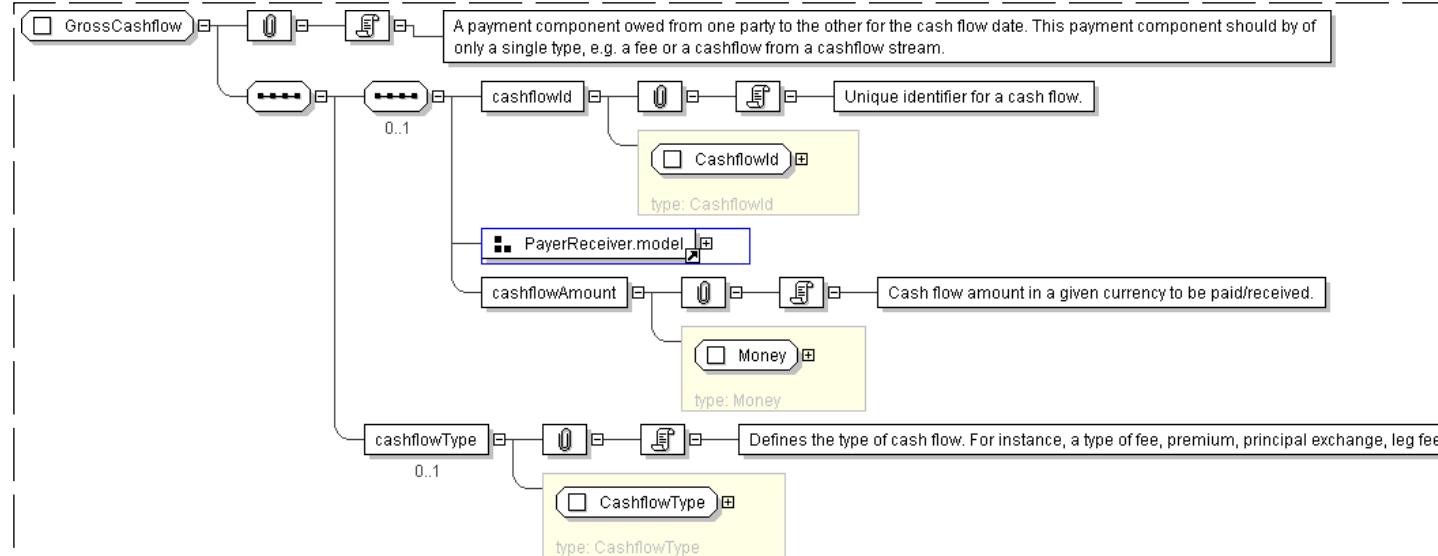
<b>Name</b>	GrossCashflow
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">PaymentDetails</a>
<b>Abstract</b>	no
<b>Documentation</b>	A payment component owed from one party to the other for the cash flow date. This payment component should be of only a single type, e.g. a fee or a cashflow from a cashflow stream.

**XML Instance Representation**

```
<...>
Start Sequence [0..1]
  <cashflowId> CashflowId </cashflowId> [1]
  'Unique identifier for a cash flow.'
  <payerPartyReference> PartyReference </payerPartyReference> [1]
  'A reference to the party responsible for making the payments defined by this structure.'
  <payerAccountReference> AccountReference </payerAccountReference> [0..1]
  'A reference to the account responsible for making the payments defined by this structure.'
  <receiverPartyReference> PartyReference </receiverPartyReference> [1]
  'A reference to the party that receives the payments corresponding to this structure.'
  <receiverAccountReference> AccountReference </receiverAccountReference> [0..1]
  'A reference to the account that receives the payments corresponding to this structure.'
  <cashflowAmount> Money </cashflowAmount> [1]
  'Cash flow amount in a given currency to be paid/received.'
End Sequence
<cashflowType> CashflowType </cashflowType> [0..1]
```

'Defines the type of cash flow. For instance, a type of fee, premium, principal exchange, leg fee.'

&lt;...&gt;

**Diagram****Schema Component Representation**

```

<xsd:complexType name="GrossCashflow">
  <xsd:sequence>
    <xsd:sequence minOccurs="0">
      <xsd:element name="cashflowId" type=" CashflowID " />
      <xsd:group ref=" PayerReceiver.model " />
      <xsd:element name="cashflowAmount" type=" Money " />
    </xsd:sequence>
    <xsd:element name="cashflowType" type=" CashflowType " minOccurs="0" />
  </xsd:sequence>
</xsd:complexType>
  
```

top

**Complex Type: IdentifiedCurrency**

Super-types:	<a href="#">xsd:normalizedString</a> < <a href="#">Scheme</a> (by restriction) < <a href="#">Currency</a> (by extension) < <b>IdentifiedCurrency</b> (by extension)
Sub-types:	None

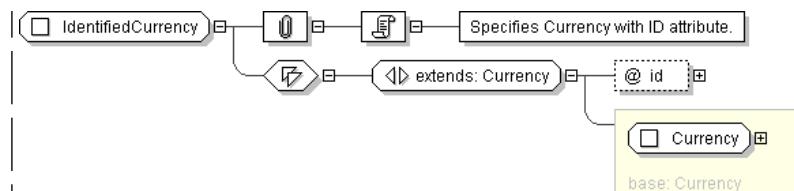
Name	IdentifiedCurrency
Abstract	no
Documentation	Specifies Currency with ID attribute.

**XML Instance Representation**

```

<...
  currencySchemes=" xsd:anyURI [0..1]" 
  id=" xsd:ID [0..1]">
  Currency
</...>
  
```

**Diagram**

**Schema Component Representation**

```

<xsd:complexType name="IdentifiedCurrency">
  <xsd:simpleContent>
    <xsd:extension base=" Currency ">
      <xsd:attribute name="id" type=" xsd:ID ">
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
  
```

top

**Complex Type: IdentifiedCurrencyReference**

**Super-types:** [Reference](#) < `IdentifiedCurrencyReference` (by extension)

**Sub-types:** None

**Name** `IdentifiedCurrencyReference`

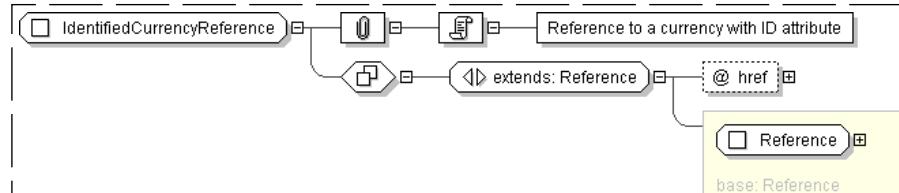
**Abstract** no

**Documentation** Reference to a currency with ID attribute

**XML Instance Representation**

```

<...
  href=" xsd:IDREF [1]" />
  
```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="IdentifiedCurrencyReference">
  <xsd:complexContent>
    <xsd:extension base=" Reference ">
      <xsd:attribute name="href" type=" xsd:IDREF " use="required" reference="IdentifiedCurrency"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
  
```

top

**Complex Type: IdentifiedDate**

**Super-types:** [xsd:date](#) < `IdentifiedDate` (by extension)

**Sub-types:** None

**Name** `IdentifiedDate`

**Used by (from the same schema document)**

Complex Type [AdjustableDate2](#) , Complex Type [AdjustableDate2](#) , Complex Type [AdjustableDates](#) , Complex Type [AdjustableDates](#) , Complex Type [AdjustableOrAdjustedDate](#) , Complex Type [RelativeDateOffset](#) , Model Group [AdjustableDate.model](#) , Model Group [AdjustableDate.model](#) , Model Group [VersionHistory.model](#)

**Abstract**

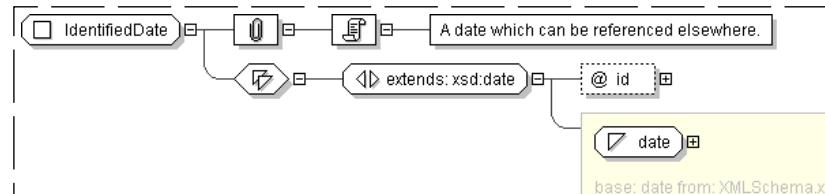
no

**Documentation**

A date which can be referenced elsewhere.

**XML Instance Representation**

```
<...  
id=" xsd:ID [0..1]">  
xsd:date  
</...>
```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="IdentifiedDate">
  <xsd:simpleContent>
    <xsd:extension base=" xsd:date ">
      <xsd:attribute name="id" type=" xsd:ID ">
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
  
```

top

**Complex Type: IdentifiedPayerReceiver****Super-types:**[PayerReceiverEnum](#) < **IdentifiedPayerReceiver** (by extension)**Sub-types:**

None

**Name**

IdentifiedPayerReceiver

**Used by (from the same schema document)**Complex Type [Strike](#) , Complex Type [Strike](#) , Complex Type [StrikeSchedule](#) , Complex Type [StrikeSchedule](#)**Abstract**

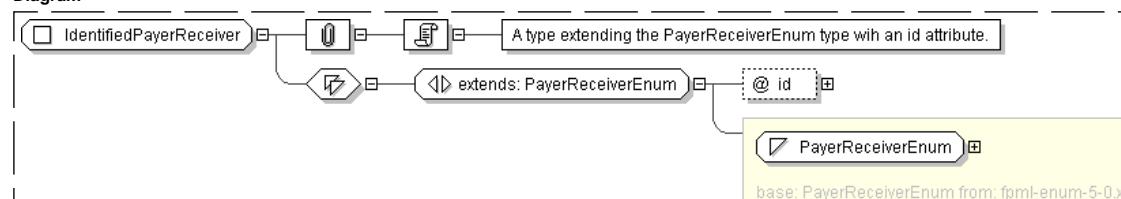
no

**Documentation**

A type extending the PayerReceiverEnum type with an id attribute.

**XML Instance Representation**

```
<...  
id=" xsd:ID [0..1]">  
PayerReceiverEnum  
</...>
```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="IdentifiedPayerReceiver">
  <xsd:simpleContent>
    <xsd:extension base=" PayerReceiverEnum ">
      <xsd:attribute name="id" type=" xsd:ID " />
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>

```

top

## Complex Type: IndustryClassification

Super-types:	<a href="#">xsd:normalizedString</a> < <a href="#">Scheme</a> (by restriction) < <b>IndustryClassification</b> (by extension)
Sub-types:	None

Name	IndustryClassification
Used by (from the same schema document)	Model Group <a href="#">PartyInformation.model</a>
Abstract	no
Documentation	A party's industry sector classification.

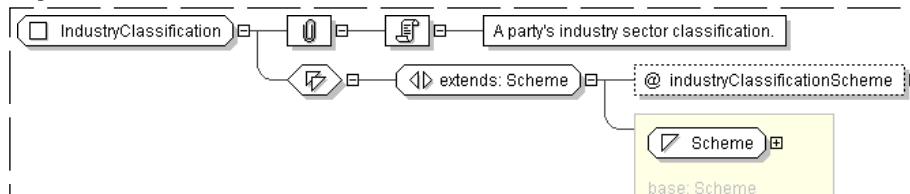
### XML Instance Representation

```

<...
industryClassificationScheme=" xsd:anyURI [0..1]">
  Scheme
</...>

```

### Diagram



### Schema Component Representation

```

<xsd:complexType name="IndustryClassification">
  <xsd:simpleContent>
    <xsd:extension base=" Scheme ">
      <xsd:attribute name="industryClassificationScheme" type=" xsd:anyURI " default="http://www.
        fpmi.org/ext/north-american-industry-classification-system"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>

```

top

## Complex Type: InformationProvider

Super-types:	<a href="#">xsd:normalizedString</a> < <a href="#">Scheme</a> (by restriction) < <b>InformationProvider</b> (by extension)
Sub-types:	None

Name	InformationProvider
Used by (from the same schema document)	Complex Type <a href="#">InformationSource</a>
Abstract	no

### XML Instance Representation

```

<...
| 

```

```
informationProviderScheme=" xsd:anyURI [ 0..1 ]"
| Scheme
|</...>
```

**Diagram****Schema Component Representation**

```
<xsd:complexType name="InformationProvider">
  <xsd:simpleContent>
    <xsd:extension base=" Scheme ">
      <xsd:attribute name="informationProviderScheme" type=" xsd:anyURI " default="http://www.
        fpml.org/coding-scheme/information-provider"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
```

[top](#)**Complex Type: InformationSource**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None

<b>Name</b>	InformationSource
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">FxSpotRateSource</a> , Complex Type <a href="#">FxSpotRateSource</a> , Complex Type <a href="#">SettlementRateSource</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type defining the source for a piece of information (e.g. a rate refix or an fx fixing).

**XML Instance Representation**

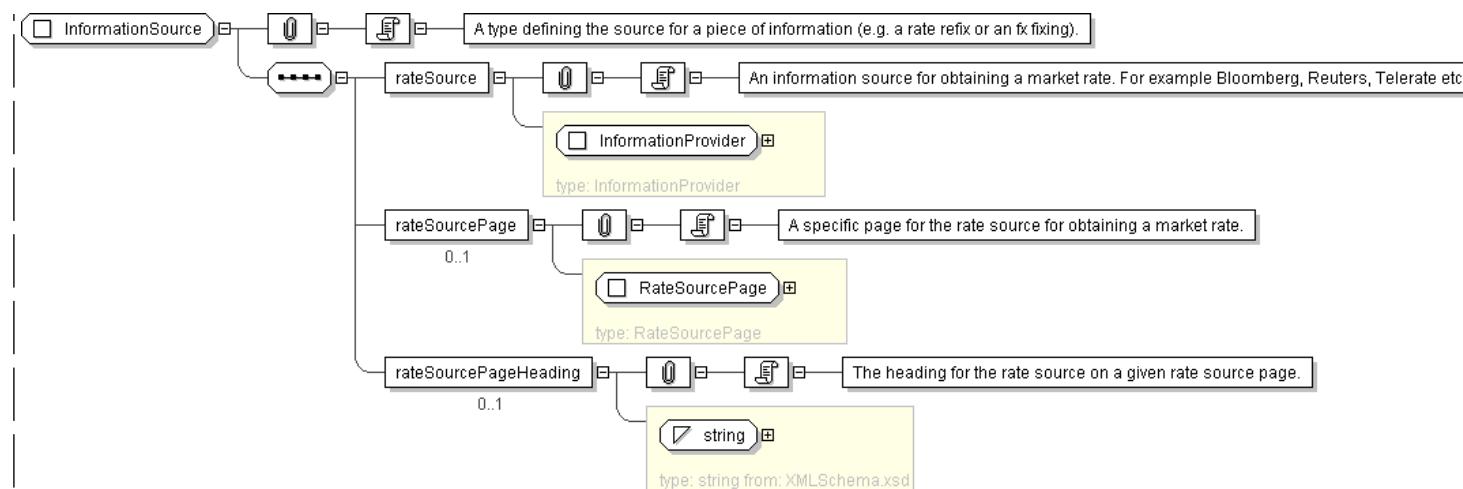
```
<...>
<rateSource> InformationProvider </rateSource> [1]
'An information source for obtaining a market rate. For example Bloomberg, Reuters,
Telerate etc.'
```

```
<rateSourcePage> RateSourcePage </rateSourcePage> [0..1]
'A specific page for the rate source for obtaining a market rate.'
```

```
<rateSourcePageHeading> xsd:string </rateSourcePageHeading> [0..1]
'The heading for the rate source on a given rate source page.'
```

```
</...>
```

**Diagram**

**Schema Component Representation**

```

<xsd:complexType name="InformationSource">
  <xsd:sequence>
    <xsd:element name="rateSource" type="InformationProvider" />
    <xsd:element name="rateSourcePage" type="RateSourcePage" minOccurs="0"/>
    <xsd:element name="rateSourcePageHeading" type="xsd:string" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>
  
```

[top](#)**Complex Type: InstrumentId**

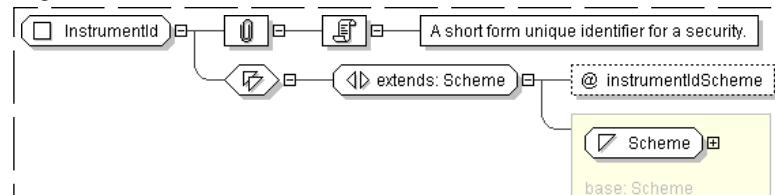
Super-types:	<code>xsd:normalizedString</code> < <a href="#">Scheme</a> > (by restriction) < <b>InstrumentId</b> > (by extension)
Sub-types:	None

Name	InstrumentId
Abstract	no
Documentation	A short form unique identifier for a security.

**XML Instance Representation**

```

<...
instrumentIdScheme="xsd:anyURI [1]">
Scheme
</...>
  
```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="InstrumentId">
  <xsd:simpleContent>
    <xsd:extension base="Scheme" />
  </xsd:simpleContent>
</xsd:complexType>
  
```

```

<xsd:attribute name="instrumentIdScheme" type="xsd:anyURI" use="required"/>
</xsd:extension>
</xsd:simpleContent>
</xsd:complexType>

```

## Complex Type: InterestAccrualsCompoundingMethod

**Super-types:** [InterestAccrualsMethod](#) < **InterestAccrualsCompoundingMethod** (by extension)

**Sub-types:** None

**Name** InterestAccrualsCompoundingMethod

**Abstract** no

**Documentation** A type defining the way in which interests are accrued: the applicable rate (fixed or floating reference) and the compounding method.

### XML Instance Representation

```

<...>
Start Choice [1]
  <floatingRateCalculation> FloatingRateCalculation </floatingRateCalculation> [1]
  'The floating rate calculation definitions'

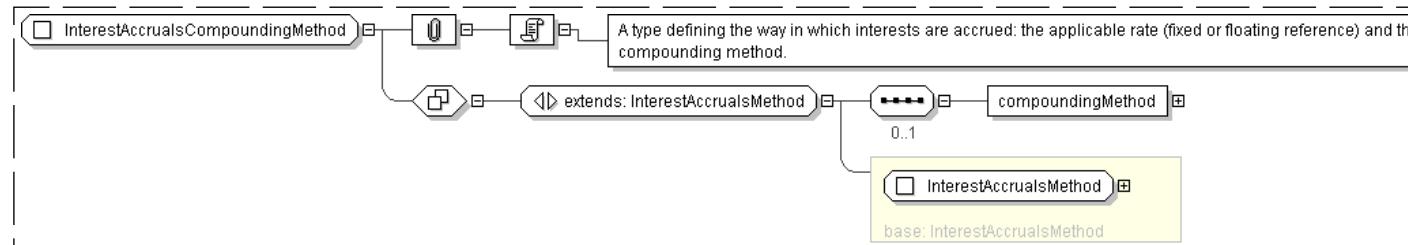
  <fixedRate> xsd:decimal </fixedRate> [1]
  'The calculation period fixed rate. A per annum rate, expressed as a decimal. A fixed rate
  of 5% would be represented as 0.05.'

End Choice
Start Sequence [0..1]
  <compoundingMethod> CompoundingMethodEnum </compoundingMethod> [1]
  '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.'

End Sequence
</...>

```

### Diagram



### Schema Component Representation

```

<xsd:complexType name="InterestAccrualsCompoundingMethod">
  <xsd:complexContent>
    <xsd:extension base=" InterestAccrualsMethod ">
      <xsd:sequence minOccurs="0">
        <xsd:element name="compoundingMethod" type="CompoundingMethodEnum" />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

**Complex Type: InterestAccrualsMethod**

Super-types:

None

Sub-types:

- [InterestAccrualsCompoundingMethod](#) (by extension)

Name

InterestAccrualsMethod

Abstract

no

Documentation

A type describing the method for accruing interests on dividends. Can be either a fixed rate reference or a floating rate reference.

**XML Instance Representation**

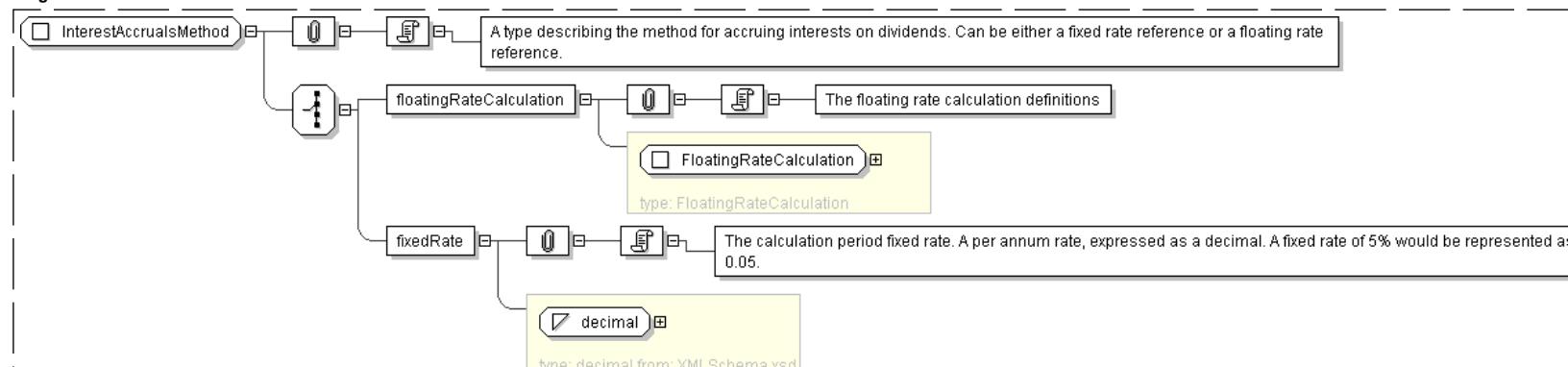
```

<...>
Start Choice [1]
  <floatingRateCalculation> FloatingRateCalculation </floatingRateCalculation> [1]
    'The floating rate calculation definitions'

  <fixedRate> xsd:decimal </fixedRate> [1]
    'The calculation period fixed rate. A per annum rate, expressed as a decimal. A fixed rate
    of 5% would be represented as 0.05.'

End Choice
</...>

```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="InterestAccrualsMethod">
  <xsd:choice>
    <xsd:element name="floatingRateCalculation" type="FloatingRateCalculation" />
    <xsd:element name="fixedRate" type="xsd:decimal" />
  </xsd:choice>
</xsd:complexType>

```

[top](#)**Complex Type: IntermediaryInformation**

Super-types:

None

Sub-types:

None

Name

IntermediaryInformation

Used by (from the same schema document)

Complex Type [SettlementInstruction](#)

Abstract

no

Documentation

A type that describes the information to identify an intermediary through which payment will be made by the correspondent bank to the ultimate beneficiary of the funds.

**XML Instance Representation**

```
<...>
Start Choice [1]
<routingIds> RoutingIds </routingIds> [1]
'A set of unique identifiers for a party, each one identifying the party within a payment system. The assumption is that each party will not have more than one identifier within the same payment system.' 

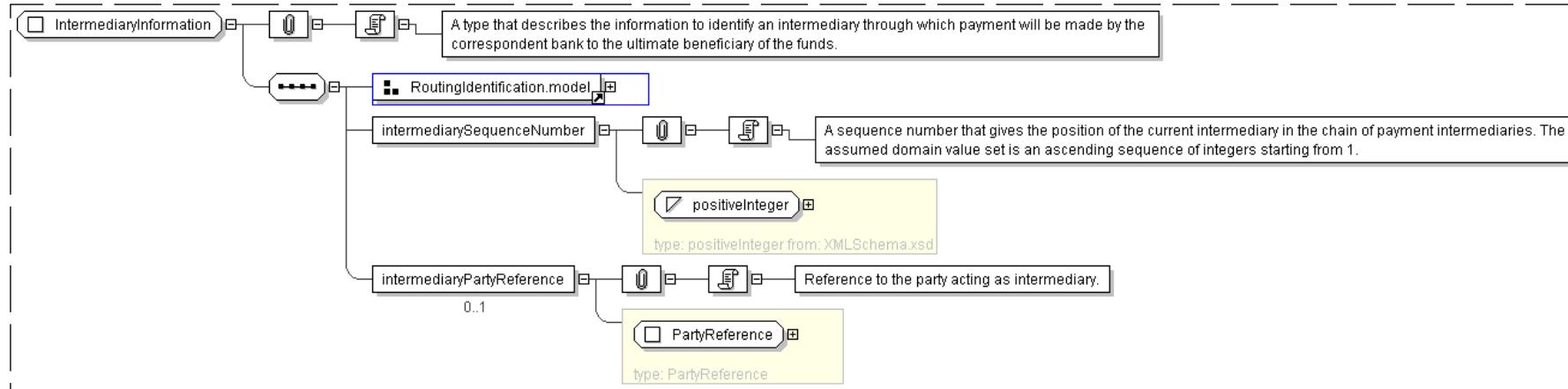
<routingExplicitDetails> RoutingExplicitDetails </routingExplicitDetails> [1]
'A set of details that is used to identify a party involved in the routing of a payment when the party does not have a code that identifies it within one of the recognized payment systems.' 

<routingIdsAndExplicitDetails> RoutingIdsAndExplicitDetails </routingIdsAndExplicitDetails> [1]
'A combination of coded payment system identifiers and details for physical addressing for a party involved in the routing of a payment.' 

End Choice
<intermediarySequenceNumber> xsd:positiveInteger </intermediarySequenceNumber> [1]
'A sequence number that gives the position of the current intermediary in the chain of payment intermediaries. The assumed domain value set is an ascending sequence of integers starting from 1.' 

<intermediaryPartyReference> PartyReference </intermediaryPartyReference> [0..1]
'Reference to the party acting as intermediary.' 

</...>
```

**Diagram****Schema Component Representation**

```
<xsd:complexType name="IntermediaryInformation">
  <xsd:sequence>
    <xsd:group ref="#">
      <xsd:element name="intermediarySequenceNumber" type="xsd:positiveInteger" />
      <xsd:element name="intermediaryPartyReference" type="PartyReference" minOccurs="0"/>
    </xsd:sequence>
</xsd:complexType>
```

top

**Complex Type: InterpolationMethod**

Super-types:

xsd:normalizedString < Scheme (by restriction) < **InterpolationMethod** (by extension)

Sub-types:

None

**Name****Abstract****Documentation**

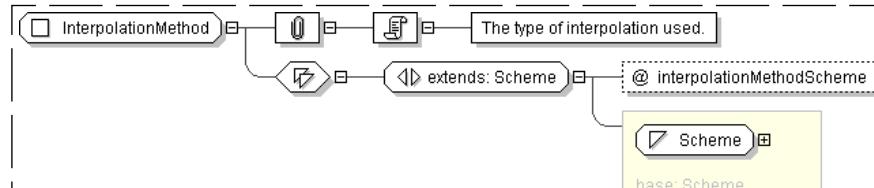
InterpolationMethod

no

The type of interpolation used.

**XML Instance Representation**

```
<...>
<interpolationMethodScheme=" xsd:anyURI [0..1]">
  Scheme
</...>
```

**Diagram****Schema Component Representation**

```
<xsd:complexType name="InterpolationMethod">
  <xsd:simpleContent>
    <xsd:extension base=" Scheme ">
      <xsd:attribute name="interpolationMethodScheme" type=" xsd:anyURI " default="http://www.
        fpml.org/coding-scheme/interpolation-method"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
```

[top](#)**Complex Type: Leg**

Super-types:

None

Sub-types:

None

**Name**

Leg

**Abstract**

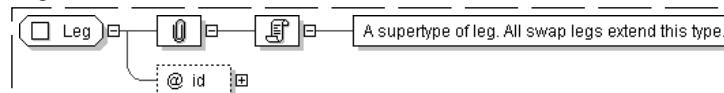
yes

**Documentation**

A supertype of leg. All swap legs extend this type.

**XML Instance Representation**

```
<...>
<id=" xsd:ID [0..1]">
```

**Diagram****Schema Component Representation**

```
<xsd:complexType name="Leg" abstract="true">
  <xsd:attribute name="id" type=" xsd:ID ">
</xsd:complexType>
```

[top](#)

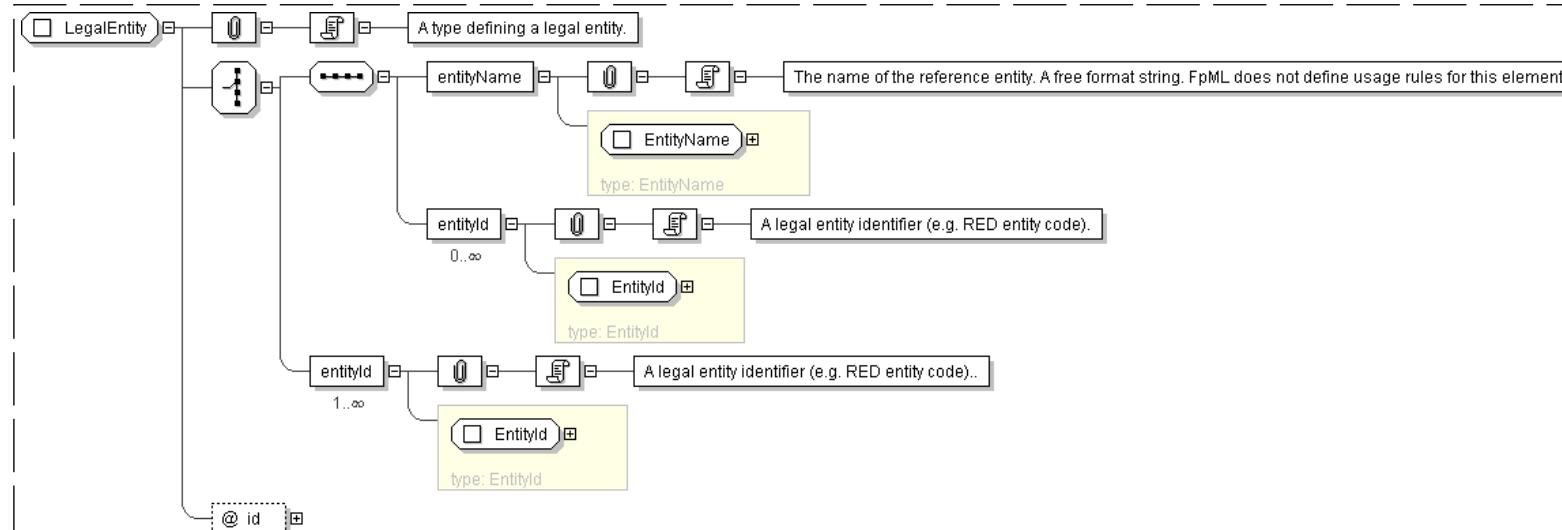
**Complex Type: LegalEntity**

**Super-types:** None  
**Sub-types:** None

<b>Name</b>	LegalEntity
<b>Abstract</b>	no
<b>Documentation</b>	A type defining a legal entity.

**XML Instance Representation**

```
<...>
  id="# xsd:ID [0..1]">
  Start Choice [1]
    <entityName> EntityName </entityName> [1]
      'The name of the reference entity. A free format string. FpML does not define usage rules
      for this element.'
    <entityId> EntityId </entityId> [0..*]
      'A legal entity identifier (e.g. RED entity code).'
    <entityId> EntityId </entityId> [1..*]
      'A legal entity identifier (e.g. RED entity code)..'
  End Choice
</...>
```

**Diagram****Schema Component Representation**

```
<xsd:complexType name="LegalEntity">
  <xsd:choice>
    <xsd:sequence>
      <xsd:element name="entityName" type=" EntityName "/>
      <xsd:element name="entityId" type=" EntityId " minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:element name="entityId" type=" EntityId " maxOccurs="unbounded"/>
  </xsd:choice>
  <xsd:attribute name="id" type=" xsd:ID "/>
</xsd:complexType>
```

## Complex Type: LegalEntityReference

Super-types:

[Reference](#) < **LegalEntityReference** (by extension)

Sub-types:

None

Name **LegalEntityReference**

Abstract

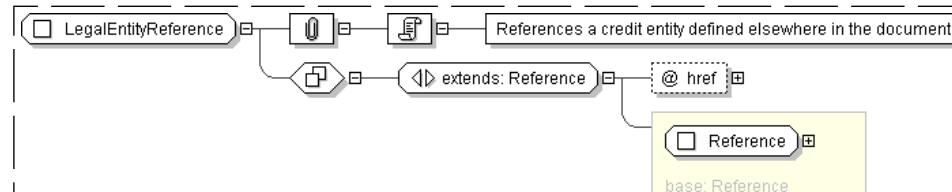
no

Documentation References a credit entity defined elsewhere in the document.

### XML Instance Representation

```
<...  
    href=" xsd:IDREF [1]" />
```

### Diagram



### Schema Component Representation

```

<xsd:complexType name="LegalEntityReference">
  <xsd:complexContent>
    <xsd:extension base=" Reference ">
      <xsd:attribute name="href" type=" xsd:IDREF " use="required" reference="LegalEntity" />
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
  
```

## Complex Type: MainPublication

Super-types:

[xsd:normalizedString](#) < [Scheme](#) (by restriction) < **MainPublication** (by extension)

Sub-types:

None

Name **MainPublication**

Abstract

no

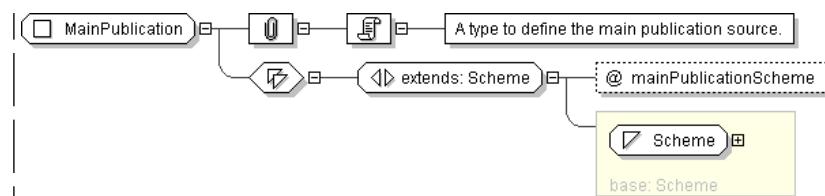
Documentation A type to define the main publication source.

### XML Instance Representation

```
<...  
    mainPublicationScheme=" xsd:anyURI [0..1]">  
    Scheme  
</...>
```

### Diagram



**Schema Component Representation**

```

<xsd:complexType name="MainPublication">
  <xsd:simpleContent>
    <xsd:extension base=" Scheme ">
      <xsd:attribute name="mainPublicationScheme" type=" xsd:anyURI " default="http://www.fpml.org/coding-scheme/inflation-main-publication"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
  
```

top

**Complex Type: ManualExercise**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None

<b>Name</b>	ManualExercise
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">ExerciseProcedure</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type defining manual exercise, i.e. that the option buyer counterparty must give notice to the option seller of exercise.

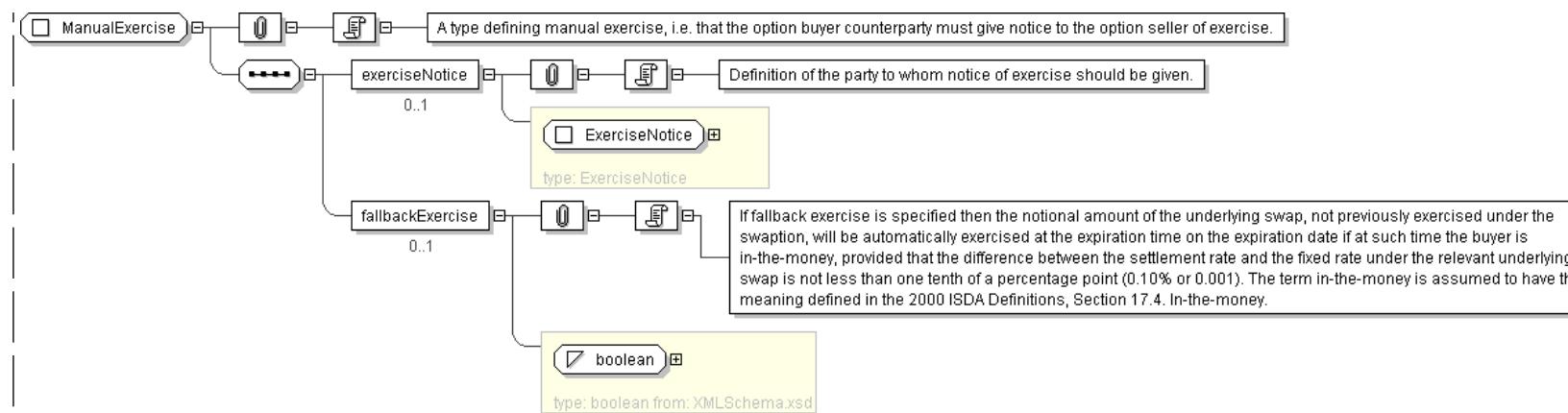
**XML Instance Representation**

```

<...>
  <exerciseNotice> ExerciseNotice </exerciseNotice> [0..1]
  'Definition of the party to whom notice of exercise should be given.'

  <fallbackExercise> xsd:boolean </fallbackExercise> [0..1]
  '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.'
</...>
  
```

**Diagram**



#### Schema Component Representation

```

<xsd:complexType name="ManualExercise">
  <xsd:sequence>
    <xsd:element name="exerciseNotice" type="#ExerciseNotice" minOccurs="0"/>
    <xsd:element name="fallbackExercise" type="xsd:boolean" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>

```

[top](#)

#### Complex Type: MasterAgreement

<b>Super-types:</b>	None
<b>Sub-types:</b>	None

<b>Name</b>	MasterAgreement
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">Documentation</a> , Complex Type <a href="#">PartyRelationshipDocumentation</a>
<b>Abstract</b>	no
<b>Documentation</b>	An entity for defining the agreement executed between the parties and intended to govern all OTC derivatives transactions between those parties.

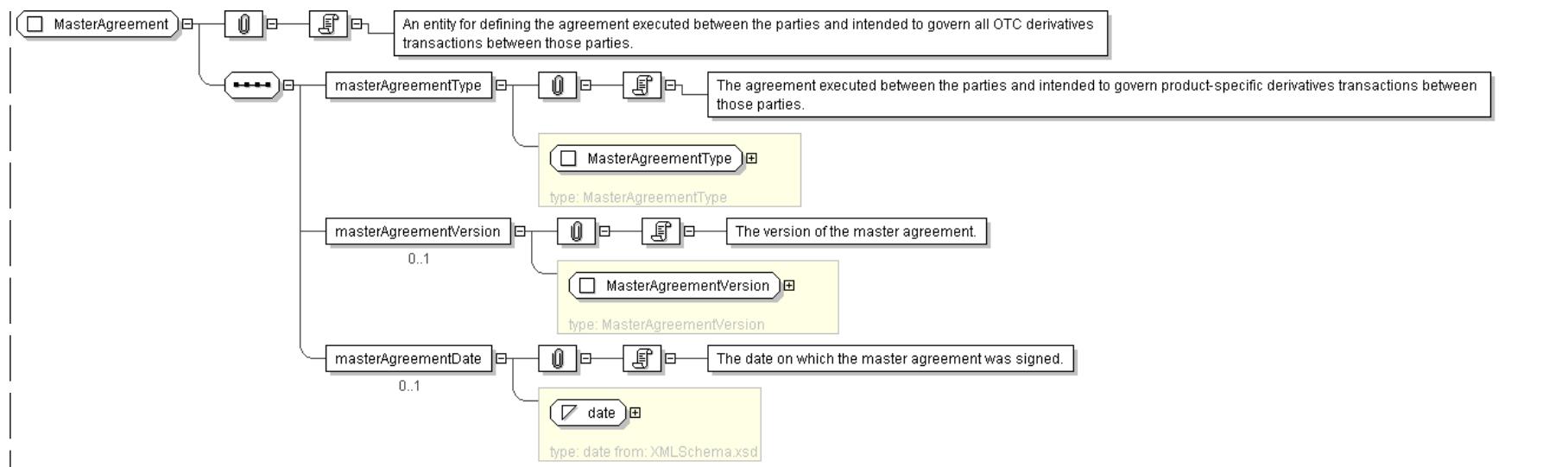
#### XML Instance Representation

```

<...>
<masterAgreementType> MasterAgreementType </masterAgreementType> [1]
  'The agreement executed between the parties and intended to govern product-specific
  derivatives transactions between those parties.'
<masterAgreementVersion> MasterAgreementVersion </masterAgreementVersion> [0..1]
  'The version of the master agreement.'
<masterAgreementDate> xsd:date </masterAgreementDate> [0..1]
  'The date on which the master agreement was signed.'
</...>

```

#### Diagram

**Schema Component Representation**

```
<xsd:complexType name="MasterAgreement">
  <xsd:sequence>
    <xsd:element name="masterAgreementType" type=" MasterAgreementType " />
    <xsd:element name="masterAgreementVersion" type=" MasterAgreementVersion " minOccurs="0" />
    <xsd:element name="masterAgreementDate" type=" xsd:date " minOccurs="0" />
  </xsd:sequence>
</xsd:complexType>
```

[top](#)**Complex Type: MasterAgreementType**

<b>Super-types:</b>	xsd:normalizedString < <a href="#">Scheme</a> (by restriction) < <b>MasterAgreementType</b> (by extension)
<b>Sub-types:</b>	None

<b>Name</b>	MasterAgreementType
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">MasterAgreement</a>
<b>Abstract</b>	no

**XML Instance Representation**

```
<...
  masterAgreementTypeScheme=" xsd:anyURI [0..1]">
  Scheme
</...>
```

**Diagram****Schema Component Representation**

```
<xsd:complexType name="MasterAgreementType">
  <xsd:simpleContent>
    <xsd:extension base=" Scheme " >
```

```

<xsd:attribute name="masterAgreementTypeScheme" type=" xsd:anyURI " default="http://www.
  fpmi.org/coding-scheme/master-agreement-type"/>
</xsd:extension>
</xsd:simpleContent>
</xsd:complexType>

```

## Complex Type: MasterAgreementVersion

Super-types:	xsd:normalizedString < Scheme (by restriction) < <b>MasterAgreementVersion</b> (by extension)
Sub-types:	None

Name	MasterAgreementVersion
Used by (from the same schema document)	Complex Type <a href="#">MasterAgreement</a>
Abstract	no

### XML Instance Representation

```

<...
  masterAgreementVersionScheme=" xsd:anyURI [0..1]">
  Scheme
</...>

```

### Diagram



### Schema Component Representation

```

<xsd:complexType name="MasterAgreementVersion">
  <xsd:simpleContent>
    <xsd:extension base=" Scheme ">
      <xsd:attribute name="masterAgreementVersionScheme" type=" xsd:anyURI " default="http://www.
        fpmi.org/coding-scheme/master-agreement-version"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>

```

## Complex Type: MasterConfirmation

Super-types:	None
Sub-types:	None

Name	MasterConfirmation
Used by (from the same schema document)	Complex Type <a href="#">Documentation</a>
Abstract	no
Documentation	An entity for defining the master confirmation agreement executed between the parties.

### XML Instance Representation

```

<...
  <masterConfirmationType> MasterConfirmationType </masterConfirmationType> [1]
  'The type of master confirmation executed between the parties.'
<masterConfirmationDate> xsd:date </masterConfirmationDate> [1]
  'The date of the confirmation executed between the parties and intended to govern all

```

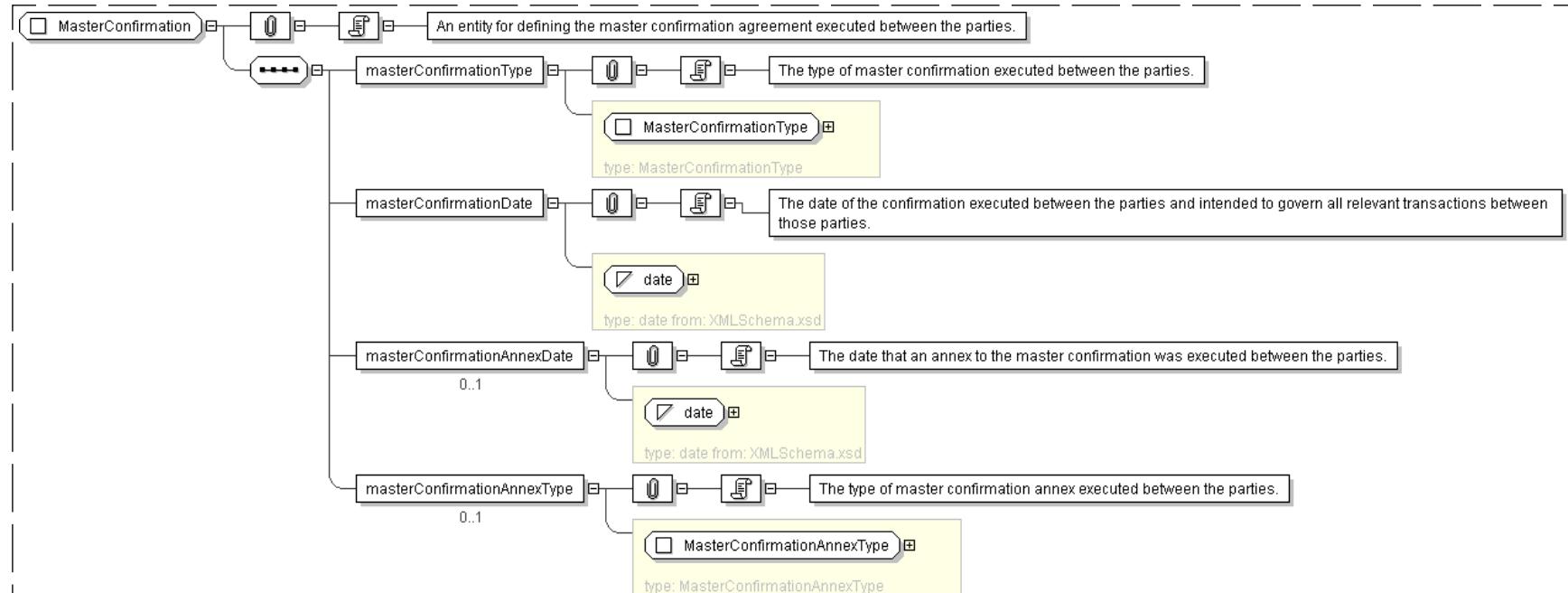
```

relevant transactions between those parties.'
```

<masterConfirmationAnnexDate> `xsd:date` </masterConfirmationAnnexDate> [0..1]  
'The date that an annex to the master confirmation was executed between the parties.'

<masterConfirmationAnnexType> `MasterConfirmationAnnexType` </masterConfirmationAnnexType> [0..1]  
'The type of master confirmation annex executed between the parties.'

</...>

**Diagram****Schema Component Representation**

```

<xsd:complexType name="MasterConfirmation">
  <xsd:sequence>
    <xsd:element name="masterConfirmationType" type=" MasterConfirmationType " />
    <xsd:element name="masterConfirmationDate" type=" xsd:date " />
    <xsd:element name="masterConfirmationAnnexDate" type=" xsd:date " minOccurs="0" />
    <xsd:element name="masterConfirmationAnnexType" type=" MasterConfirmationAnnexType "
      " minOccurs="0" />
  </xsd:sequence>
</xsd:complexType>

```

top

**Complex Type: MasterConfirmationAnnexType**

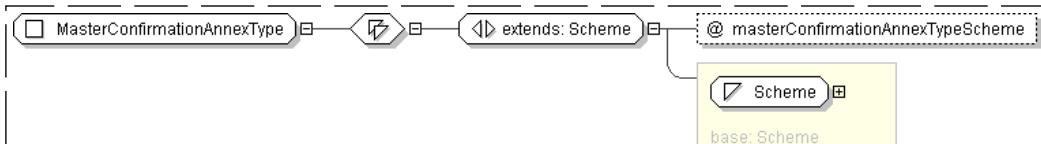
**Super-types:** `xsd:normalizedString` < [Scheme](#) (by restriction) < **MasterConfirmationAnnexType** (by extension)

**Sub-types:** None

<b>Name</b>	MasterConfirmationAnnexType
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">MasterConfirmation</a>
<b>Abstract</b>	no

**XML Instance Representation**

```
<...
  masterConfirmationAnnexTypeScheme=" xsd:anyURI [0..1]">
  Scheme
</...>
```

**Diagram****Schema Component Representation**

```
<xsd:complexType name="MasterConfirmationAnnexType">
  <xsd:simpleContent>
    <xsd:extension base=" Scheme ">
      <xsd:attribute name="masterConfirmationAnnexTypeScheme" type=" xsd:anyURI " default="http://www.fpml.org/coding-scheme/master-confirmation-annex-type"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
```

[top](#)**Complex Type: MasterConfirmationType**

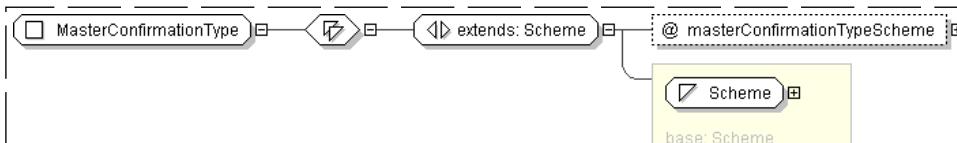
**Super-types:** xsd:normalizedString < Scheme (by restriction) < **MasterConfirmationType** (by extension)

**Sub-types:** None

<b>Name</b>	MasterConfirmationType
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">MasterConfirmation</a>
<b>Abstract</b>	no

**XML Instance Representation**

```
<...
  masterConfirmationTypeScheme=" xsd:anyURI [0..1]">
  Scheme
</...>
```

**Diagram****Schema Component Representation**

```
<xsd:complexType name="MasterConfirmationType">
  <xsd:simpleContent>
    <xsd:extension base=" Scheme ">
      <xsd:attribute name="masterConfirmationTypeScheme" type=" xsd:anyURI " default="http://www.fpml.org/coding-scheme/master-confirmation-type"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
```

[top](#)

**Complex Type: MatchId**

**Super-types:** [xsd:normalizedString](#) < [Scheme](#) (by restriction) < **MatchId** (by extension)

**Sub-types:** None

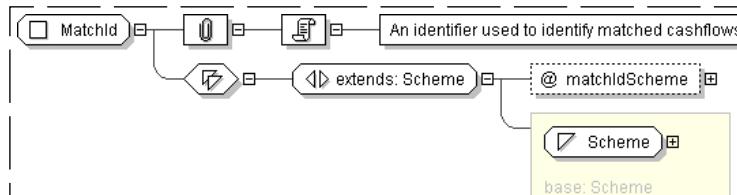
<b>Name</b>	MatchId
-------------	---------

<b>Abstract</b>	no
-----------------	----

<b>Documentation</b>	An identifier used to identify matched cashflows.
----------------------	---

**XML Instance Representation**

```
<...>
<matchIdScheme=" xsd:anyURI [0..1]">
  Scheme
</...>
```

**Diagram****Schema Component Representation**

```
<xsd:complexType name="MatchId">
  <xsd:simpleContent>
    <xsd:extension base=" Scheme ">
      <xsd:attribute name="matchIdScheme" type=" xsd:anyURI " />
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
```

[top](#)**Complex Type: Math**

**Super-types:** None

**Sub-types:** None

<b>Name</b>	Math
-------------	------

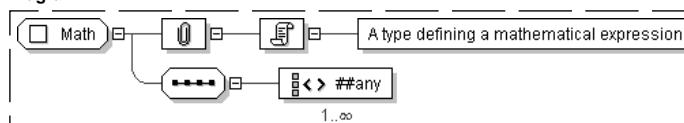
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">Formula</a>
--	--------------------------------------

<b>Abstract</b>	no
-----------------	----

<b>Documentation</b>	A type defining a mathematical expression.
----------------------	--

**XML Instance Representation**

```
<...>
<!-- Mixed content -->
  Allow any elements from any namespace (skip validation). [1..*]
</...>
```

**Diagram**

**Schema Component Representation**

```
<xsd:complexType name="Math" mixed="true">
  <xsd:sequence>
    <xsd:any namespace="##any" processContents="skip" maxOccurs="unbounded" />
  </xsd:sequence>
</xsd:complexType>
```

[top](#)**Complex Type: MatrixTerm**

**Super-types:** [xsd:normalizedString](#) < [Scheme](#) (by restriction) < **MatrixTerm** (by extension)

**Sub-types:** None

<b>Name</b>	MatrixTerm
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">ContractualMatrix</a>
<b>Abstract</b>	no

**XML Instance Representation**

```
<...
  matrixTermScheme=" xsd:anyURI [0..1]">
  Scheme
</...>
```

**Diagram****Schema Component Representation**

```
<xsd:complexType name="MatrixTerm">
  <xsd:simpleContent>
    <xsd:extension base=" Scheme ">
      <xsd:attribute name="matrixTermScheme" type=" xsd:anyURI " default="http://www.fpml.org/
        coding-scheme/credit-matrix-transaction-type"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
```

[top](#)**Complex Type: MatrixType**

**Super-types:** [xsd:normalizedString](#) < [Scheme](#) (by restriction) < **MatrixType** (by extension)

**Sub-types:** None

<b>Name</b>	MatrixType
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">ContractualMatrix</a>
<b>Abstract</b>	no

**XML Instance Representation**

```
<...
  matrixTypeScheme=" xsd:anyURI [0..1]">
  Scheme
</...>
```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="MatrixType">
  <xsd:simpleContent>
    <xsd:extension base=" Scheme ">
      <xsd:attribute name="matrixTypeScheme" type=" xsd:anyURI " default="http://www.fpml.org/
        coding-scheme/matrix-type"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
  
```

[top](#)**Complex Type:MimeType**

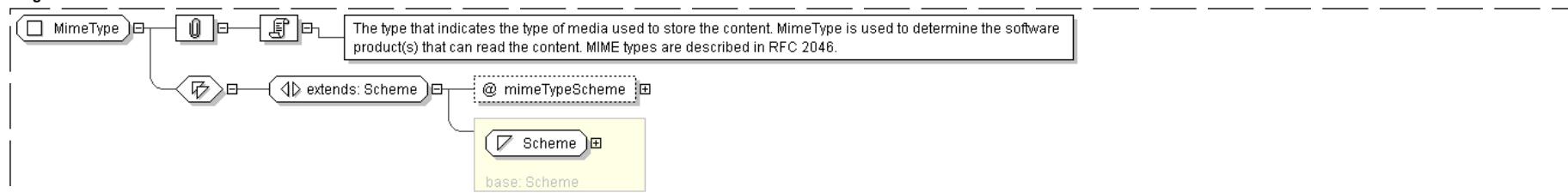
<b>Super-types:</b>	xsd:normalizedString < <a href="#">Scheme</a> (by restriction) < <b>MimeType</b> (by extension)
<b>Sub-types:</b>	None

<b>Name</b>	MimeType
<b>Abstract</b>	no
<b>Documentation</b>	The type that indicates the type of media used to store the content. MimeType is used to determine the software product(s) that can read the content. MIME types are described in RFC 2046.

**XML Instance Representation**

```

<...
  mimeTypeScheme=" xsd:anyURI [0..1]">
  Scheme
</...>
  
```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="MimeType">
  <xsd:simpleContent>
    <xsd:extension base=" Scheme ">
      <xsd:attribute name="mimeTypeScheme" type=" xsd:anyURI " />
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
  
```

[top](#)**Complex Type:Money**

**Super-types:** [MoneyBase](#) < **Money** (by extension)

**Sub-types:** None

**Name**

**Used by (from the same schema document)**

Money

Complex Type [GrossCashflow](#), Complex Type [Payment](#), Complex Type [Payment](#), Complex Type [SimplePayment](#), Complex Type [SplitSettlement](#), Complex Type [StubValue](#), Model Group [PaymentDiscounting.model](#), Model Group [Premium.model](#), Model Group [SettlementAmountOrCurrency.model](#)

**Abstract**

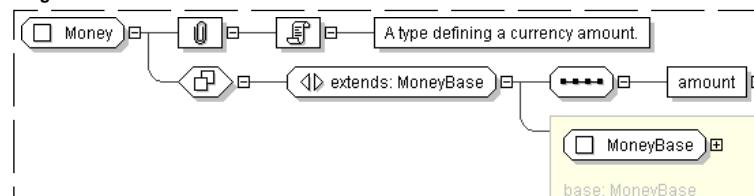
no

**Documentation**

A type defining a currency amount.

**XML Instance Representation**

```
<...>
  id="# xsd:ID [0..1]">
  <currency> Currency </currency> [1]
    'The currency in which an amount is denominated.'
  <amount> xsd:decimal </amount> [1]
    'The monetary quantity in currency units.'
</...>
```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="Money">
  <xsd:complexContent>
    <xsd:extension base=" MoneyBase ">
      <xsd:sequence>
        <xsd:element name="amount" type="xsd:decimal" />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
  
```

[top](#)

**Complex Type: MoneyBase**

**Super-types:** None

**Sub-types:**

- [Money](#) (by extension)
- [NonNegativeMoney](#) (by extension)
  - [FutureValueAmount](#) (by extension)
  - [NotionalAmount](#) (by extension)
- [PositiveMoney](#) (by extension)

**Name**

MoneyBase

**Abstract**

yes

**Documentation**

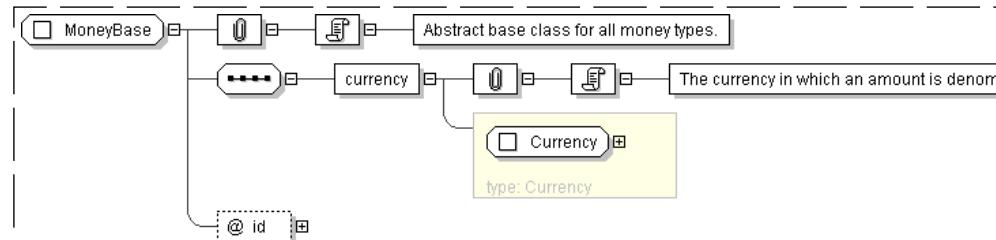
Abstract base class for all money types.

**XML Instance Representation**

```
<...>
  id="# xsd:ID [0..1]">
  <currency> Currency </currency> [1]
```

'The currency in which an amount is denominated.'

&lt;/...&gt;

**Diagram****Schema Component Representation**

```

<xsd:complexType name="MoneyBase" abstract="true">
  <xsd:sequence>
    <xsd:element name="currency" type="Currency" />
  </xsd:sequence>
  <xsd:attribute name="id" type="xsd:ID" />
</xsd:complexType>
  
```

top

**Complex Type: MultipleExercise**

Super-types:

None

Sub-types:

None

<b>Name</b>	MultipleExercise
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">AmericanExercise</a> , Complex Type <a href="#">BermudaExercise</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type defining multiple exercises. As defining 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 exercised must be equal to or, be an integral multiple of, the integral multiple amount.

**XML Instance Representation**

&lt;...&gt;

```

<notionalReference> NotionalReference </notionalReference> [0..*]
  
```

'A pointer style reference to the associated notional schedule defined elsewhere in the document. This element has been made optional as part of its integration in the OptionBaseExtended, because not required for the options on securities.'

```

<integralMultipleAmount> xsd:decimal </integralMultipleAmount> [0..1]
  
```

'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.'

Start Choice [1]

```

<minimumNotionalAmount> xsd:decimal </minimumNotionalAmount> [1]
  
```

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

```

<minimumNumberOfOptions> xsd:nonNegativeInteger </minimumNumberOfOptions> [1]
  
```

'The minimum number of options that can be exercised on a given exercise date.'

End Choice

Start Choice [0..1]

```
<maximumNotionalAmount> xsd:decimal </maximumNotionalAmount> [1]
```

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

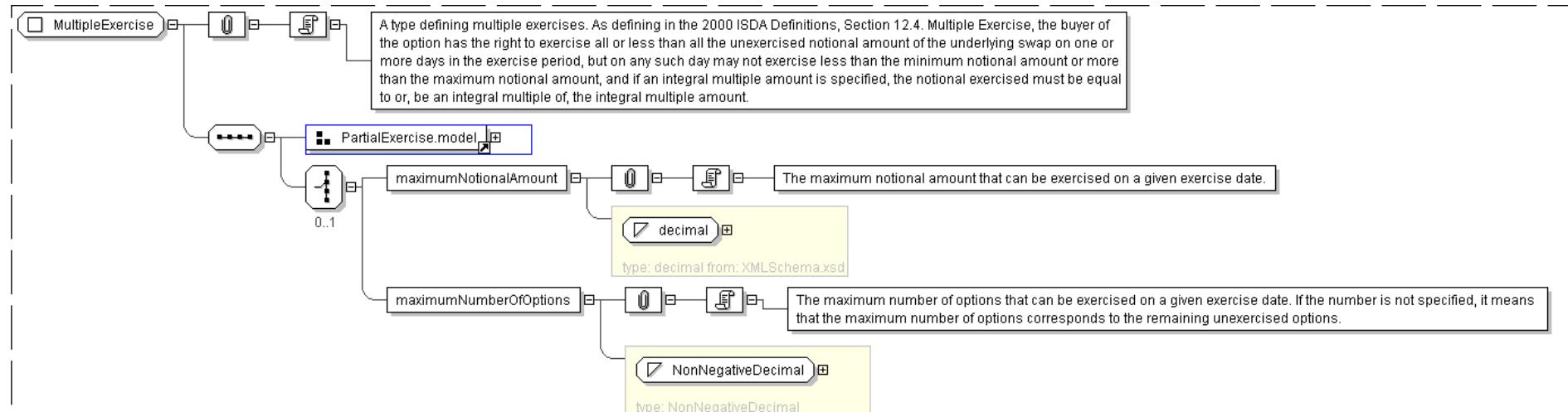
```
<maximumNumberOfOptions> NonNegativeDecimal </maximumNumberOfOptions> [1]
```

'The maximum number of options that can be exercised on a given exercise date. If the number is not specified, it means that the maximum number of options corresponds to the remaining unexercised options.'

End Choice

</...>

#### Diagram



#### Schema Component Representation

```
<xsd:complexType name="MultipleExercise">
  <xsd:sequence>
    <xsd:group ref=" PartialExercise.model " />
    <xsd:choice minOccurs="0">
      <xsd:element name="maximumNotionalAmount" type=" xsd:decimal " />
      <xsd:element name="maximumNumberOfOptions" type=" NonNegativeDecimal " />
    </xsd:choice>
  </xsd:sequence>
</xsd:complexType>
```

top

## Complex Type: NonNegativeAmountSchedule

Super-types:

[NonNegativeSchedule](#) < **NonNegativeAmountSchedule** (by extension)

Sub-types:

None

Name	NonNegativeAmountSchedule
Abstract	no
Documentation	A type defining a currency amount or a currency amount schedule.

#### XML Instance Representation

```
<...
  id=" xsd:ID [0..1]">
  <initialValue> NonNegativeDecimal </initialValue> [1]
```

'The non-negative initial rate or amount, as the case may be. An initial rate of 5% would be represented as 0.05.'

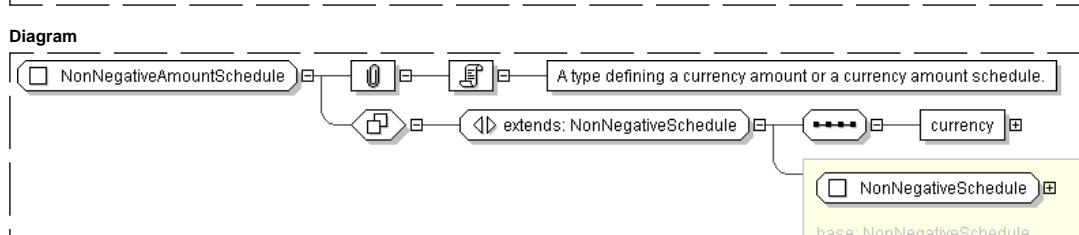
```
<step> NonNegativeStep </step> [0..*
```

'The schedule of step date and non-negative 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.'

```
<currency> Currency </currency> [ ]
```

'The currency in which an amount is denominated.

| </...>



## Schema Component Representation

```
<xsd:complexType name="NonNegativeAmountSchedule">
  <xsd:complexContent>
    <xsd:extension base="NonNegativeSchedule">
      <xsd:sequence>
        <xsd:element name="currency" type="Currency" />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

top

**Complex Type:** NonNegativeMoney

<b>Super-types:</b>	<a href="#">MoneyBase</a> < <a href="#">NonNegativeMoney</a> (by extension)
<b>Sub-types:</b>	<ul style="list-style-type: none"><li>• <a href="#">FutureValueAmount</a> (by extension)</li><li>• <a href="#">NotionalAmount</a> (by extension)</li></ul>

Name	NonNegativeMoney
Used by (from the same schema document)	Complex Type <a href="#">NonNegativePayment</a>
Abstract	no
Documentation	A type defining a non negative money amount.

## XML Instance Representation

```
| <...  
| id=" xsd:ID [ 0..1]">
```

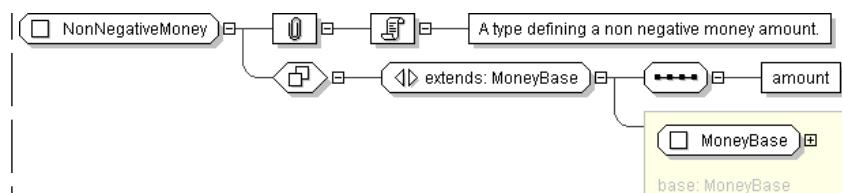
**<currency>** Currency **</currency>** [1]  
*'The currency in which an amount is denominated.'*

<amount> NonNegativeDecimal </amount> [1]

'The non negative monetary quantity in currency units.

| </...>

## Diagram

**Schema Component Representation**

```

<xsd:complexType name="NonNegativeMoney">
  <xsd:complexContent>
    <xsd:extension base=" MoneyBase ">
      <xsd:sequence>
        <xsd:element name="amount" type=" NonNegativeDecimal " />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
  
```

[top](#)**Complex Type: NonNegativePayment**

<b>Super-types:</b>	PaymentBase < PaymentBaseExtended (by extension) < <b>NonNegativePayment</b> (by extension)
<b>Sub-types:</b>	None

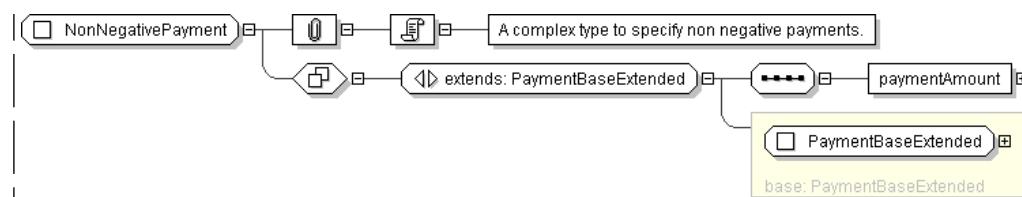
<b>Name</b>	NonNegativePayment
<b>Abstract</b>	no
<b>Documentation</b>	A complex type to specify non negative payments.

**XML Instance Representation**

```

<...>
<id=" xsd:ID [0..1]">
  <payerPartyReference> PartyReference </payerPartyReference> [1]
  'A reference to the party responsible for making the payments defined by this structure.'
  <payerAccountReference> AccountReference </payerAccountReference> [0..1]
  'A reference to the account responsible for making the payments defined by this structure.'
  <receiverPartyReference> PartyReference </receiverPartyReference> [1]
  'A reference to the party that receives the payments corresponding to this structure.'
  <receiverAccountReference> AccountReference </receiverAccountReference> [0..1]
  'A reference to the account that receives the payments corresponding to this structure.'
  <paymentDate> AdjustableOrRelativeDate </paymentDate> [1]
  'The payment date, which can be expressed as either an adjustable or relative date.'
  <paymentAmount> NonNegativeMoney </paymentAmount> [1]
  'Non negative payment amount.'
</...>
  
```

**Diagram**

**Schema Component Representation**

```

<xsd:complexType name="NonNegativePayment">
  <xsd:complexContent>
    <xsd:extension base=" PaymentBaseExtended ">
      <xsd:sequence>
        <xsd:element name="paymentAmount" type=" NonNegativeMoney " />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
  
```

[top](#)**Complex Type: NonNegativeSchedule**

<b>Super-types:</b>	None
<b>Sub-types:</b>	<ul style="list-style-type: none"> <li>• <a href="#">NonNegativeAmountSchedule</a> (by extension)</li> </ul>

<b>Name</b>	NonNegativeSchedule
<b>Abstract</b>	no
<b>Documentation</b>	A type defining a schedule of non-negative rates 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.

**XML Instance Representation**

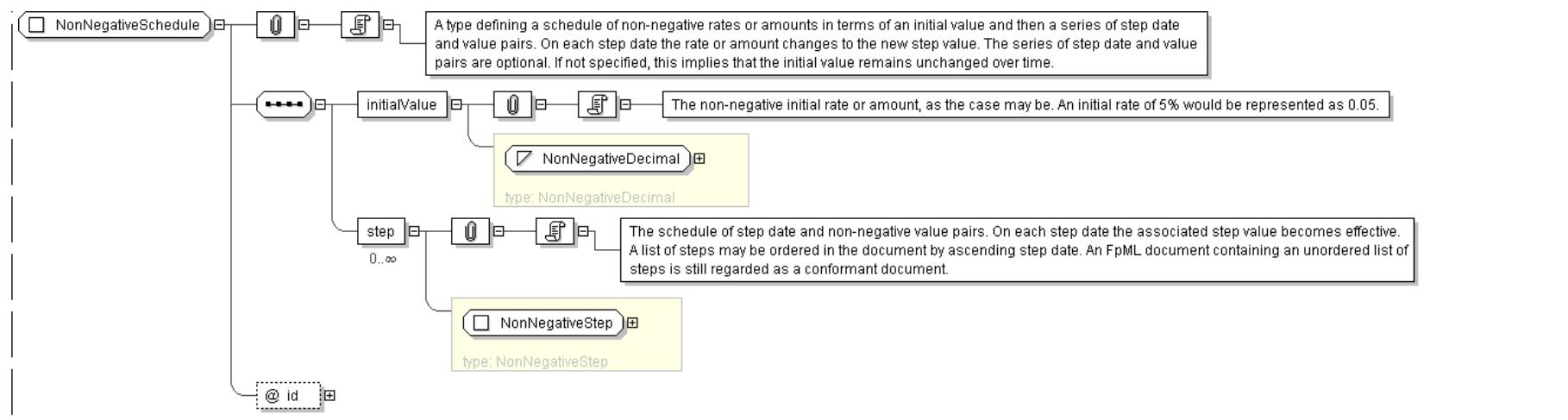
```

<...
  id=" xsd:ID [0..1]">
  <initialValue> NonNegativeDecimal </initialValue> [1]
  'The non-negative initial rate or amount, as the case may be. An initial rate of 5% would
  be represented as 0.05.'

  <step> NonNegativeStep </step> [0..*]
  'The schedule of step date and non-negative 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.'

</...>
  
```

**Diagram**



#### Schema Component Representation

```

<xsd:complexType name="NonNegativeSchedule">
  <xsd:sequence>
    <xsd:element name="initialValue" type="NonNegativeDecimal" />
    <xsd:element name="step" type="NonNegativeStep" minOccurs="0" maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:attribute name="id" type="xsd:ID" />
</xsd:complexType>

```

top

#### Complex Type: NonNegativeStep

Super-types:	<a href="#">StepBase</a> < <code>NonNegativeStep</code> (by extension)
Sub-types:	None

Name	NonNegativeStep
Used by (from the same schema document)	Complex Type <a href="#">NonNegativeSchedule</a>
Abstract	no
Documentation	A type defining a step date and non-negative step value pair. This step definitions are used to define varying rate or amount schedules, e.g. a notional amortization or a step-up coupon schedule.

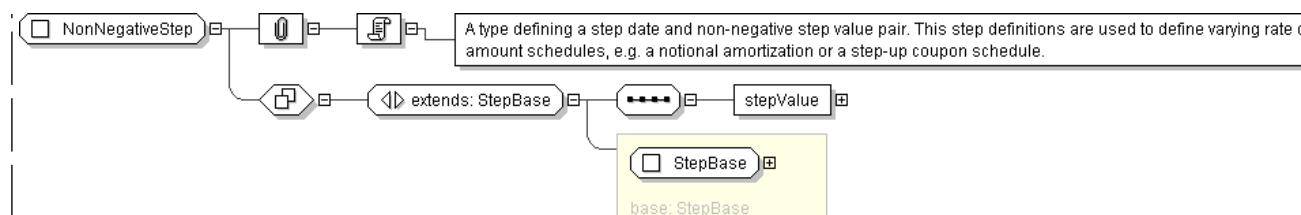
#### XML Instance Representation

```

<...
  id="xsd:ID [0..1]">
  <stepDate> xsd:date </stepDate> [1]
    'The date on which the associated stepValue becomes effective. This day may be subject
    to adjustment in accordance with a business day convention.'
  <stepValue> NonNegativeDecimal </stepValue> [1]
    'The non-negative rate or amount which becomes effective on the associated stepDate. A rate
    of 5% would be represented as 0.05.'
</...>

```

#### Diagram

**Schema Component Representation**

```

<xsd:complexType name="NonNegativeStep">
  <xsd:complexContent>
    <xsd:extension base=" StepBase ">
      <xsd:sequence>
        <xsd:element name="stepValue" type=" NonNegativeDecimal " />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
  
```

top

**Complex Type: NotionalAmount**

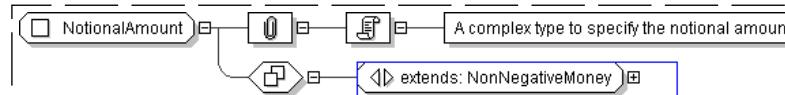
<b>Super-types:</b>	<a href="#">MoneyBase</a> < <a href="#">NonNegativeMoney</a> (by extension) < <b>NotionalAmount</b> (by extension)
<b>Sub-types:</b>	None

<b>Name</b>	NotionalAmount
<b>Abstract</b>	no
<b>Documentation</b>	A complex type to specify the notional amount.

**XML Instance Representation**

```

<...
  id=" xsd:ID [0..1]">
  <currency> Currency </currency> [1]
  'The currency in which an amount is denominated.'
  <amount> NonNegativeDecimal </amount> [1]
  'The non negative monetary quantity in currency units.'
<...>
  
```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="NotionalAmount">
  <xsd:complexContent>
    <xsd:extension base=" NonNegativeMoney ">
  </xsd:extension>
</xsd:complexContent>
</xsd:complexType>
  
```

top

**Complex Type: NotionalAmountReference**

**Super-types:** [Reference](#) < **NotionalAmountReference** (by extension)

**Sub-types:** None

**Name** NotionalAmountReference

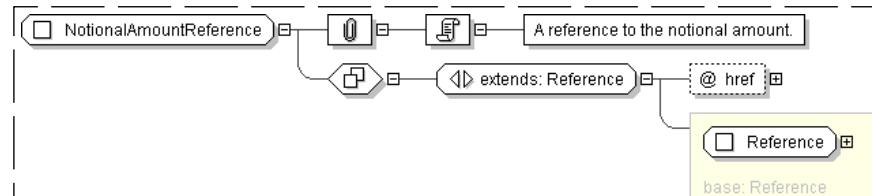
**Abstract** no

**Documentation** A reference to the notional amount.

#### XML Instance Representation

```
<...  
  href="# IDREF [1]" />
```

#### Diagram



#### Schema Component Representation

```

<xsd:complexType name="NotionalAmountReference">
  <xsd:complexContent>
    <xsd:extension base=" Reference ">
      <xsd:attribute name="href" type=" xsd:IDREF " use="required" />
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

[top](#)

## Complex Type: NotionalReference

**Super-types:** [Reference](#) < **NotionalReference** (by extension)

**Sub-types:** None

**Name** NotionalReference

**Used by (from the same schema document)** Complex Type [ExerciseFee](#), Model Group [PartialExercise.model](#)

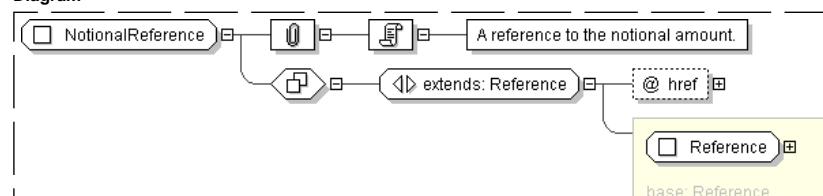
**Abstract** no

**Documentation** A reference to the notional amount.

#### XML Instance Representation

```
<...  
  href="# IDREF [1]" />
```

#### Diagram



#### Schema Component Representation

```

<xsd:complexType name="NotionalReference">
  <xsd:complexContent>

```

```

<xsd:extension base=" Reference ">
  <xsd:attribute name="href" type=" xsd:IDREF" use="required" reference="Notional"/>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>

```

## Complex Type: [Offset](#)

**Super-types:** [Period](#) < **Offset** (by extension)

**Sub-types:**

- [DateOffset](#) (by extension)
- [RelativeDateOffset](#) (by extension)
  - [AdjustedRelativeDateOffset](#) (by extension)
  - [RelativeDates](#) (by extension)

<b>Name</b>	Offset
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">OffsetPrevailingTime</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type 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.

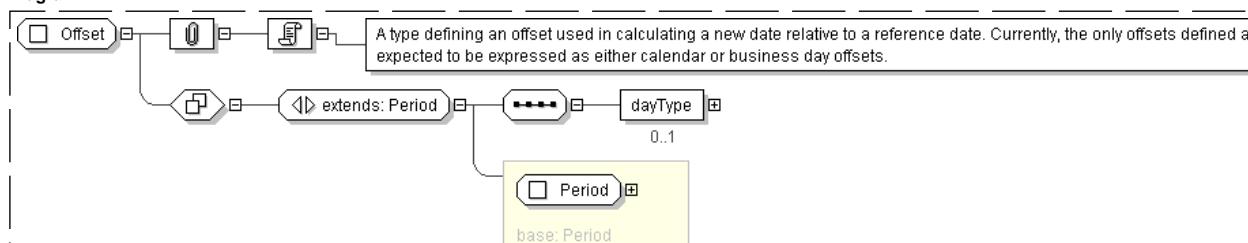
### XML Instance Representation

```

<...
  id=" xsd:ID [0..1]">
  <periodMultiplier> xsd:integer </periodMultiplier> [1]
  '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.'
  <period> PeriodEnum </period> [1]
  'A time period, e.g. a day, week, month or year of the stream. If the periodMultiplier value
  is 0 (zero) then period must contain the value D (day).'
  <dayType> DayTypeEnum </dayType> [0..1]
  '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.'
</...>

```

### Diagram



### Schema Component Representation

```

<xsd:complexType name="Offset">
  <xsd:complexContent>
    <xsd:extension base=" Period ">
      <xsd:sequence>
        <xsd:element name="dayType" type=" DayTypeEnum" minOccurs="0"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

```
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>
```

## Complex Type: OffsetPrevailingTime

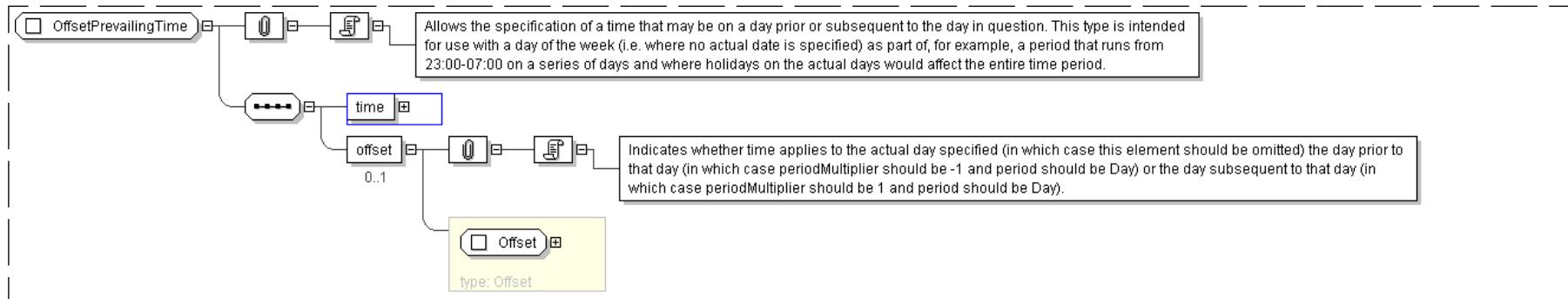
Super-types:	None
Sub-types:	None

Name	OffsetPrevailingTime
Abstract	no
Documentation	Allows the specification of a time that may be on a day prior or subsequent to the day in question. This type is intended for use with a day of the week (i.e. where no actual date is specified) as part of, for example, a period that runs from 23:00-07:00 on a series of days and where holidays on the actual days would affect the entire time period.

### XML Instance Representation

```
<...>
  <time> PrevailingTime </time> [1]
  <offset> Offset </offset> [0..1]
  'Indicates whether time applies to the actual day specified (in which case this element
  should be omitted) the day prior to that day (in which case periodMultiplier should be -1
  and period should be Day) or the day subsequent to that day (in which case
  periodMultiplier should be 1 and period should be Day).'
</...>
```

### Diagram



### Schema Component Representation

```
<xsd:complexType name="OffsetPrevailingTime">
  <xsd:sequence>
    <xsd:element name="time" type="#_PrevailingTime" />
    <xsd:element name="offset" type="Offset" minOccurs="0" />
  </xsd:sequence>
</xsd:complexType>
```

## Complex Type: PartialExercise

Super-types:	None
Sub-types:	None

Name	PartialExercise
Used by (from the same schema document)	Complex Type <a href="#">EuropeanExercise</a>

<b>Abstract</b>	no
<b>Documentation</b>	A type defining 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.

**XML Instance Representation**

```

<...>
<notionalReference> NotionalReference </notionalReference> [0..*]
'A pointer style reference to the associated notional schedule defined elsewhere in
the document. This element has been made optional as part of its integration in
the OptionBaseExtended, because not required for the options on securities.'

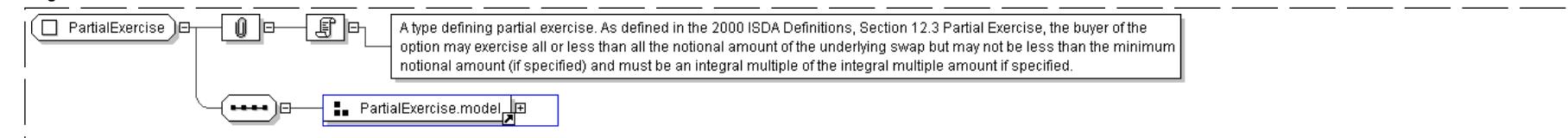
<integralMultipleAmount> xsd:decimal </integralMultipleAmount> [0..1]
'A notional amount which restricts the amount of notional that can be exercised when
partial exercise or multiple exercise is applicable. The integral multiple amount defines
a lower limit of notional that can be exercised and also defines a unit multiple of
notional that can be exercised, i.e. only integer multiples of this amount can be exercised.'

Start Choice [1]
<minimumNotionalAmount> xsd:decimal </minimumNotionalAmount> [1]
'The minimum notional amount that can be exercised on a given exercise date.
See multipleExercise.'

<minimumNumberOfOptions> xsd:nonNegativeInteger </minimumNumberOfOptions> [1]
'The minimum number of options that can be exercised on a given exercise date.'

End Choice
</...>

```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="PartialExercise">
  <xsd:sequence>
    <xsd:group ref="PartialExercise.model" />
  </xsd:sequence>
</xsd:complexType>

```

top

**Complex Type: Party**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None

<b>Name</b>	Party
<b>Used by (from the same schema document)</b>	Model Group <a href="#">PartiesAndAccounts.model</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type defining a legal entity or a subdivision of a legal entity.  Parties can perform multiple roles in a trade lifecycle. For example, the principal parties obligated to make payments from time to time during the term of the trade, but may include other parties involved in, or incidental to, the trade, such as parties acting in the role of novation transferor/transferee, broker, calculation agent, etc. In FpML roles are defined in multiple places within a document.

**XML Instance Representation**

```

<...>
| 

```

```

  id=" xsd:ID [1]
  'The id uniquely identifying the Party within the document.'

  ">

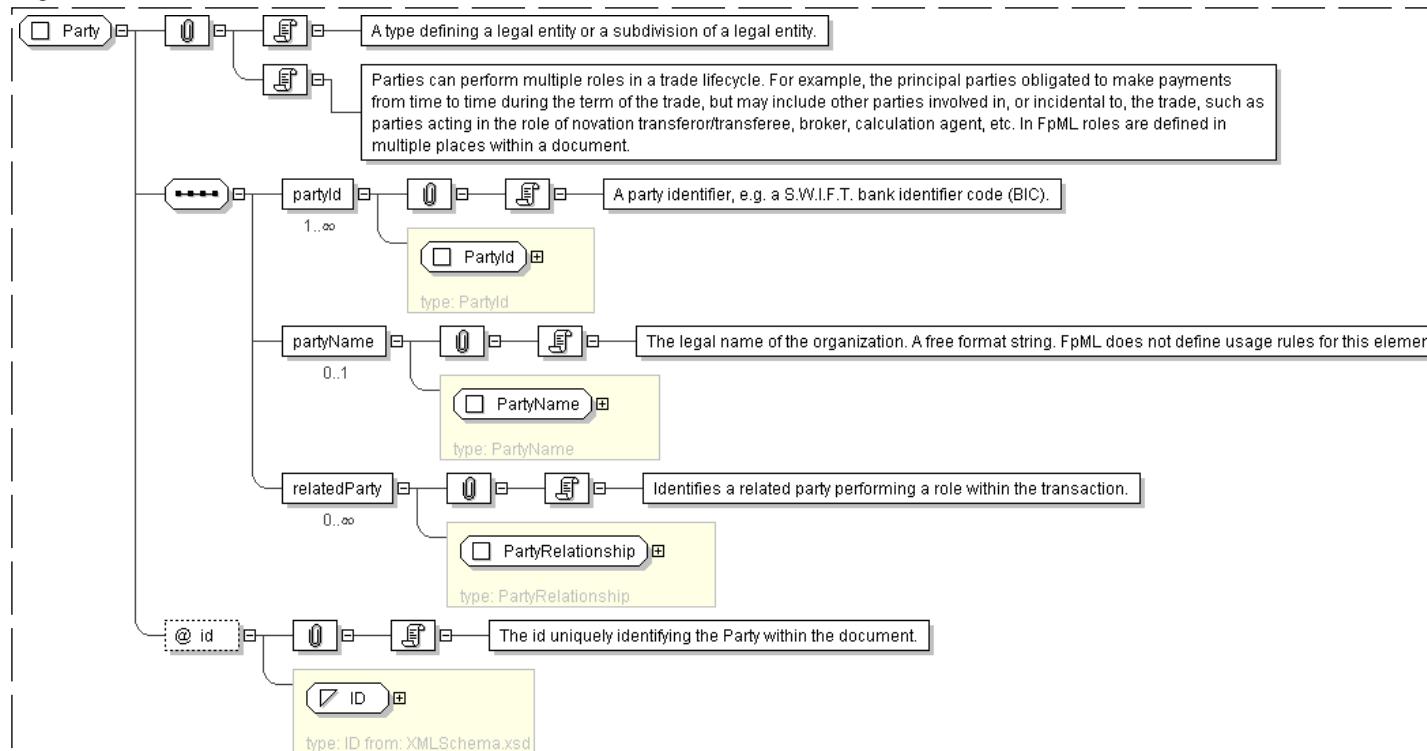
<partyId> PartyId </partyId> [1..*]
  'A party identifier, e.g. a S.W.I.F.T. bank identifier code (BIC).'

<partyName> PartyName </partyName> [0..1]
  'The legal name of the organization. A free format string. FpML does not define usage rules
  for this element.'

<relatedParty> PartyRelationship </relatedParty> [0..*]
  'Identifies a related party performing a role within the transaction.'

</...>

```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="Party">
  <xsd:sequence>
    <xsd:element name="partyId" type=" PartyId " maxOccurs="unbounded" />
    <xsd:element name="partyName" type=" PartyName " minOccurs="0" />
    <xsd:element name="relatedParty" type=" PartyRelationship " minOccurs="0"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:attribute name="id" type=" xsd:ID " use="required" />
</xsd:complexType>

```

**Complex Type: PartyId**

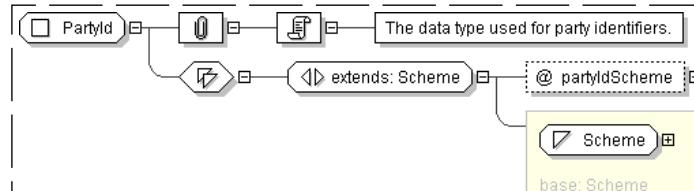
**Super-types:** [xsd:normalizedString](#) < [Scheme](#) (by restriction) < **PartyId** (by extension)

None

<b>Name</b>	PartyId
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">Party</a>
<b>Abstract</b>	no
<b>Documentation</b>	The data type used for party identifiers.

**XML Instance Representation**

```
<...
  partyIdScheme=" xsd:anyURI [0..1]">
  Scheme
</...>
```

**Diagram****Schema Component Representation**

```
<xsd:complexType name="PartyId">
  <xsd:simpleContent>
    <xsd:extension base=" Scheme ">
      <xsd:attribute name="partyIdScheme" type=" xsd:anyURI " default="http://www.fpml.org/
        ext/iso9362"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
```

[top](#)**Complex Type: PartyName**

**Super-types:** [xsd:normalizedString](#) < [Scheme](#) (by restriction) < **PartyName** (by extension)

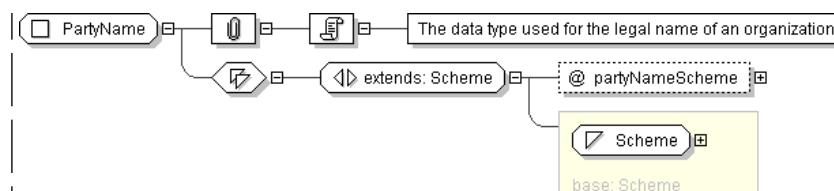
None

<b>Name</b>	PartyName
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">Party</a>
<b>Abstract</b>	no
<b>Documentation</b>	The data type used for the legal name of an organization.

**XML Instance Representation**

```
<...
  partyNameScheme=" xsd:anyURI [0..1]">
  Scheme
</...>
```

**Diagram**

**Schema Component Representation**

```

<xsd:complexType name="PartyName">
  <xsd:simpleContent>
    <xsd:extension base=" Scheme ">
      <xsd:attribute name="partyNameScheme" type=" xsd:anyURI " use="optional" />
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
  
```

top

**Complex Type: PartyReference**

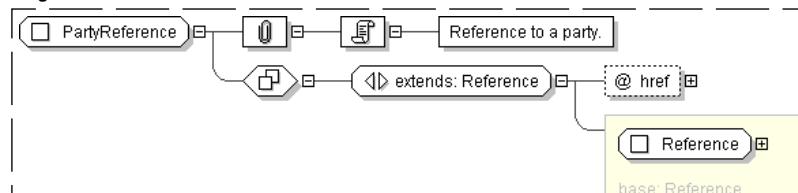
<b>Super-types:</b>	<a href="#">Reference</a> < <b>PartyReference</b> (by extension)
<b>Sub-types:</b>	None

<b>Name</b>	PartyReference
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">Account</a> , Complex Type <a href="#">Account</a> , Complex Type <a href="#">Account</a> , Complex Type <a href="#">Beneficiary</a> , Complex Type <a href="#">CalculationAgent</a> , Complex Type <a href="#">CorrespondentInformation</a> , Complex Type <a href="#">ExerciseNotice</a> , Complex Type <a href="#">ExerciseNotice</a> , Complex Type <a href="#">IntermediaryInformation</a> , Complex Type <a href="#">SettlementInstruction</a> , Model Group <a href="#">BuyerSeller.model</a> , Model Group <a href="#">BuyerSeller.model</a> , Model Group <a href="#">PartyAndAccountReferences.model</a> , Model Group <a href="#">PayerReceiver.model</a> , Model Group <a href="#">PayerReceiver.model</a>
<b>Abstract</b>	no
<b>Documentation</b>	Reference to a party.

**XML Instance Representation**

```

<...>
  href=" xsd:IDREF [1]" />
  
```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="PartyReference">
  <xsd:complexContent>
    <xsd:extension base=" Reference ">
      <xsd:attribute name="href" type=" xsd:IDREF " use="required" reference="Party" />
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
  
```

top

**Complex Type: PartyRelationship**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None

<b>Name</b>	PartyRelationship
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">Party</a>
<b>Abstract</b>	no

**XML Instance Representation**

```

<...>
  <partyReference> PartyReference </partyReference> [1]
    'Reference to a party.'

  <accountReference> AccountReference </accountReference> [0..1]
    'Reference to an account.'

  <role> PartyRole </role> [1]
    'The category of the relationship. The related party performs the role specified in this
    field for the base party. For example, if the role is \"Guarantor\", the related party acts
    as a guarantor for the base party.'

  <type> PartyRoleType </type> [0..1]
    'Additional definition refining the type of relationship. For example, if the \"role\"'
    'is Guarantor, this element may be used to specify whether all positions are guaranteed, or
    only a subset of them.'

  <effectiveDate> xsd:date </effectiveDate> [0..1]
    'The date on which the relationship begins or began.'

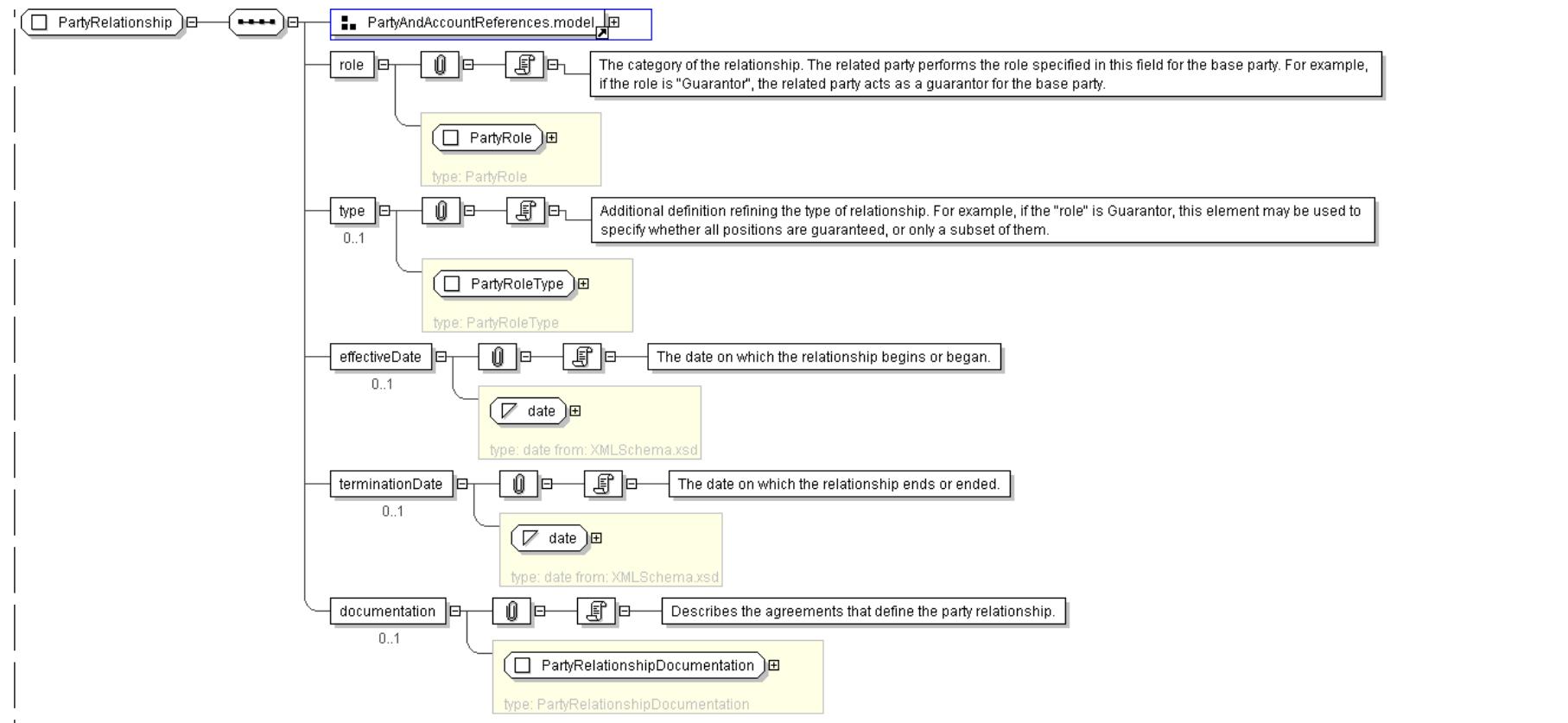
  <terminationDate> xsd:date </terminationDate> [0..1]
    'The date on which the relationship ends or ended.'

  <documentation> PartyRelationshipDocumentation </documentation> [0..1]
    'Describes the agreements that define the party relationship.'

</...>

```

**Diagram**



## Schema Component Representation

```
<xsd:complexType name="PartyRelationship">
  <xsd:sequence>
    <xsd:group ref="PartyAndAccountReferences.model" />
    <xsd:element name="role" type="PartyRole" />
    <xsd:element name="type" type="PartyRoleType" minOccurs="0" />
    <xsd:element name="effectiveDate" type="xsd:date" minOccurs="0" />
    <xsd:element name="terminationDate" type="xsd:date" minOccurs="0" />
    <xsd:element name="documentation" type="PartyRelationshipDocumentation" minOccurs="0" />
  </xsd:sequence>
</xsd:complexType>
```

top

## Complex Type: PartyRelationshipDocumentation

<i>Super-types:</i>	None
<i>Sub-types:</i>	None
<b>Name</b>	PartyRelationshipDocumentation
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">PartyRelationship</a>
<b>Abstract</b>	no
<b>Documentation</b>	A description of the legal agreement(s) and definitions that document a party's relationships with other parties

## XML Instance Representation

1

```

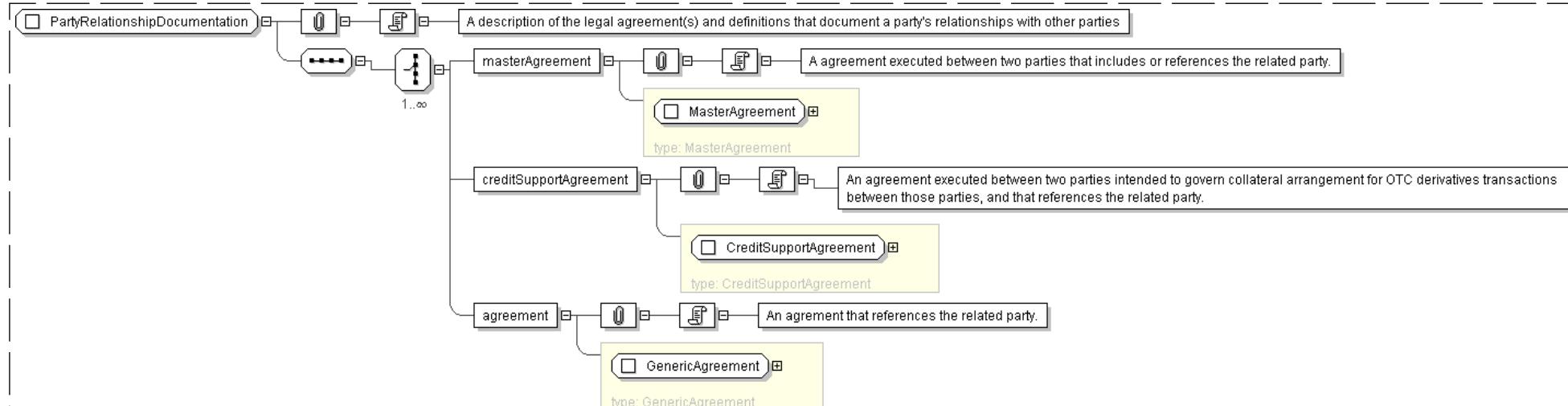
| Start Choice [1..*]
<masterAgreement> MasterAgreement </masterAgreement> [1]
'A agreement executed between two parties that includes or references the related party.'

<creditSupportAgreement> CreditSupportAgreement </creditSupportAgreement> [1]
'An agreement executed between two parties intended to govern collateral arrangement for OTC derivatives transactions between those parties, and that references the related party.'

<agreement> GenericAgreement </agreement> [1]
'An agreement that references the related party.'

End Choice
</...>

```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="PartyRelationshipDocumentation">
  <xsd:sequence>
    <xsd:choice minOccurs="1" maxOccurs="unbounded">
      <xsd:element name="masterAgreement" type="MasterAgreement" />
      <xsd:element name="creditSupportAgreement" type="CreditSupportAgreement" />
      <xsd:element name="agreement" type="GenericAgreement" />
    </xsd:choice>
  </xsd:sequence>
</xsd:complexType>

```

top

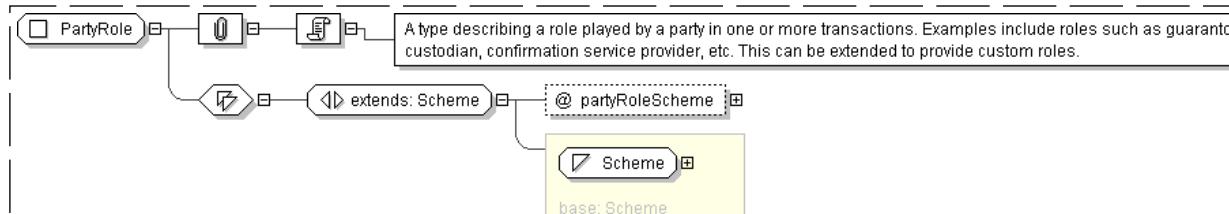
**Complex Type: PartyRole**

Super-types:	<u>xsd:normalizedString</u> < <u>Scheme</u> (by restriction) < <b>PartyRole</b> (by extension)
Sub-types:	None

Name	PartyRole
Used by (from the same schema document)	Complex Type <u>PartyRelationship</u> , Complex Type <u>RelatedParty</u>
Abstract	no
Documentation	A type describing a role played by a party in one or more transactions. Examples include roles such as guarantor, custodian, confirmation service provider, etc. This can be extended to provide custom roles.

**XML Instance Representation**

```
<...>
<@ partyRoleScheme=" xsd:anyURI [0..1]">
  Scheme
</...>
```

**Diagram****Schema Component Representation**

```
<xsd:complexType name="PartyRole">
  <xsd:simpleContent>
    <xsd:extension base=" Scheme ">
      <xsd:attribute name="partyRoleScheme" type=" xsd:anyURI " default="http://www.fpml.org/coding-scheme/party-role"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
```

top

**Complex Type: PartyRoleType**

**Super-types:** xsd:normalizedString < [Scheme](#) (by restriction) < **PartyRoleType** (by extension)

**Sub-types:** None

**Name** PartyRoleType

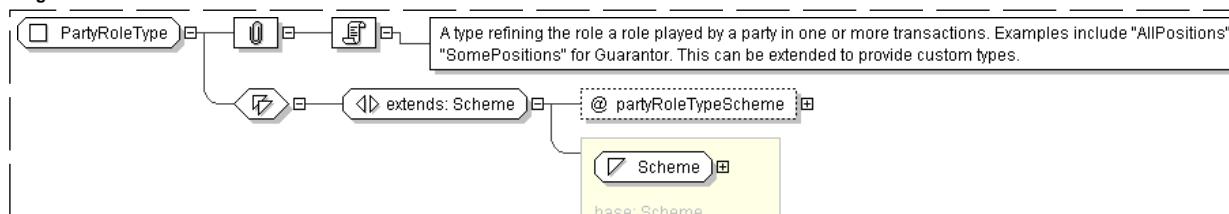
**Used by (from the same schema document)** Complex Type [PartyRelationship](#), Complex Type [RelatedParty](#)

**Abstract** no

**Documentation** A type refining the role a role played by a party in one or more transactions. Examples include "AllPositions" and "SomePositions" for Guarantor. This can be extended to provide custom types.

**XML Instance Representation**

```
<...>
<@ partyRoleTypeScheme=" xsd:anyURI [0..1]">
  Scheme
</...>
```

**Diagram****Schema Component Representation**

```
<xsd:complexType name="PartyRoleType">
  <xsd:simpleContent>
    <xsd:extension base=" Scheme ">
```

```

<xsd:attribute name="partyRoleTypeScheme" type=" xsd:anyURI " default="http://www.fpml.org/coding-scheme/party-role-type"/>
</xsd:extension>
</xsd:simpleContent>
</xsd:complexType>

```

## Complex Type: Payment

Super-types: [PaymentBase](#) < **Payment** (by extension)

Sub-types: None

Name	Payment
Abstract	no
Documentation	A type for defining payments

### XML Instance Representation

```

<...>
  id=" xsd:ID [0..1]"
  href=" xsd:IDREF [0..1]
  'Can be used to reference the yield curve used to estimate the discount factor'

  ">
    <payerPartyReference> PartyReference </payerPartyReference> [1]
    'A reference to the party responsible for making the payments defined by this structure.'

    <payerAccountReference> AccountReference </payerAccountReference> [0..1]
    'A reference to the account responsible for making the payments defined by this structure.'

    <receiverPartyReference> PartyReference </receiverPartyReference> [1]
    'A reference to the party that receives the payments corresponding to this structure.'

    <receiverAccountReference> AccountReference </receiverAccountReference> [0..1]
    'A reference to the account that receives the payments corresponding to this structure.'

    <paymentAmount> Money </paymentAmount> [1]
    'The currency amount of the payment.'

    <paymentDate> AdjustableOrAdjustedDate </paymentDate> [0..1]
    'The payment date. This date is subject to adjustment in accordance with any applicable business day convention.'

    <paymentType> PaymentType </paymentType> [0..1]
    '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.'

    <settlementInformation> SettlementInformation </settlementInformation> [0..1]
    'The information required to settle a currency payment that results from a trade.'

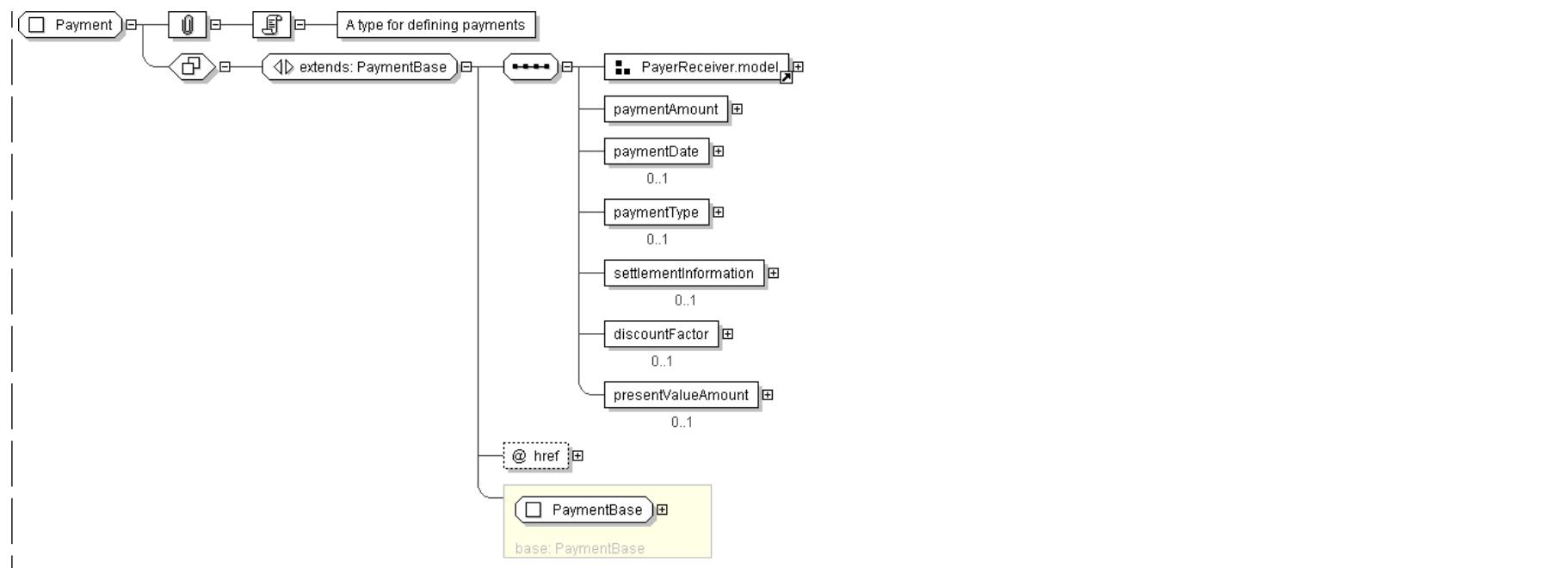
    <discountFactor> xsd:decimal </discountFactor> [0..1]
    'The value representing the discount factor used to calculate the present value of the cash flow.'

    <presentValueAmount> Money </presentValueAmount> [0..1]
    'The amount representing the present value of the forecast payment.'

  </...>

```

### Diagram

**Schema Component Representation**

```

<xsd:complexType name="Payment">
  <xsd:complexContent>
    <xsd:extension base=" PaymentBase ">
      <xsd:sequence>
        <xsd:group ref=" PayerReceiver.model ">
          <xsd:element name="paymentAmount" type=" Money ">
          <xsd:element name="paymentDate" type=" AdjustableOrAdjustedDate " minOccurs="0"/>
          <xsd:element name="paymentType" type=" PaymentType " minOccurs="0"/>
          <xsd:element name="settlementInformation" type=" SettlementInformation " minOccurs="0"/>
          <xsd:element name="discountFactor" type=" xsd:decimal " minOccurs="0"/>
          <xsd:element name="presentValueAmount" type=" Money " minOccurs="0"/>
        </xsd:sequence>
        <xsd:attribute name="href" type=" xsd:IDREF " reference="PricingStructure"/>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>

```

top

**Complex Type: PaymentBase****Super-types:**

None

**Sub-types:**

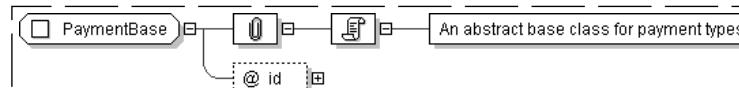
- **Payment** (by extension)
- **PaymentBaseExtended** (by extension)
  - **NonNegativePayment** (by extension)
  - **PositivePayment** (by extension)
- **SimplePayment** (by extension)

<b>Name</b>	PaymentBase
<b>Abstract</b>	yes
<b>Documentation</b>	An abstract base class for payment types.

**XML Instance Representation**

&lt; . . .

```
| id=" xsd:ID [0..1]" />
```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="PaymentBase" abstract="true">
  <xsd:attribute name="id" type=" xsd:ID " />
</xsd:complexType>

```

top

**Complex Type: PaymentBaseExtended**

<b>Super-types:</b>	<a href="#">PaymentBase</a> < <b>PaymentBaseExtended</b> (by extension)
<b>Sub-types:</b>	<ul style="list-style-type: none"> <li>• <a href="#">NonNegativePayment</a> (by extension)</li> <li>• <a href="#">PositivePayment</a> (by extension)</li> </ul>

<b>Name</b>	PaymentBaseExtended
<b>Abstract</b>	yes
<b>Documentation</b>	Base type for payments.

**XML Instance Representation**

```

<...
  id=" xsd:ID [0..1]">
  <payerPartyReference> PartyReference </payerPartyReference> [1]
  'A reference to the party responsible for making the payments defined by this structure.'

  <payerAccountReference> AccountReference </payerAccountReference> [0..1]
  'A reference to the account responsible for making the payments defined by this structure.'

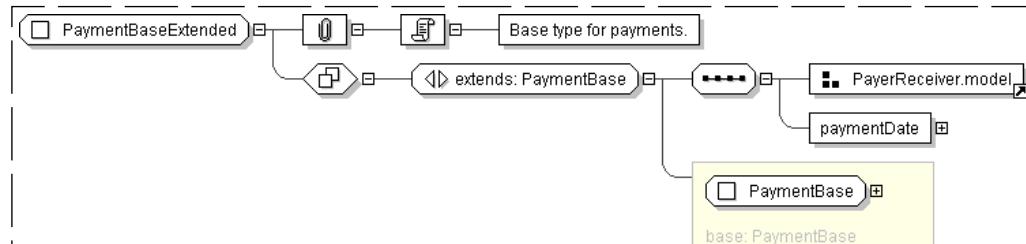
  <receiverPartyReference> PartyReference </receiverPartyReference> [1]
  'A reference to the party that receives the payments corresponding to this structure.'

  <receiverAccountReference> AccountReference </receiverAccountReference> [0..1]
  'A reference to the account that receives the payments corresponding to this structure.'

  <paymentDate> AdjustableOrRelativeDate </paymentDate> [1]
  'The payment date, which can be expressed as either an adjustable or relative date.'

</...>

```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="PaymentBaseExtended" abstract="true">

```

```

<xsd:complexContent>
  <xsd:extension base=" PaymentBase ">
    <xsd:sequence>
      <xsd:group ref=" PayerReceiver.model ">
        <xsd:element name="paymentDate" type=" AdjustableOrRelativeDate " />
      </xsd:sequence>
    </xsd:extension>
</xsd:complexContent>
</xsd:complexType>

```

[top](#)

## Complex Type: [PaymentDetails](#)

<b>Super-types:</b>	None
<b>Sub-types:</b>	None

<b>Name</b>	<a href="#">PaymentDetails</a>
<b>Abstract</b>	no
<b>Documentation</b>	Details on the referenced payment. e.g. Its cashflow components, settlement details.

### XML Instance Representation

```

<...>
  <paymentReference> PaymentReference </paymentReference> [1]
  'The reference to the identified payment strucuture.'

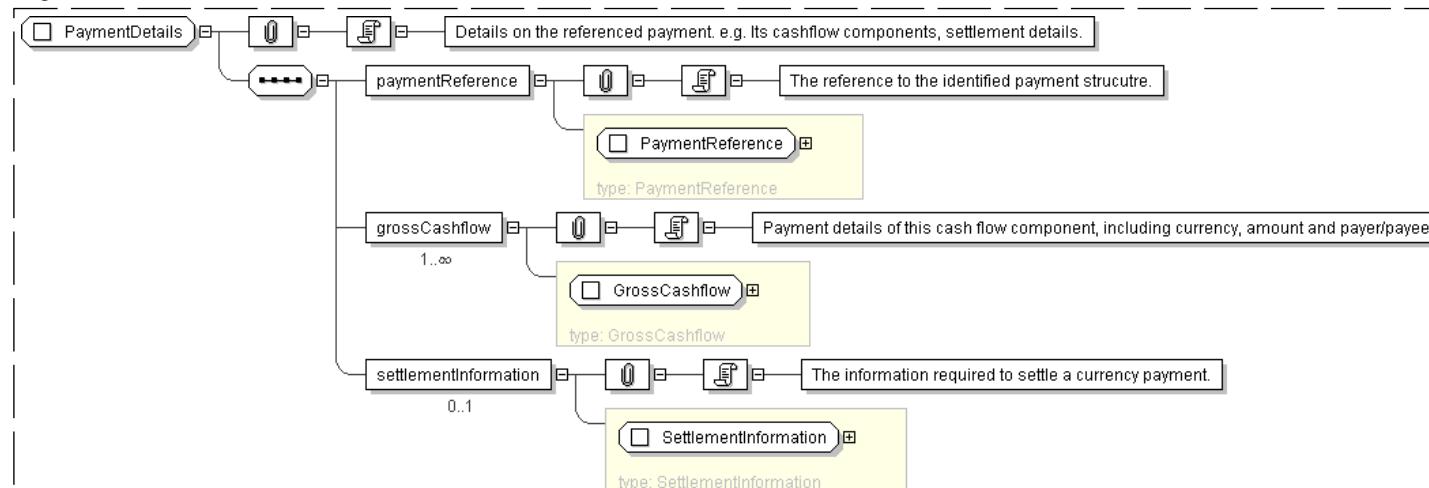
  <grossCashflow> GrossCashflow </grossCashflow> [1..*]
  'Payment details of this cash flow component, including currency, amount and payer/payee.'

  <settlementInformation> SettlementInformation </settlementInformation> [0..1]
  'The information required to settle a currency payment.'

</...>

```

### Diagram



### Schema Component Representation

```

<xsd:complexType name="PaymentDetails">
  <xsd:sequence>
    <xsd:element name="paymentReference" type=" PaymentReference " />
    <xsd:element name="grossCashflow" type=" GrossCashflow " maxOccurs="unbounded"/>

```

```

<xsd:element name="settlementInformation" type=" SettlementInformation " minOccurs="0" />
</xsd:sequence>
</xsd:complexType>

```

## Complex Type: PaymentReference

Super-types:	<a href="#">Reference</a> < <b>PaymentReference</b> (by extension)
Sub-types:	None

Name	PaymentReference
Used by (from the same schema document)	Complex Type <a href="#">PaymentDetails</a>
Abstract	no
Documentation	Reference to a payment.

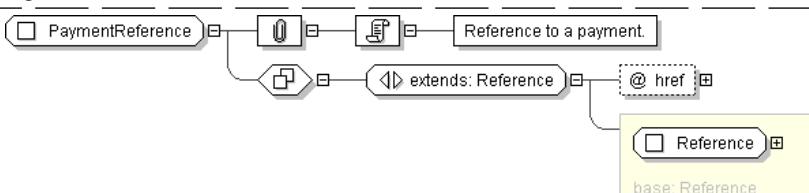
### XML Instance Representation

```

<...
  href=" xsd:IDREF [1]" />

```

### Diagram



### Schema Component Representation

```

<xsd:complexType name="PaymentReference">
  <xsd:complexContent>
    <xsd:extension base=" Reference ">
      <xsd:attribute name="href" type=" xsd:IDREF " use="required" reference="PaymentBase" />
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

## Complex Type: PaymentType

Super-types:	<a href="#">xsd:normalizedString</a> < <a href="#">Scheme</a> (by restriction) < <b>PaymentType</b> (by extension)
Sub-types:	None

Name	PaymentType
Used by (from the same schema document)	Complex Type <a href="#">Payment</a>
Abstract	no

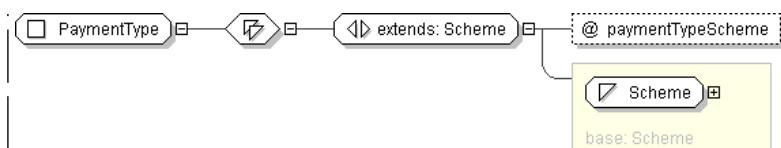
### XML Instance Representation

```

<...
  paymentTypeScheme=" xsd:anyURI [ 0..1 ]">
  Scheme
</...>

```

### Diagram

**Schema Component Representation**

```

<xsd:complexType name="PaymentType">
  <xsd:simpleContent>
    <xsd:extension base=" Scheme ">
      <xsd:attribute name="paymentTypeScheme" type=" xsd:anyURI ">
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
  
```

[top](#)**Complex Type: Period**

Super-types:

None

Sub-types:

- [Offset](#) (by extension)
  - [DateOffset](#) (by extension)
  - [RelativeDateOffset](#) (by extension)
  - [AdjustedRelativeDateOffset](#) (by extension)
  - [RelativeDates](#) (by extension)

**Name**

Period

**Used by (from the same schema document)**Complex Type [ForecastRateIndex](#), Model Group [FloatingRateIndex.model](#)**Abstract**

no

**Documentation**

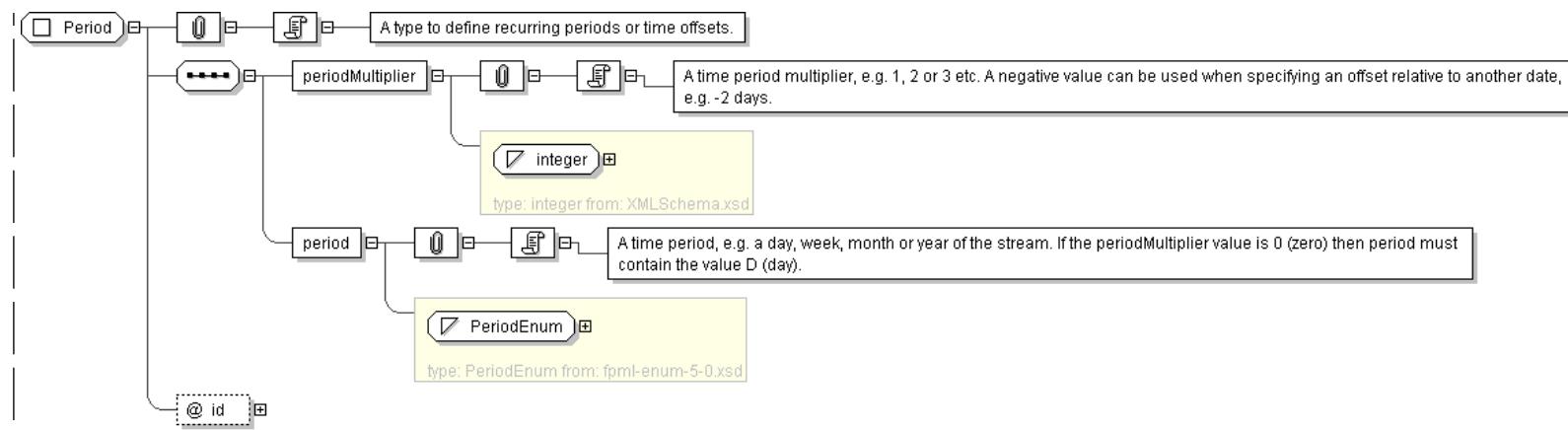
A type to define recurring periods or time offsets.

**XML Instance Representation**

```

<...
  id=" xsd:ID [0..1]">
  <periodMultiplier> xsd:integer </periodMultiplier> [1]
  '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.'
  <period> PeriodEnum </period> [1]
  'A time period, e.g. a day, week, month or year of the stream. If the periodMultiplier value
  is 0 (zero) then period must contain the value D (day)..'
</...>
  
```

**Diagram**



#### Schema Component Representation

```

<xsd:complexType name="Period">
  <xsd:sequence>
    <xsd:element name="periodMultiplier" type="xsd:integer" />
    <xsd:element name="period" type="PeriodEnum" />
  </xsd:sequence>
  <xsd:attribute name="id" type="xsd:ID" />
</xsd:complexType>

```

top

#### Complex Type: PeriodicDates

Super-types:	None
Sub-types:	None

Name	PeriodicDates
Used by (from the same schema document)	Complex Type <a href="#">AdjustableRelativeOrPeriodicDates</a> , Complex Type <a href="#">AdjustableRelativeOrPeriodicDates2</a>
Abstract	no

#### XML Instance Representation

```

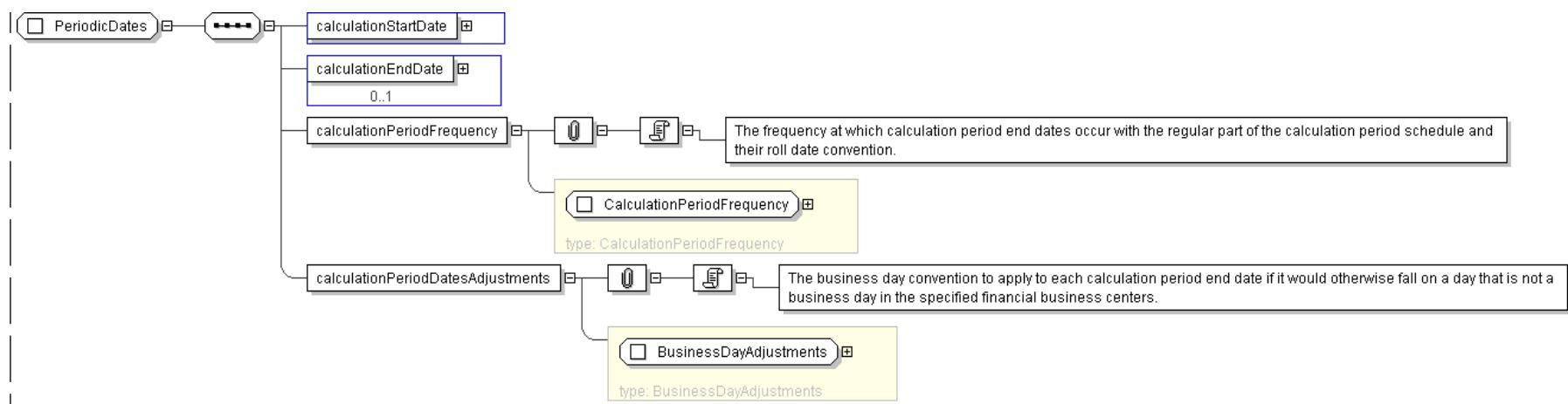
<...>
  <calculationStartDate> AdjustableOrRelativeDate </calculationStartDate> [1]
  <calculationEndDate> AdjustableOrRelativeDate </calculationEndDate> [0..1]
  <calculationPeriodFrequency> CalculationPeriodFrequency </calculationPeriodFrequency> [1]
  'The frequency at which calculation period end dates occur with the regular part of
  the calculation period schedule and their roll date convention.'

  <calculationPeriodDatesAdjustments> BusinessDayAdjustments </
  calculationPeriodDatesAdjustments> [1]
  '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.'

</...>

```

#### Diagram



#### Schema Component Representation

```

<xsd:complexType name="PeriodicDates">
  <xsd:sequence>
    <xsd:element name="calculationStartDate" type=" AdjustableOrRelativeDate " />
    <xsd:element name="calculationEndDate" type=" AdjustableOrRelativeDate " minOccurs="0" />
    <xsd:element name="calculationPeriodFrequency" type=" CalculationPeriodFrequency " />
    <xsd:element name="calculationPeriodDatesAdjustments" type=" BusinessDayAdjustments " />
  </xsd:sequence>
</xsd:complexType>
  
```

top

#### Complex Type: PositiveAmountSchedule

**Super-types:** [PositiveSchedule](#) < **PositiveAmountSchedule** (by extension)

**Sub-types:** None

<b>Name</b>	PositiveAmountSchedule
-------------	------------------------

<b>Abstract</b>	no
-----------------	----

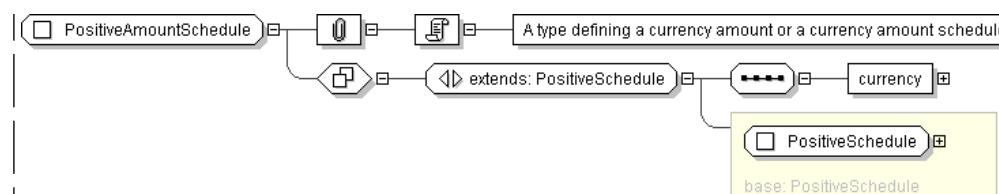
<b>Documentation</b>	A type defining a currency amount or a currency amount schedule.
----------------------	--

#### XML Instance Representation

```

<...>
<id=" xsd:ID [0..1]">
  <initialValue> PositiveDecimal </initialValue> [1]
  'The strictly-positive initial rate or amount, as the case may be. An initial rate of 5%
  would be represented as 0.05.'
  <step> PositiveStep </step> [0..*]
  'The schedule of step date and strictly-positive 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.'
  <currency> Currency </currency> [1]
  'The currency in which an amount is denominated.'
</...>
  
```

#### Diagram

**Schema Component Representation**

```

<xsd:complexType name="PositiveAmountSchedule">
  <xsd:complexContent>
    <xsd:extension base=" PositiveSchedule ">
      <xsd:sequence>
        <xsd:element name="currency" type=" Currency " />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
  
```

[top](#)**Complex Type: PositiveMoney**

<b>Super-types:</b>	<a href="#">MoneyBase</a> < <b>PositiveMoney</b> (by extension)
<b>Sub-types:</b>	None

<b>Name</b>	PositiveMoney
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">PositivePayment</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type defining a positive money amount

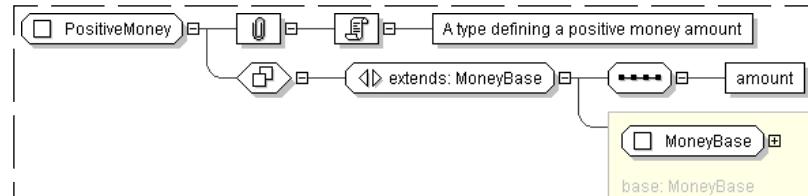
**XML Instance Representation**

```

<...
  id="#ID [0..1]">
  <currency> Currency </currency> [1]
  'The currency in which an amount is denominated.'

  <amount> PositiveDecimal </amount> [1]
  'The positive monetary quantity in currency units.'

</...>
  
```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="PositiveMoney">
  <xsd:complexContent>
    <xsd:extension base=" MoneyBase ">
      <xsd:sequence>
        <xsd:element name="amount" type=" PositiveDecimal " />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
  
```

```
</xsd:complexContent>
</xsd:complexType>
```

top

## Complex Type: PositivePayment

**Super-types:** PaymentBase < PaymentBaseExtended (by extension) < PositivePayment (by extension)

**Sub-types:** None

<b>Name</b>	PositivePayment
-------------	-----------------

<b>Abstract</b>	no
-----------------	----

<b>Documentation</b>	A complex type to specify positive payments.
----------------------	--

### XML Instance Representation

```
<...>
  id="xsd:ID [0..1]">
    <payerPartyReference> PartyReference </payerPartyReference> [1]
    'A reference to the party responsible for making the payments defined by this structure.'

    <payerAccountReference> AccountReference </payerAccountReference> [0..1]
    'A reference to the account responsible for making the payments defined by this structure.'

    <receiverPartyReference> PartyReference </receiverPartyReference> [1]
    'A reference to the party that receives the payments corresponding to this structure.'

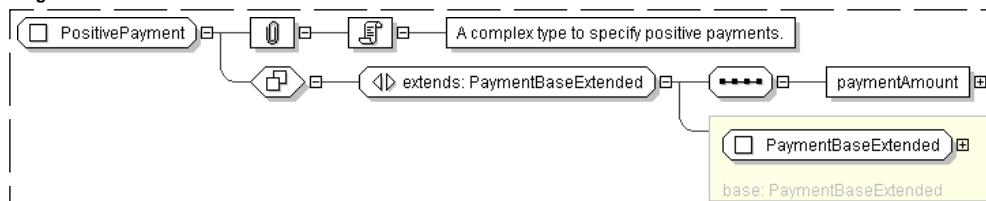
    <receiverAccountReference> AccountReference </receiverAccountReference> [0..1]
    'A reference to the account that receives the payments corresponding to this structure.'

    <paymentDate> AdjustableOrRelativeDate </paymentDate> [1]
    'The payment date, which can be expressed as either an adjustable or relative date.'

    <paymentAmount> PositiveMoney </paymentAmount> [1]
    'Positive payment amount.'

  </...>
```

### Diagram



### Schema Component Representation

```
<xsd:complexType name="PositivePayment">
  <xsd:complexContent>
    <xsd:extension base=" PaymentBaseExtended ">
      <xsd:sequence>
        <xsd:element name="paymentAmount" type=" PositiveMoney "/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

top

**Complex Type: PositiveSchedule**

Super-types:

None

Sub-types:

• [PositiveAmountSchedule](#) (by extension)**Name****Abstract****Documentation**

PositiveSchedule

no

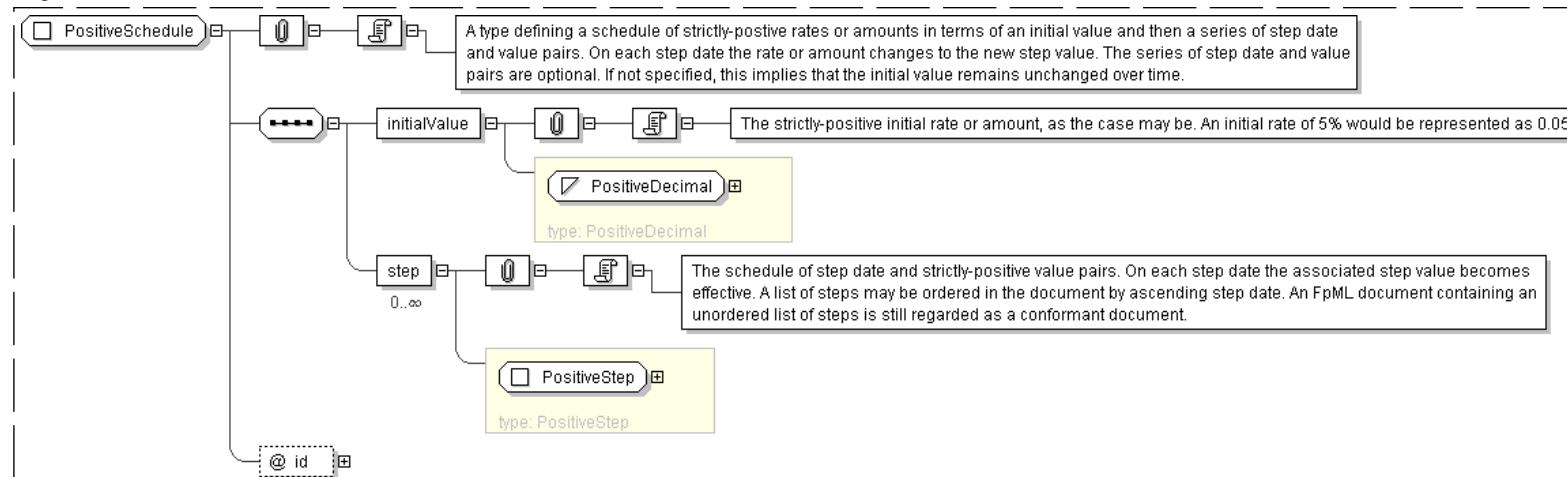
A type defining a schedule of strictly-positive rates or amounts in terms of an initial value and then a series of step date and value pairs. On each step date the rate or amount changes to the new step value. The series of step date and value pairs are optional. If not specified, this implies that the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

**XML Instance Representation**

```
<...>
  id=" xsd:ID [0..1]">
    <initialValue> PositiveDecimal </initialValue> [1]
    'The strictly-positive initial rate or amount, as the case may be. An initial rate of 5%
    would be represented as 0.05.'

    <step> PositiveStep </step> [0..*]
    'The schedule of step date and strictly-positive 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.'

  </...>
```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="PositiveSchedule">
  <xsd:sequence>
    <xsd:element name="initialValue" type=" PositiveDecimal "/>
    <xsd:element name="step" type=" PositiveStep " minOccurs="0" maxOccurs="unbounded"/>
  </xsd:sequence>
  <xsd:attribute name="id" type=" xsd:ID "/>
</xsd:complexType>
  
```

top

**Complex Type: PositiveStep**

**Super-types:** [StepBase](#) < **PositiveStep** (by extension)

**Sub-types:** None

**Name** PositiveStep

**Used by (from the same schema document)** Complex Type [PositiveSchedule](#)

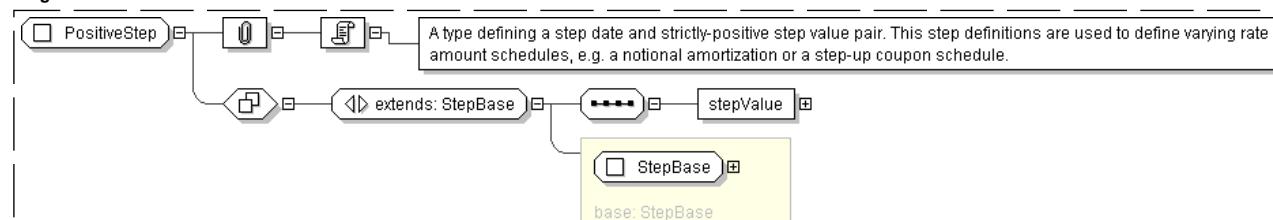
**Abstract** no

**Documentation** A type defining a step date and strictly-positive step value pair. This step definitions are used to define varying rate or amount schedules, e.g. a notional amortization or a step-up coupon schedule.

#### XML Instance Representation

```
<...>
  id="#ID [0..1]">
  <stepDate> xsd:date </stepDate> [1]
    'The date on which the associated stepValue becomes effective. This day may be subject
    to adjustment in accordance with a business day convention.'
  <stepValue> PositiveDecimal </stepValue> [1]
    'The strictly positive rate or amount which becomes effective on the associated stepDate.
    A rate of 5% would be represented as 0.05.'
</...>
```

#### Diagram



#### Schema Component Representation

```

<xsd:complexType name="PositiveStep">
  <xsd:complexContent>
    <xsd:extension base=" StepBase ">
      <xsd:sequence>
        <xsd:element name="stepValue" type=" PositiveDecimal " />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
  
```

[top](#)

## Complex Type: PrevailingTime

**Super-types:** None

**Sub-types:** None

**Name** PrevailingTime

**Used by (from the same schema document)** Complex Type [OffsetPrevailingTime](#)

**Abstract** no

**Documentation** A type for defining a time with respect to a geographic location, for example 11:00 Phoenix, USA. This type should be used where a wider range of locations than those available as business centres is required.

#### XML Instance Representation

```
<...>
  <hourMinuteTime> HourMinuteTime </hourMinuteTime> [1]
    'A time specified in hh:mm:ss format where the second component must be \'00\', e.g. 11am
  
```

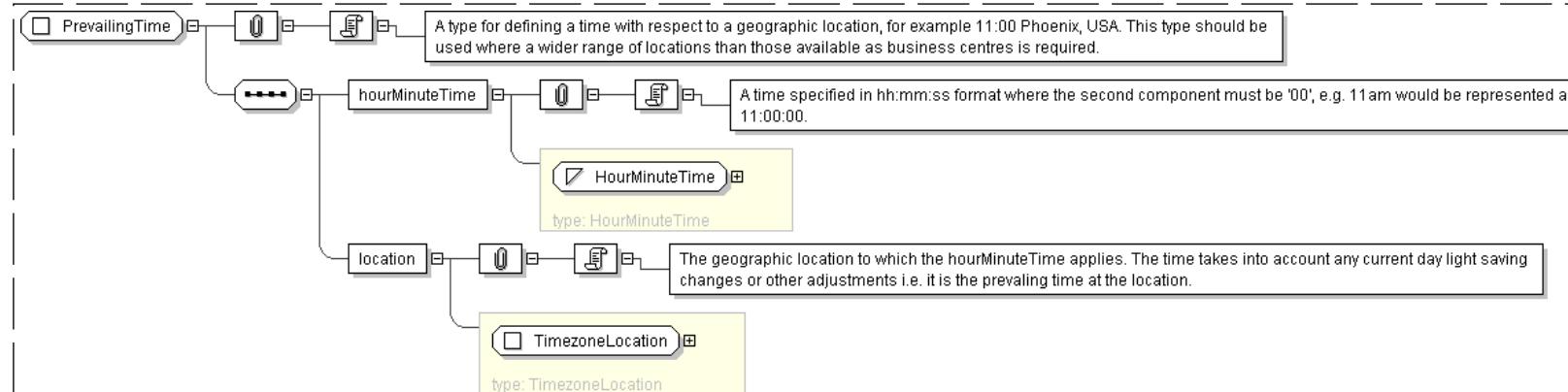
would be represented as 11:00:00.'

<location> TimezoneLocation </location> [1]

'The geographic location to which the hourMinuteTime applies. The time takes into account any current day light saving changes or other adjustments i.e. it is the prevailing time at the location.'

</...>

#### Diagram



#### Schema Component Representation

```

<xsd:complexType name="PrevailingTime">
  <xsd:sequence>
    <xsd:element name="hourMinuteTime" type=" HourMinuteTime " />
    <xsd:element name="location" type=" TimezoneLocation " />
  </xsd:sequence>
</xsd:complexType>

```

[top](#)

## Complex Type: PricingStructure

Super-types:

None

Sub-types:

None

Name	PricingStructure
------	------------------

Abstract	yes
----------	-----

Documentation	An abstract pricing structure base type. Used as a base for structures such as yield curves and volatility matrices.
---------------	--

#### XML Instance Representation

```

<...
  id=" xsd:ID [0..1]">
  <name> xsd:normalizedString </name> [0..1]
  'The name of the structure, e.g \'USDLIBOR-3M EOD Curve\' .'

```

```

  <currency> Currency </currency> [0..1]

```

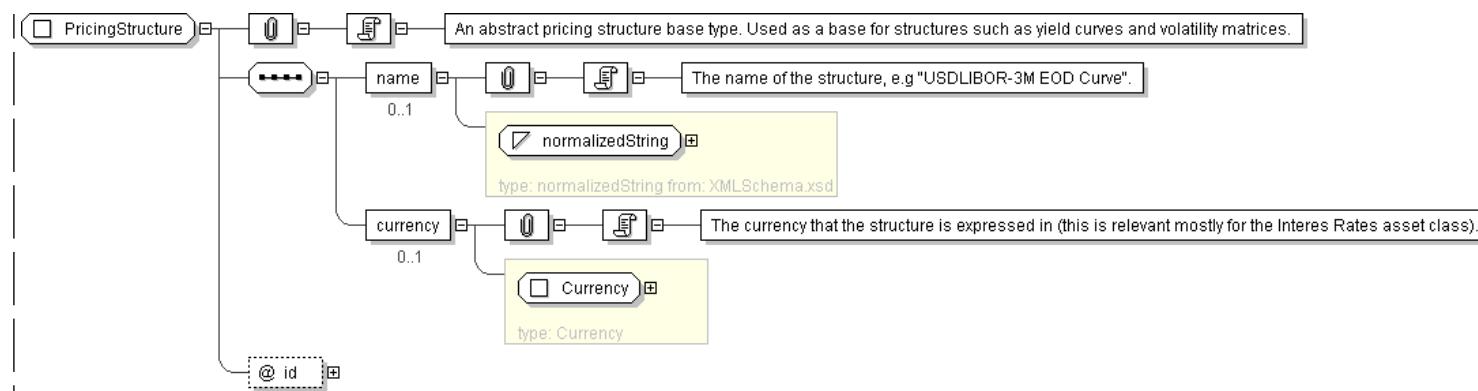
```

  'The currency that the structure is expressed in (this is relevant mostly for the Interest Rates asset class).'

```

</...>

#### Diagram

**Schema Component Representation**

```

<xsd:complexType name="PricingStructure" abstract="true">
  <xsd:sequence>
    <xsd:element name="name" type="xsd:normalizedString" minOccurs="0"/>
    <xsd:element name="currency" type="Currency" minOccurs="0"/>
  </xsd:sequence>
  <xsd:attribute name="id" type="xsd:ID" />
</xsd:complexType>

```

[top](#)**Complex Type: PricingStructureReference**

<b>Super-types:</b>	<a href="#">Reference</a> < <b>PricingStructureReference</b> (by extension)
<b>Sub-types:</b>	None

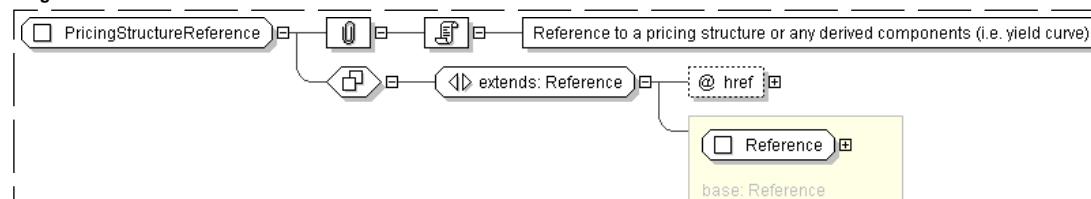
<b>Name</b>	PricingStructureReference
<b>Abstract</b>	no
<b>Documentation</b>	Reference to a pricing structure or any derived components (i.e. yield curve).

**XML Instance Representation**

```

<...
  href=" xsd:IDREF [1]" />

```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="PricingStructureReference">
  <xsd:complexContent>
    <xsd:extension base=" Reference ">
      <xsd:attribute name="href" type="xsd:IDREF" use="required" reference="PricingStructure" />
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

[top](#)

**Complex Type: PrincipalExchanges**

**Super-types:** None  
**Sub-types:** None

**Name** PrincipalExchanges

**Abstract** no

**Documentation** A type defining which principal exchanges occur for the stream.

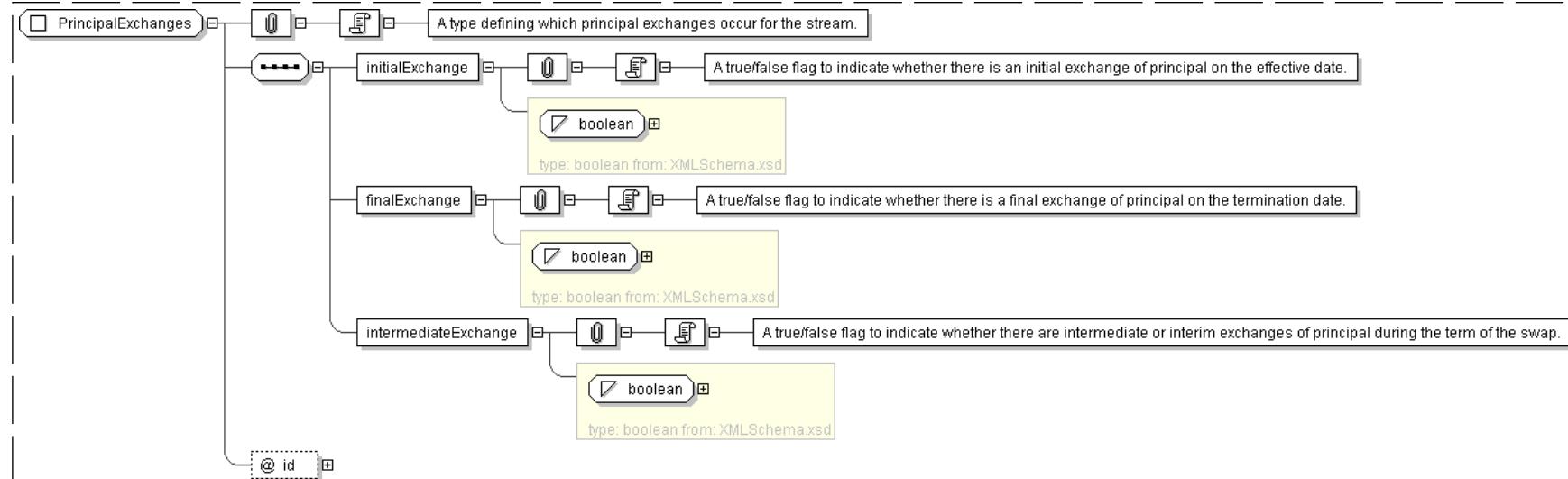
**XML Instance Representation**

```
<...>
<id=" xsd:ID [0..1]">
<initialExchange> xsd:boolean </initialExchange> [1]
'A true/false flag to indicate whether there is an initial exchange of principal on
the effective date.'

<finalExchange> xsd:boolean </finalExchange> [1]
'A true/false flag to indicate whether there is a final exchange of principal on
the termination date.'

<intermediateExchange> xsd:boolean </intermediateExchange> [1]
'A true/false flag to indicate whether there are intermediate or interim exchanges of
principal during the term of the swap.'

</...>
```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="PrincipalExchanges">
  <xsd:sequence>
    <xsd:element name="initialExchange" type="xsd:boolean" />
    <xsd:element name="finalExchange" type="xsd:boolean" />
    <xsd:element name="intermediateExchange" type="xsd:boolean" />
  </xsd:sequence>
  <xsd:attribute name="id" type="xsd:ID" use="optional" />
</xsd:complexType>

```

## Complex Type: Product

Super-types:	None
Sub-types:	None

Name	Product
Used by (from the same schema document)	Element <a href="#">product</a>
Abstract	yes
Documentation	The base type which all FpML products extend.

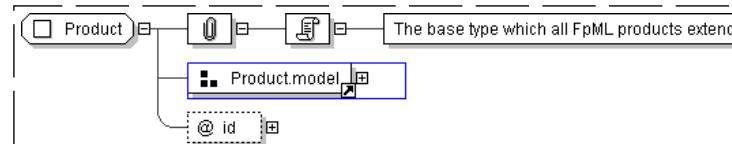
### XML Instance Representation

```
<...>
<id=" xsd:ID [0..1]">
<productType> ProductType </productType> [0..*]
'A classification of the type of product. FpML defines a simple product categorization using
a coding scheme.'

<productId> ProductId </productId> [0..*]
'A product 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.

</...>
```

### Diagram



### Schema Component Representation

```
<xsd:complexType name="Product" abstract="true">
  <xsd:group ref="#Product.model"/>
  <xsd:attribute name="id" type="xsd:ID" />
</xsd:complexType>
```

## Complex Type: ProductId

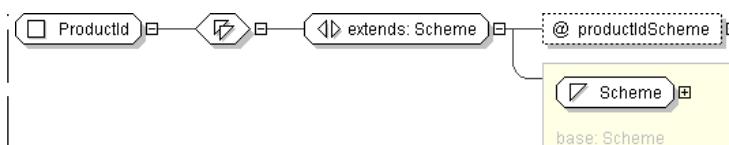
Super-types:	<a href="#">xsd:normalizedString</a> < <a href="#">Scheme</a> (by restriction) < <b>ProductId</b> (by extension)
Sub-types:	None

Name	ProductId
Used by (from the same schema document)	Model Group <a href="#">Product.model</a>
Abstract	no

### XML Instance Representation

```
<...>
<productIdScheme=" xsd:anyURI [0..1]">
  Scheme
</...>
```

### Diagram

**Schema Component Representation**

```

<xsd:complexType name="ProductId">
  <xsd:simpleContent>
    <xsd:extension base=" Scheme ">
      <xsd:attribute name="productIdScheme" type=" xsd:anyURI "/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
  
```

[top](#)**Complex Type: ProductReference**

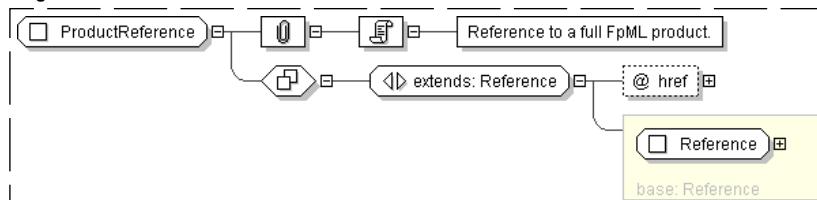
Super-types:	<a href="#">Reference</a> < <b>ProductReference</b> (by extension)
Sub-types:	None

Name	ProductReference
Abstract	no
Documentation	Reference to a full FpML product.

**XML Instance Representation**

```

<...
  href=" xsd:IDREF [1]" />
  
```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="ProductReference">
  <xsd:complexContent>
    <xsd:extension base=" Reference ">
      <xsd:attribute name="href" type=" xsd:IDREF " use="required" reference="Product"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
  
```

[top](#)**Complex Type: ProductType**

Super-types:	<a href="#">xsd:normalizedString</a> < <a href="#">Scheme</a> (by restriction) < <b>ProductType</b> (by extension)
Sub-types:	None

Name	ProductType
Used by (from the same schema document)	Model Group <a href="#">Product.model</a>

**Abstract**

no

**XML Instance Representation**

```
<...>
<productTypeScheme=" xsd:anyURI [ 0..1 ]>
Scheme
</...>
```

**Diagram****Schema Component Representation**

```
<xsd:complexType name="ProductType">
  <xsd:simpleContent>
    <xsd:extension base=" Scheme ">
      <xsd:attribute name="productTypeScheme" type=" xsd:anyURI " default="http://www.fpml.org/
        coding-scheme/product-type-simple"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
```

[top](#)**Complex Type: QuotedCurrencyPair**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None

<b>Name</b>	QuotedCurrencyPair
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">FxFixing</a> , Complex Type <a href="#">FxRate</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type that describes the composition of a rate that has been quoted or is to be quoted. This includes the two currencies and the quotation relationship between the two currencies and is used as a building block throughout the FX specification.

**XML Instance Representation**

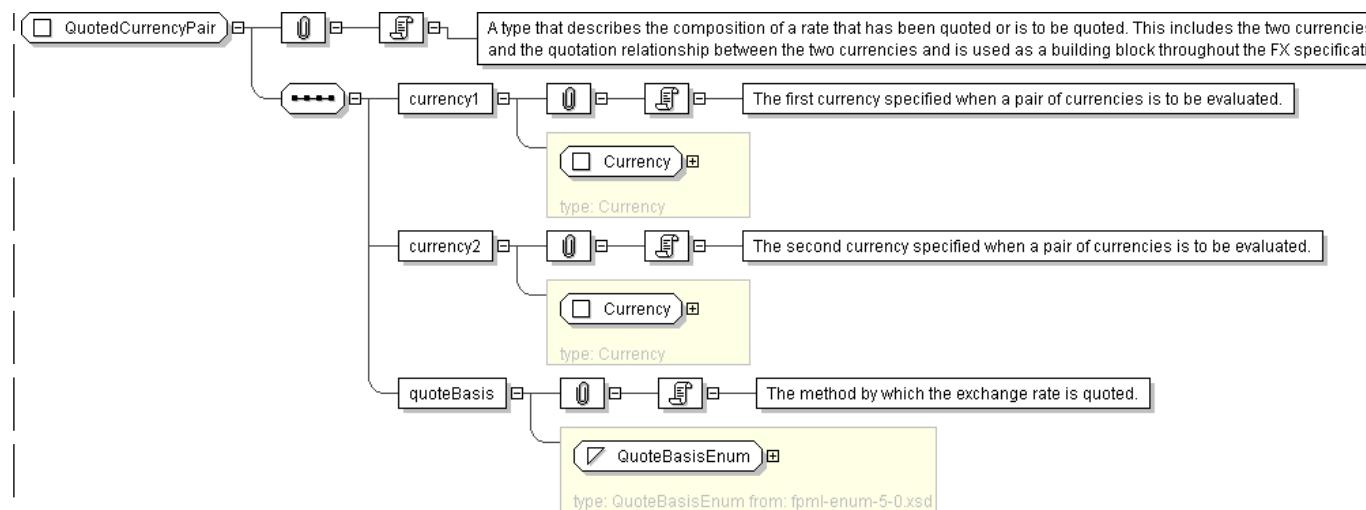
```
<...>
<curreny1> Currency </curreny1> [1]
'The first currency specified when a pair of currencies is to be evaluated.'

<curreny2> Currency </curreny2> [1]
'The second currency specified when a pair of currencies is to be evaluated.'

<quoteBasis> QuoteBasisEnum </quoteBasis> [1]
'The method by which the exchange rate is quoted.'

</...>
```

**Diagram**

**Schema Component Representation**

```

<xsd:complexType name="QuotedCurrencyPair">
  <xsd:sequence>
    <xsd:element name="currency1" type="Currency" />
    <xsd:element name="currency2" type="Currency" />
    <xsd:element name="quoteBasis" type="QuoteBasisEnum" />
  </xsd:sequence>
</xsd:complexType>
  
```

top

**Complex Type: Rate**

Super-types:

None

Sub-types:

- [FloatingRate](#) (by extension)
  - [FloatingRateCalculation](#) (by extension)

**Name**

Rate

**Abstract**

yes

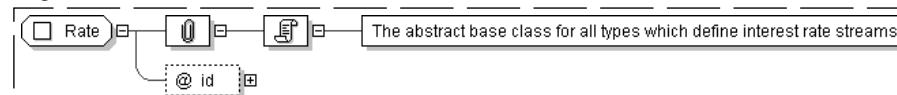
**Documentation**

The abstract base class for all types which define interest rate streams.

**XML Instance Representation**

```

<...>
  id=" xsd:ID [0..1]" />
  
```

**Diagram****Schema Component Representation**

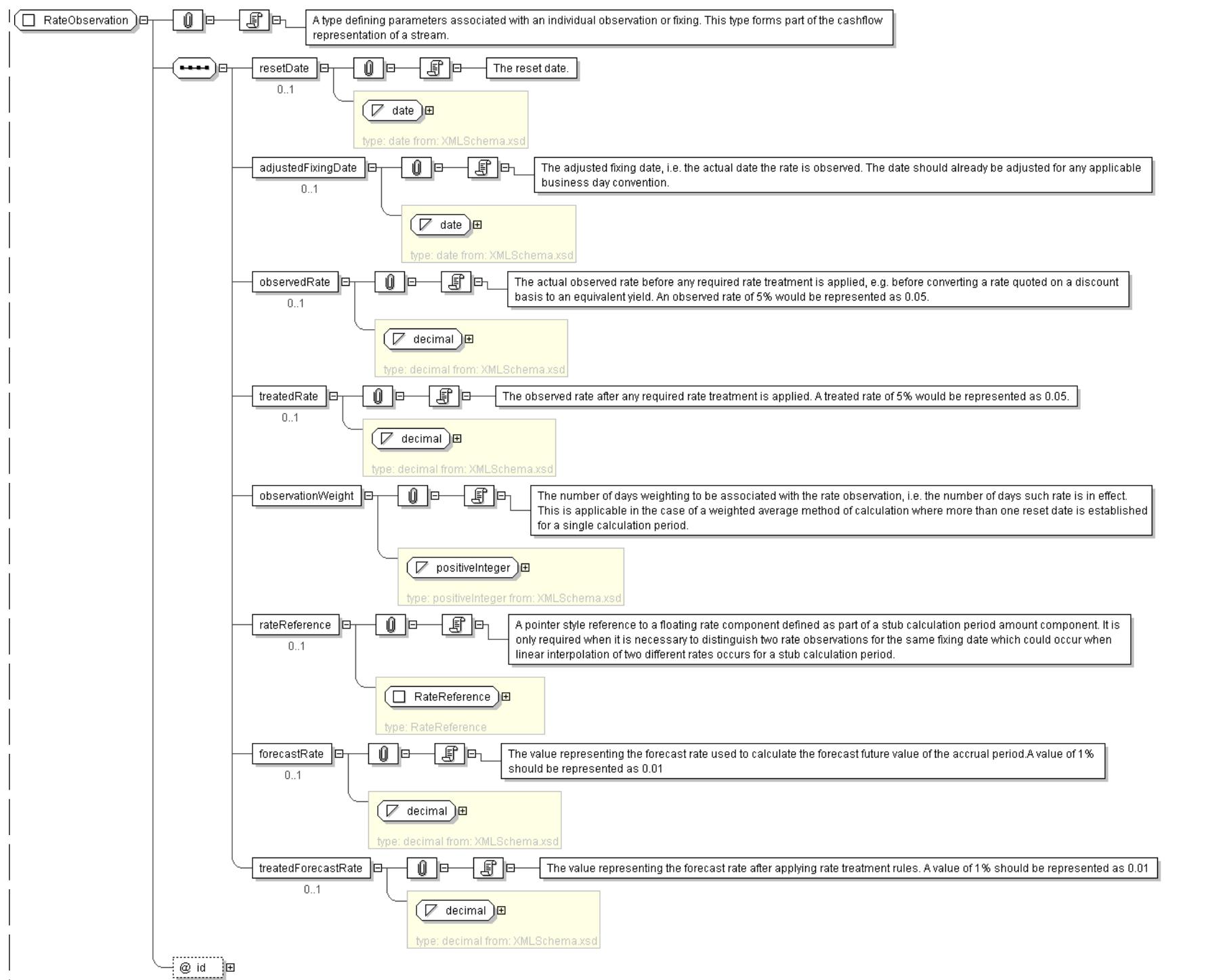
```

<xsd:complexType name="Rate" abstract="true">
  <xsd:attribute name="id" type="xsd:ID" />
</xsd:complexType>
  
```

top

**Complex Type: RateObservation**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None
<b>Name</b>	RateObservation
<b>Abstract</b>	no
<b>Documentation</b>	A type defining parameters associated with an individual observation or fixing. This type forms part of the cashflow representation of a stream.
<b>XML Instance Representation</b>	
<...>	
< <b>id</b> = <b>xsd:ID</b> [0..1]>	
< <b>resetDatexsd:date</b> </ <b>resetDate</b> > [0..1]	
'The reset date.'	
< <b>adjustedFixingDatexsd:date</b> </ <b>adjustedFixingDate</b> > [0..1]	
'The adjusted fixing date, i.e. the actual date the rate is observed. The date should already be adjusted for any applicable business day convention.'	
< <b>observedRatexsd:decimal</b> </ <b>observedRate</b> > [0..1]	
'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.'	
< <b>treatedRatexsd:decimal</b> </ <b>treatedRate</b> > [0..1]	
'The observed rate after any required rate treatment is applied. A treated rate of 5% would be represented as 0.05.'	
< <b>observationWeightxsd:positiveInteger</b> </ <b>observationWeight</b> > [1]	
'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.'	
< <b>rateReferenceRateReference &lt;/<b>rateReference</b>&gt; [0..1]</b>	
'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.'	
< <b>forecastRatexsd:decimal</b> </ <b>forecastRate</b> > [0..1]	
'The value representing the forecast rate used to calculate the forecast future value of the accrual period. A value of 1% should be represented as 0.01'	
< <b>treatedForecastRatexsd:decimal</b> </ <b>treatedForecastRate</b> > [0..1]	
'The value representing the forecast rate after applying rate treatment rules. A value of 1% should be represented as 0.01'	
</...>	
<b>Diagram</b>	

**Schema Component Representation**

```
<xsd:complexType name="RateObservation">
```

```

<xsd:sequence>
  <xsd:element name="resetDate" type="xsd:date" minOccurs="0"/>
  <xsd:element name="adjustedFixingDate" type="xsd:date" minOccurs="0"/>
  <xsd:element name="observedRate" type="xsd:decimal" minOccurs="0"/>
  <xsd:element name="treatedRate" type="xsd:decimal" minOccurs="0"/>
  <xsd:element name="observationWeight" type="xsd:positiveInteger"/>
  <xsd:element name="rateReference" type="RateReference" minOccurs="0"/>
  <xsd:element name="forecastRate" type="xsd:decimal" minOccurs="0"/>
  <xsd:element name="treatedForecastRate" type="xsd:decimal" minOccurs="0"/>
</xsd:sequence>
<xsd:attribute name="id" type="xsd:ID"/>
</xsd:complexType>

```

[top](#)

## Complex Type: RateReference

<b>Super-types:</b>	None
<b>Sub-types:</b>	None

<b>Name</b>	RateReference
-------------	---------------

<b>Used by (from the same schema document)</b>	Complex Type <a href="#">RateObservation</a>
--	--

<b>Abstract</b>	no
-----------------	----

<b>Documentation</b>	Reference to any rate (floating, inflation) derived from the abstract Rate component.
----------------------	---

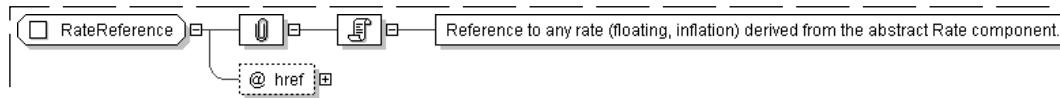
### XML Instance Representation

```

<...
  href=" xsd:IDREF [1]" />

```

### Diagram



### Schema Component Representation

```

<xsd:complexType name="RateReference">
  <xsd:attribute name="href" type="xsd:IDREF" use="required" reference="Rate" />
</xsd:complexType>

```

[top](#)

## Complex Type: RateSourcePage

<b>Super-types:</b>	<a href="#">xsd:normalizedString</a> < <a href="#">Scheme</a> (by restriction) < <b>RateSourcePage</b> (by extension)
<b>Sub-types:</b>	None

<b>Name</b>	RateSourcePage
-------------	----------------

<b>Used by (from the same schema document)</b>	Complex Type <a href="#">InformationSource</a>
--	--

<b>Abstract</b>	no
-----------------	----

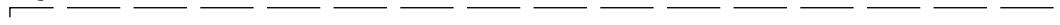
### XML Instance Representation

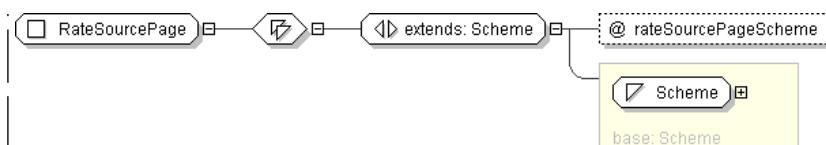
```

<...
  rateSourcePageScheme=" xsd:anyURI [0..1]">
  Scheme
</...>

```

### Diagram



**Schema Component Representation**

```

<xsd:complexType name="RateSourcePage">
  <xsd:simpleContent>
    <xsd:extension base=" Scheme ">
      <xsd:attribute name="rateSourcePageScheme" type=" xsd:anyURI " />
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>

```

top

**Complex Type: Reference****Super-types:**

None

**Sub-types:**

- [AccountReference](#) (by extension)
- [AmountReference](#) (by extension)
- [BusinessCentersReference](#) (by extension)
- [BusinessDayAdjustmentsReference](#) (by extension)
- [DateReference](#) (by extension)
- [DeterminationMethodReference](#) (by extension)
- [IdentifiedCurrencyReference](#) (by extension)
- [LegalEntityReference](#) (by extension)
- [NotionalAmountReference](#) (by extension)
- [NotionalReference](#) (by extension)
- [PartyReference](#) (by extension)
- [PaymentReference](#) (by extension)
- [PricingStructureReference](#) (by extension)
- [ProductReference](#) (by extension)
- [ReturnSwapNotionalAmountReference](#) (by extension)
- [ScheduleReference](#) (by extension)
- [SpreadScheduleReference](#) (by extension)

**Name**

Reference

**Abstract**

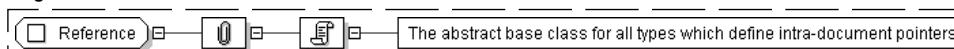
yes

**Documentation**

The abstract base class for all types which define intra-document pointers.

**XML Instance Representation**

&lt;... /&gt;

**Diagram****Schema Component Representation**

```

<xsd:complexType name="Reference" abstract="true">

```

top

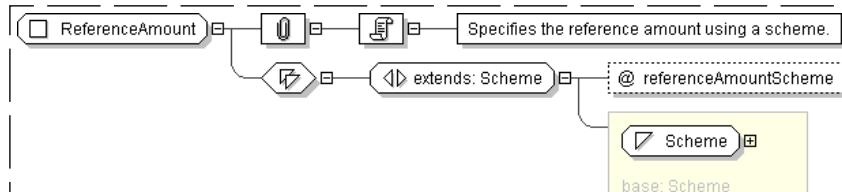
**Complex Type: ReferenceAmount****Super-types:**[xsd:normalizedString](#) < [Scheme](#) (by restriction) < **ReferenceAmount** (by extension)**Sub-types:**

None

<b>Name</b>	ReferenceAmount
<b>Abstract</b>	no
<b>Documentation</b>	Specifies the reference amount using a scheme.

**XML Instance Representation**

```
<...>
<referenceAmountScheme=" xsd:anyURI [0..1]">
  Scheme
</...>
```

**Diagram****Schema Component Representation**

```
<xsd:complexType name="ReferenceAmount">
  <xsd:simpleContent>
    <xsd:extension base=" Scheme ">
      <xsd:attribute name="referenceAmountScheme" type=" xsd:anyURI " />
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
```

[top](#)**Complex Type: ReferenceBank**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None

<b>Name</b>	ReferenceBank
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">CashSettlementReferenceBanks</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type to describe an institution (party) identified by means of a coding scheme and an optional name.

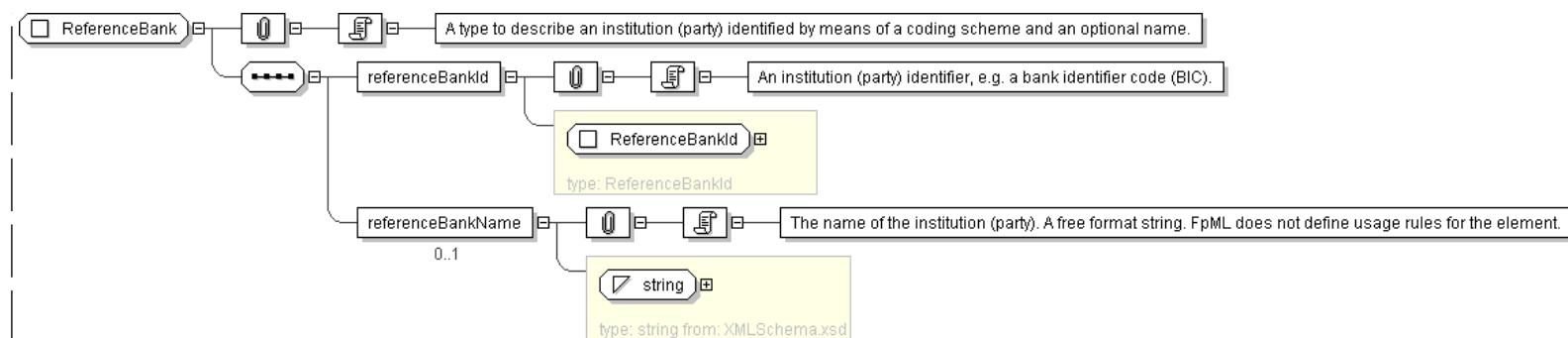
**XML Instance Representation**

```
<...>
<referenceBankId> ReferenceBankId </referenceBankId> [1]
'An institution (party) identifier, e.g. a bank identifier code (BIC).'

<referenceBankName> xsd:string </referenceBankName> [0..1]
'The name of the institution (party). A free format string. FpML does not define usage
rules for the element.'

</...>
```

**Diagram**

**Schema Component Representation**

```

<xsd:complexType name="ReferenceBank">
  <xsd:sequence>
    <xsd:element name="referenceBankId" type="#ReferenceBankId" />
    <xsd:element name="referenceBankName" type="xsd:string" minOccurs="0" />
  </xsd:sequence>
</xsd:complexType>
  
```

[top](#)**Complex Type: ReferenceBankId**

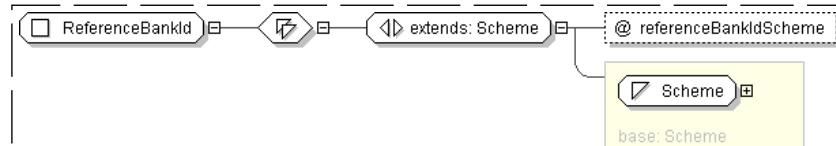
<b>Super-types:</b>	<a href="#">xsd:normalizedString</a> < <a href="#">Scheme</a> (by restriction) < <b>ReferenceBankId</b> (by extension)
<b>Sub-types:</b>	None

<b>Name</b>	ReferenceBankId
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">ReferenceBank</a>
<b>Abstract</b>	no

**XML Instance Representation**

```

<...
  referenceBankIdScheme=" xsd:anyURI [0..1]">
  Scheme
</...>
  
```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="ReferenceBankId">
  <xsd:simpleContent>
    <xsd:extension base="#Scheme">
      <xsd:attribute name="referenceBankIdScheme" type="xsd:anyURI" />
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
  
```

[top](#)**Complex Type: RelatedParty**

**Super-types:** None

**Sub-types:** None

<b>Name</b>	RelatedParty
-------------	--------------

<b>Abstract</b>	no
-----------------	----

#### XML Instance Representation

```
<...>
<partyReference> PartyReference </partyReference> [1]
'Reference to a party.'

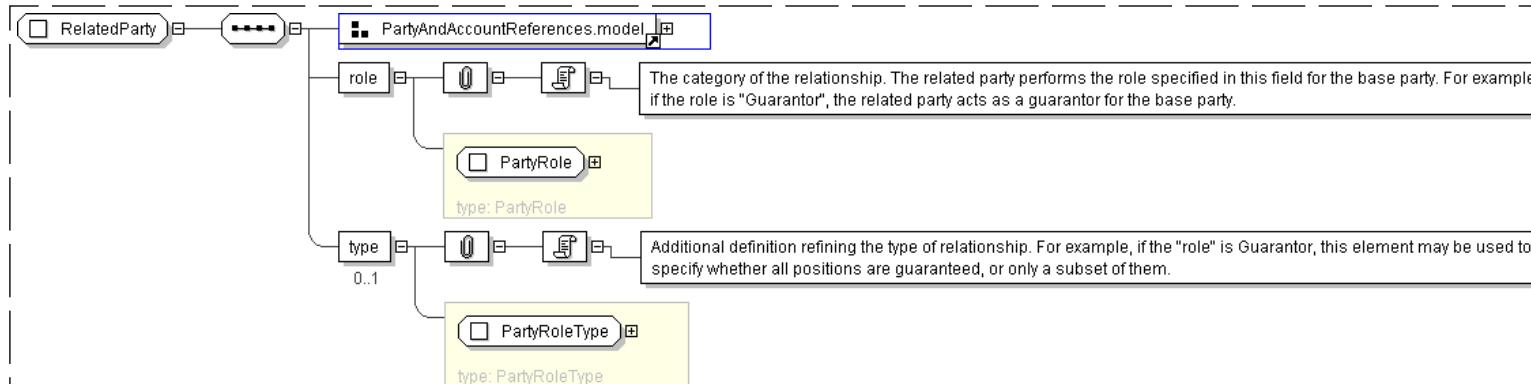
<accountReference> AccountReference </accountReference> [0..1]
'Reference to an account.'

<role> PartyRole </role> [1]
'The category of the relationship. The related party performs the role specified in this
field for the base party. For example, if the role is \"Guarantor\", the related party acts
as a guarantor for the base party.'

<type> PartyRoleType </type> [0..1]
'Additional definition refining the type of relationship. For example, if the \"role\" is
Guarantor, this element may be used to specify whether all positions are guaranteed, or
only a subset of them.'

</...>
```

#### Diagram



#### Schema Component Representation

```
<xsd:complexType name="RelatedParty">
  <xsd:sequence>
    <xsd:group ref=" PartyAndAccountReferences.model " />
    <xsd:element name="role" type=" PartyRole " />
    <xsd:element name="type" type=" PartyRoleType " minOccurs="0" />
  </xsd:sequence>
</xsd:complexType>
```

top

#### Complex Type: RelativeDateOffset

**Super-types:**

Period < Offset (by extension) < **RelativeDateOffset** (by extension)

**Sub-types:**

- AdjustedRelativeDateOffset (by extension)
- RelativeDates (by extension)

<b>Name</b>	RelativeDateOffset
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">AdjustableDatesOrRelativeDateOffset</a> , Complex Type <a href="#">AdjustableOrRelativeDate</a> , Complex Type <a href="#">ExerciseFee</a> , Complex Type <a href="#">ExerciseFeeSchedule</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type 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 or 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 specifies any required adjustment to the unadjusted derived date. A negative or positive value in the periodMultiplier indicates whether the unadjusted derived precedes or follows the anchor date. The businessDayConvention should contain a value NONE if the day type 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).

**XML Instance Representation**

```

<...
  id=" xsd:ID [0..1]">
  <periodMultiplier> xsd:integer </periodMultiplier> [1]
  '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.'

  <period> PeriodEnum </period> [1]
  'A time period, e.g. a day, week, month or year of the stream. If the periodMultiplier value
  is 0 (zero) then period must contain the value D (day)..'

  <dayType> DayTypeEnum </dayType> [0..1]
  '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.'

  <businessDayConvention> BusinessDayConventionEnum </businessDayConvention> [1]
  'The convention for adjusting a date if it would otherwise fall on a day that is not a
  business day.'

Start Group: BusinessCentersOrReference.model [0..1]
Start Choice [1]
  <businessCentersReference> BusinessCentersReference </businessCentersReference> [1]
  '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.'

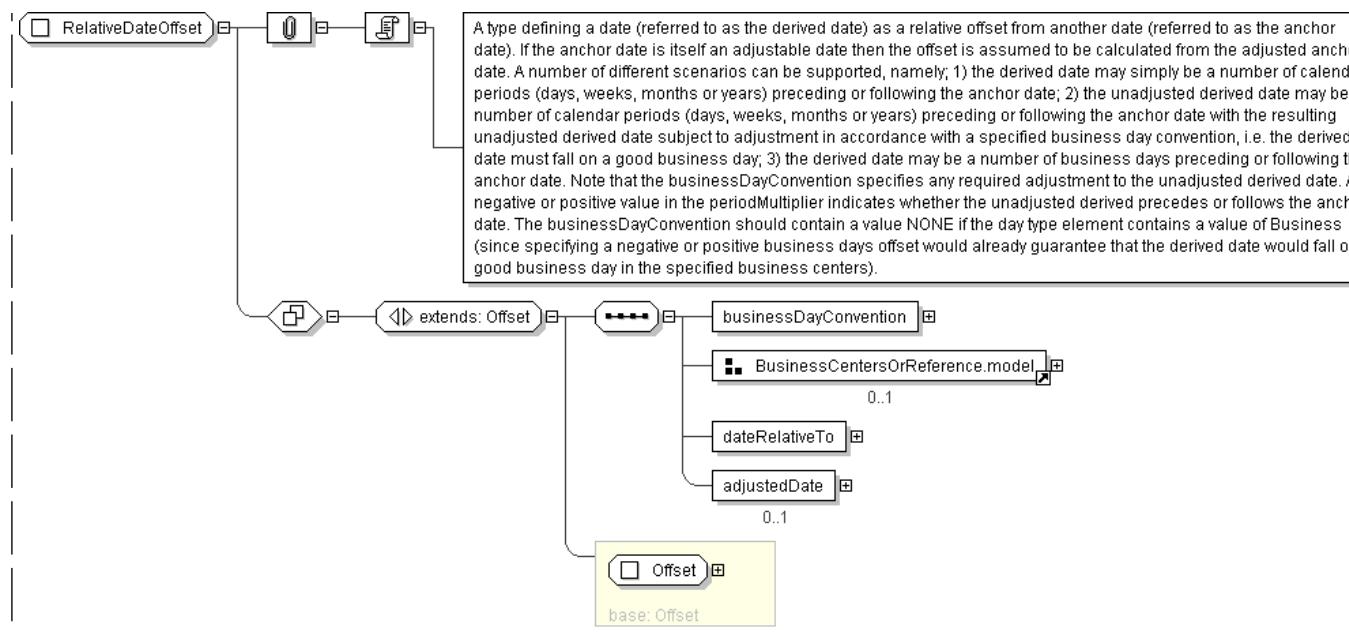
  <businessCenters> BusinessCenters </businessCenters> [1]
End Choice
End Group: BusinessCentersOrReference.model
  <dateRelativeTo> DateReference </dateRelativeTo> [1]
  'Specifies the anchor as an href attribute. The href attribute value is a pointer
  style reference to the element or component elsewhere in the document where the anchor date
  is defined.'

  <adjustedDate> IdentifiedDate </adjustedDate> [0..1]
  'The date once the adjustment has been performed. (Note that this date may change if
  the business center holidays change.).'

</...>

```

**Diagram**



#### Schema Component Representation

```

<xsd:complexType name="RelativeDateOffset">
  <xsd:complexContent>
    <xsd:extension base=" Offset ">
      <xsd:sequence>
        <xsd:element name="businessDayConvention" type="BusinessDayConventionEnum" />
        <xsd:group ref="BusinessCentersOrReference.model" minOccurs="0"/>
        <xsd:element name="dateRelativeTo" type="DateReference" />
        <xsd:element name="adjustedDate" type="IdentifiedDate" minOccurs="0"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

top

#### Complex Type: `RelativeDateSequence`

Super-types:	None
Sub-types:	None

Name	RelativeDateSequence
Used by (from the same schema document)	Complex Type <a href="#">AdjustableRelativeOrPeriodicDates</a>
Abstract	no
Documentation	A type describing a date when this date is defined in reference to another date through one or several date offsets.

#### XML Instance Representation

```

<...>
  <dateRelativeTo> DateReference </dateRelativeTo> [1]
  'Specifies the anchor as the href attribute. The href attribute value is a pointer
  style reference to the element or component elsewhere in the document where the anchor date
  is defined.'
  <dateOffset> DateOffset </dateOffset> [1..*]
  Start Group: BusinessCentersOrReference.model [0..1]

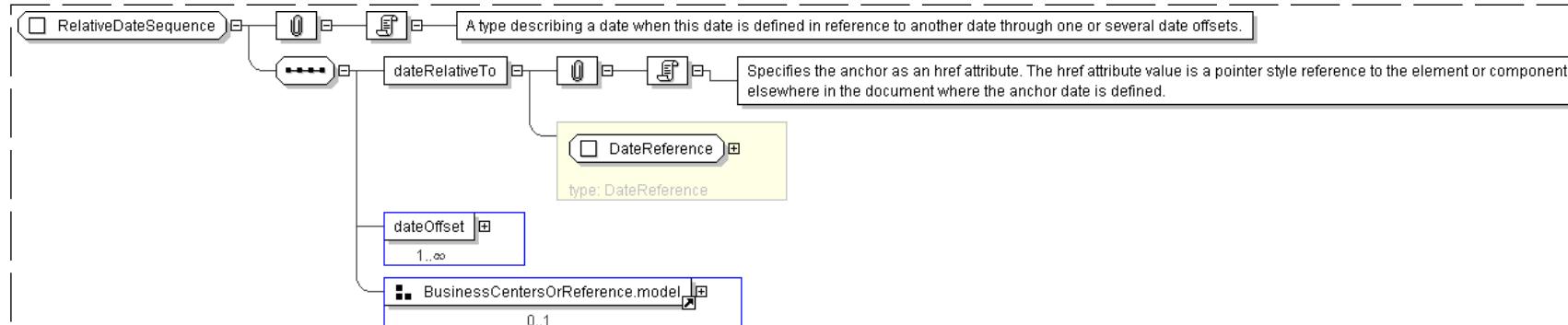
```

| Start Choice [1]  
 <businessCentersReference> BusinessCentersReference </businessCentersReference> [1]

'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.'

<businessCenters> BusinessCenters </businessCenters> [1]

End Choice  
 End Group: BusinessCentersOrReference.model  
 </...>

**Diagram****Schema Component Representation**

```

<xsd:complexType name="RelativeDateSequence">
  <xsd:sequence>
    <xsd:element name="dateRelativeTo" type="#DateReference" />
    <xsd:element name="dateOffset" type="#DateOffset" maxOccurs="unbounded" />
    <xsd:group ref="BusinessCentersOrReference.model" minOccurs="0" />
  </xsd:sequence>
</xsd:complexType>
  
```

top

**Complex Type: RelativeDates**

**Super-types:** Period < Offset (by extension) < RelativeDateOffset (by extension) < **RelativeDates** (by extension)

**Sub-types:** None

<b>Name</b>	RelativeDates
-------------	---------------

<b>Used by (from the same schema document)</b>	Complex Type <a href="#">AdjustableOrRelativeDates</a> , Complex Type <a href="#">AdjustableRelativeOrPeriodicDates2</a>
--	--

<b>Abstract</b>	no
-----------------	----

<b>Documentation</b>	A type describing a set of dates defined as relative to another set of dates.
----------------------	---

**XML Instance Representation**

```

<...
  id="xsd:ID [0..1]">
  <periodMultiplier> xsd:integer </periodMultiplier> [1]
  '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.'
  
```

```

  <period> PeriodEnum </period> [1]
  'A time period, e.g. a day, week, month or year of the stream. If the periodMultiplier value
  is 0 (zero) then period must contain the value D (day)..'
  
```

```

  <dayType> DayTypeEnum </dayType> [0..1]
  '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.'

<businessDayConvention> BusinessDayConventionEnum </businessDayConvention> [1]  
'The convention for adjusting a date if it would otherwise fall on a day that is not a business day.'

Start Group: BusinessCentersOrReference.model [0..1]  
Start Choice [1]  
<businessCentersReference> BusinessCentersReference </businessCentersReference> [1]  
'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.'

<businessCenters> BusinessCenters </businessCenters> [1]

End Choice  
End Group: BusinessCentersOrReference.model

<dateRelativeTo> DateReference </dateRelativeTo> [1]  
'Specifies the anchor as an href attribute. The href attribute value is a pointer style reference to the element or component elsewhere in the document where the anchor date is defined.'

<adjustedDate> IdentifiedDate </adjustedDate> [0..1]

'The date once the adjustment has been performed. (Note that this date may change if the business center holidays change)..'

<periodSkip> xsd:positiveInteger </periodSkip> [0..1]

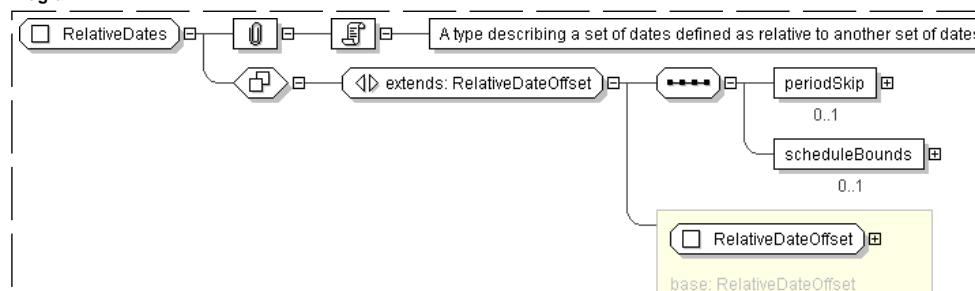
'The number of periods in the referenced date schedule that are between each date in the relative date schedule. Thus a skip of 2 would mean that dates are relative to every second date in the referenced schedule. If present this should have a value greater than 1.'

<scheduleBounds> DateRange </scheduleBounds> [0..1]

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

<...>

#### Diagram



#### Schema Component Representation

```

<xsd:complexType name="RelativeDates">
  <xsd:complexContent>
    <xsd:extension base=" RelativeDateOffset ">
      <xsd:sequence>
        <xsd:element name="periodSkip" type=" xsd:positiveInteger " minOccurs="0"/>
        <xsd:element name="scheduleBounds" type=" DateRange " minOccurs="0"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>

```

## Complex Type: RequiredIdentifierDate

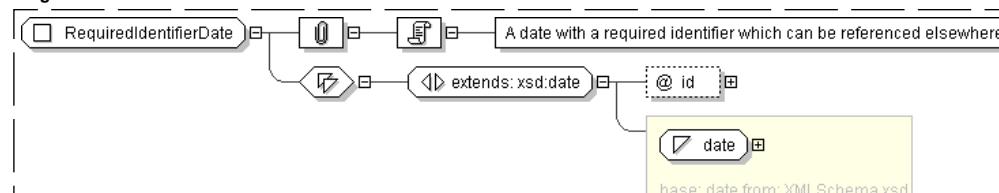
Super-types: [xsd:date < RequiredIdentifierDate](#) (by extension)  
 Sub-types: None

<b>Name</b>	RequiredIdentifierDate
<b>Abstract</b>	no
<b>Documentation</b>	A date with a required identifier which can be referenced elsewhere.

### XML Instance Representation

```
<...>
  id="xsd:ID [1]">
  xsd:date
</...>
```

### Diagram



### Schema Component Representation

```
<xsd:complexType name="RequiredIdentifierDate">
  <xsd:simpleContent>
    <xsd:extension base="xsd:date">
      <xsd:attribute name="id" type="xsd:ID" use="required"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
```

## Complex Type: ResetFrequency

Super-types: [Frequency < ResetFrequency](#) (by extension)  
 Sub-types: None

<b>Name</b>	ResetFrequency
<b>Abstract</b>	no
<b>Documentation</b>	A type defining the reset frequency. In the case of a weekly reset, also specifies the day of the week that the reset occurs. If the reset frequency is greater than the calculation period frequency this implies that more or more reset dates is established for each calculation period and some form of rate averaging is applicable. The specific averaging method of calculation is specified in FloatingRateCalculation. In case the reset frequency is of value T (Term), the period is defined by the swapSwapStreamcalculationPeriodDatesEffectiveDate and the swapSwapStreamcalculationPeriodDatesTerminationDate.

### XML Instance Representation

```
<...>
  id="xsd:ID [0..1]">
  <periodMultiplier> xsd:integer </periodMultiplier> [1]
  '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.'
```

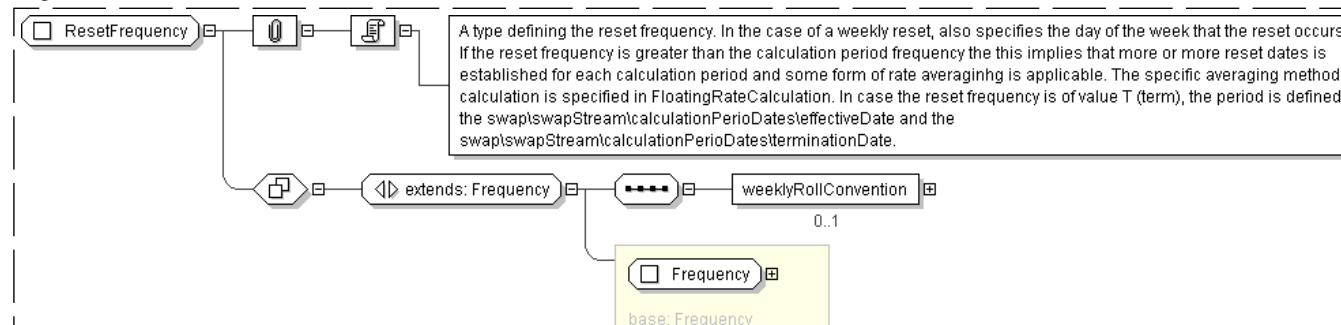
```
<period> PeriodExtendedEnum </period> [1]
```

'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).'

```
<weeklyRollConvention> WeeklyRollConventionEnum </weeklyRollConvention> [0..1]
```

'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.'

&lt;/...&gt;

**Diagram****Schema Component Representation**

```

<xsd:complexType name="ResetFrequency">
  <xsd:complexContent>
    <xsd:extension base=" Frequency ">
      <xsd:sequence>
        <xsd:element name="weeklyRollConvention" type=" WeeklyRollConventionEnum " minOccurs="0"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
  
```

top

**Complex Type: ReturnSwapNotionalAmountReference**

Super-types:

[Reference](#) < ReturnSwapNotionalAmountReference (by extension)

Sub-types:

None

**Name**

ReturnSwapNotionalAmountReference

**Abstract**

no

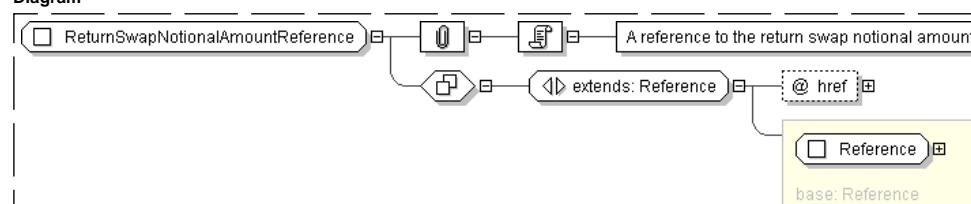
**Documentation**

A reference to the return swap notional amount.

**XML Instance Representation**

```

<...
  href=" xsd:IDREF [1]" />
  
```

**Diagram**

**Schema Component Representation**

```
<xsd:complexType name="ReturnSwapNotionalAmountReference">
  <xsd:complexContent>
    <xsd:extension base=" Reference ">
      <xsd:attribute name="href" type=" xsd:IDREF " use="required" reference="NotionalAmount" />
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

top

**Complex Type: Rounding**

Super-types:	None
Sub-types:	None

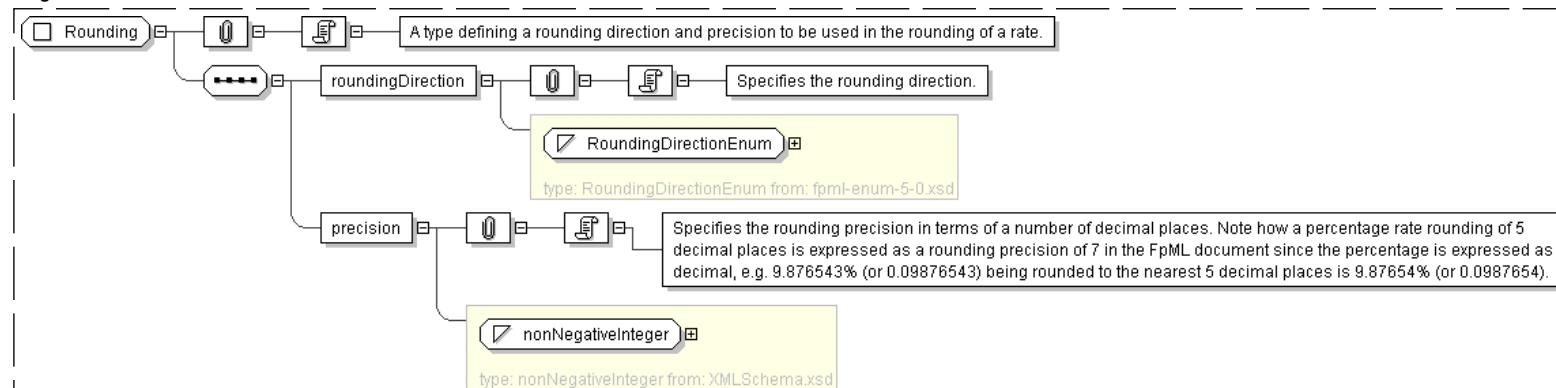
Name	Rounding
Used by (from the same schema document)	Complex Type <a href="#">FloatingRateCalculation</a>
Abstract	no
Documentation	A type defining a rounding direction and precision to be used in the rounding of a rate.

**XML Instance Representation**

```
<...>
  <roundingDirection> RoundingDirectionEnum </roundingDirection> [1]
    'Specifies the rounding direction.'

  <precision> xsd:nonNegativeInteger </precision> [1]
    '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).'

</...>
```

**Diagram****Schema Component Representation**

```
<xsd:complexType name="Rounding">
  <xsd:sequence>
    <xsd:element name="roundingDirection" type=" RoundingDirectionEnum " />
    <xsd:element name="precision" type=" xsd:nonNegativeInteger " />
  </xsd:sequence>
</xsd:complexType>
```

top

## Complex Type: Routing

<i>Super-types:</i>	None
<i>Sub-types:</i>	None
<b>Name</b>	Routing
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">SplitSettlement</a> , Complex Type <a href="#">SplitSettlement</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type that provides three alternative ways of identifying a party involved in the routing of a payment. The identification may use payment system identifiers only: actual name, address and other reference information; or a combination of both.

## XML Instance Representation

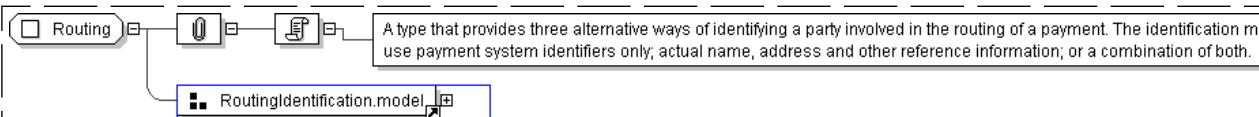
```
<...>
Start Choice [1]
  <routingIds> RoutingIds </routingIds> [1]
    'A set of unique identifiers for a party, each one identifying the party within a payment system. The assumption is that each party will not have more than one identifier within the same payment system.'

  <routingExplicitDetails> RoutingExplicitDetails </routingExplicitDetails> [1]
    'A set of details that is used to identify a party involved in the routing of a payment when the party does not have a code that identifies it within one of the recognized payment systems.'

  <routingIdsAndExplicitDetails> RoutingIdsAndExplicitDetails </routingIdsAndExplicitDetails> [1]
    'A combination of coded payment system identifiers and details for physical addressing for a party involved in the routing of a payment.'

End Choice
</...>
```

### Diagram



## Schema Component Representation

```
<xsd:complexType name="Routing">
  <xsd:group ref=" RoutingIdentification.model " />
</xsd:complexType>
```

top

#### Complex Type: RoutingExplicitDetails

<b>Super-types:</b>	None
<b>Sub-types:</b>	None
<b>Name</b>	RoutingExplicitDetails
<b>Used by (from the same schema document)</b>	Model Group <a href="#">RoutingIdentification.model</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type that models name, address and supplementary textual information for the purposes of identifying a party involved in the routing of a payment.

```
<...>
  <routingName> xsd:string </routingName> [1]
    'A real name that is used to identify a party involved in the routing of a payment'
```

```

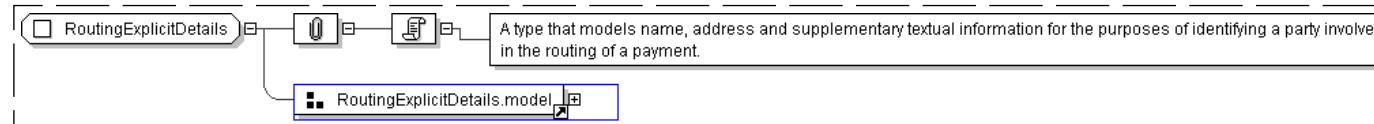
<routingAddress> Address </routingAddress> [0..1]
'A physical postal address via which a payment can be routed.'

<routingAccountNumber> xsd:string </routingAccountNumber> [0..1]
'An account number via which a payment can be routed.'

<routingReferenceText> xsd:string </routingReferenceText> [0..*]
'A piece of free-format text used to assist the identification of a party involved in
the routing of a payment.'

</...>

```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="RoutingExplicitDetails">
  <xsd:group ref=" RoutingExplicitDetails.model " />
</xsd:complexType>

```

top

**Complex Type: [RoutingId](#)**

<b>Super-types:</b>	<a href="#">xsd:normalizedString</a> < <a href="#">Scheme</a> (by restriction) < <b>RoutingId</b> (by extension)
<b>Sub-types:</b>	None

<b>Name</b>	RoutingId
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">RoutingIds</a>
<b>Abstract</b>	no

**XML Instance Representation**

```

<...
  routingIdCodeScheme=" xsd:anyURI [0..1]">
  Scheme
</...>

```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="RoutingId">
  <xsd:simpleContent>
    <xsd:extension base=" Scheme ">
      <xsd:attribute name="routingIdCodeScheme" type=" xsd:anyURI " default="http://www.fpml.org/
        ext/iso9362"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>

```

## Complex Type: RoutingIds

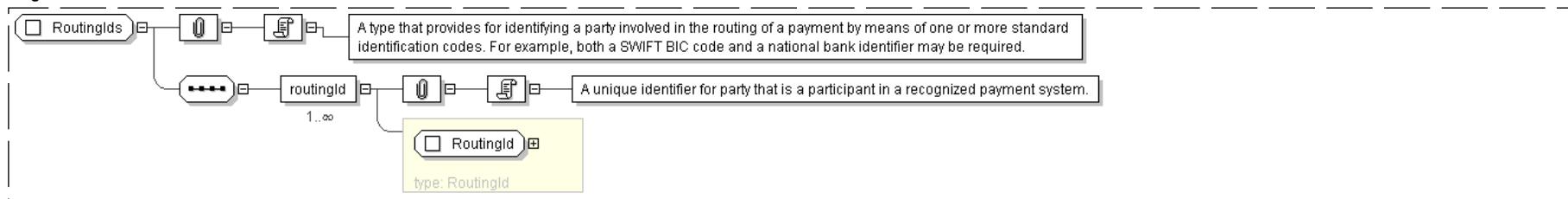
Super-types:	None
Sub-types:	None

Name	RoutingIds
Used by (from the same schema document)	Complex Type <a href="#">RoutingIdsAndExplicitDetails</a> , Model Group <a href="#">RoutingIdentification.model</a>
Abstract	no
Documentation	A type that provides for identifying a party involved in the routing of a payment by means of one or more standard identification codes. For example, both a SWIFT BIC code and a national bank identifier may be required.

### XML Instance Representation

```
<...>
<routingId> RoutingId </routingId> [1...*]
  'A unique identifier for party that is a participant in a recognized payment system.'
</...>
```

### Diagram



### Schema Component Representation

```
<xsd:complexType name="RoutingIds">
  <xsd:sequence>
    <xsd:element name="routingId" type=" RoutingId " maxOccurs="unbounded" />
  </xsd:sequence>
</xsd:complexType>
```

## Complex Type: RoutingIdsAndExplicitDetails

Super-types:	None
Sub-types:	None

Name	RoutingIdsAndExplicitDetails
Used by (from the same schema document)	Model Group <a href="#">RoutingIdentification.model</a>
Abstract	no
Documentation	A type that provides a combination of payment system identification codes with physical postal address details, for the purposes of identifying a party involved in the routing of a payment.

### XML Instance Representation

```
<...>
<routingIds> RoutingIds </routingIds> [1...*]
  'A set of unique identifiers for a party, eachone identifying the party within a payment system. The assumption is that each party will not have more than one identifier within the same payment system.'

<routingName> xsd:string </routingName> [1]
```

'A real name that is used to identify a party involved in the routing of a payment.'

<routingAddress> [Address](#) </routingAddress> [0..1]

'A physical postal address via which a payment can be routed.'

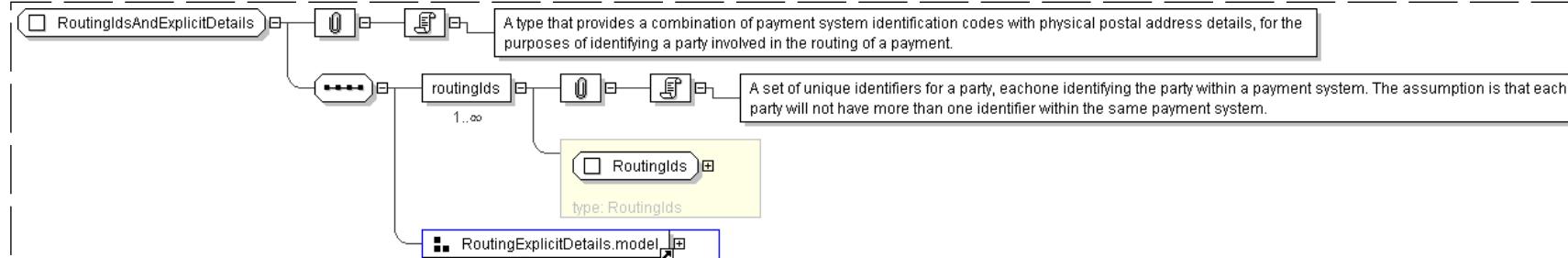
<routingAccountNumber> xsd:string </routingAccountNumber> [0..1]

'An account number via which a payment can be routed.'

<routingReferenceText> xsd:string </routingReferenceText> [0..\*]

'A piece of free-format text used to assist the identification of a party involved in the routing of a payment.'

&lt;/...&gt;

**Diagram****Schema Component Representation**

```

<xsd:complexType name="RoutingIdsAndExplicitDetails">
  <xsd:sequence>
    <xsd:element name="routingIds" type="<a href="#">RoutingIds</a>" maxOccurs="unbounded"/>
    <xsd:group ref="<a href="#">RoutingExplicitDetails.model</a>" />
  </xsd:sequence>
</xsd:complexType>

```

top

**Complex Type: Schedule**

Super-types:

None

Sub-types:

- [AmountSchedule](#) (by extension)
- [SpreadSchedule](#) (by extension)
- [StrikeSchedule](#) (by extension)

**Name**

Schedule

**Used by (from the same schema document)**Complex Type [ExerciseFeeSchedule](#), Complex Type [FloatingRate](#)**Abstract**

no

**Documentation**

A type defining a schedule of rates 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.

**XML Instance Representation**

```

<...
id="<a href="#">xsd:ID</a> [0..1]">
  <initialValue> xsd:decimal </initialValue> [1]

```

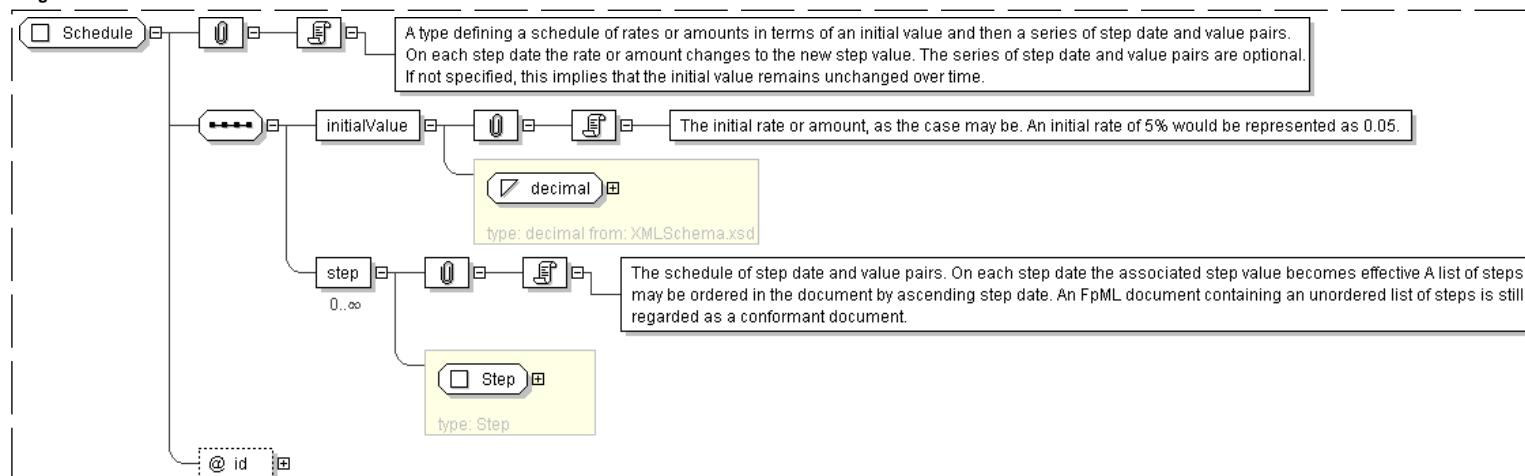
'The initial rate or amount, as the case may be. An initial rate of 5% would be represented as 0.05.'

<step> [Step](#) </step> [0..\*]

'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.'*

&lt;/...&gt;

**Diagram****Schema Component Representation**

```

<xsd:complexType name="Schedule">
  <xsd:sequence>
    <xsd:element name="initialValue" type="xsd:decimal" />
    <xsd:element name="step" type="Step" minOccurs="0" maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:attribute name="id" type="xsd:ID" />
</xsd:complexType>
  
```

top

**Complex Type: ScheduleReference**

**Super-types:** [Reference](#) < **ScheduleReference** (by extension)

**Sub-types:** None

<b>Name</b>	ScheduleReference
-------------	-------------------

<b>Used by (from the same schema document)</b>	Complex Type <a href="#">ExerciseFeeSchedule</a>
--	--

<b>Abstract</b>	no
-----------------	----

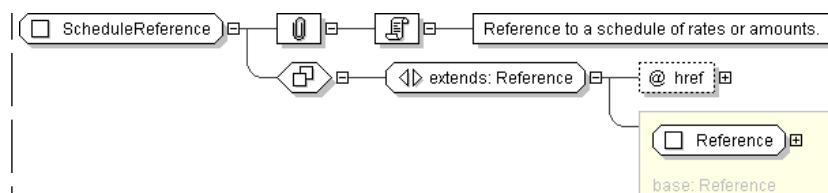
<b>Documentation</b>	Reference to a schedule of rates or amounts.
----------------------	--

**XML Instance Representation**

```

<...
  href="#xsd:IDREF [1]" />
  
```

**Diagram**

**Schema Component Representation**

```

<xsd:complexType name="ScheduleReference">
  <xsd:complexContent>
    <xsd:extension base=" Reference ">
      <xsd:attribute name="href" type=" xsd:IDREF " use="required" reference="Schedule"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
  
```

[top](#)**Complex Type: SettlementInformation**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None

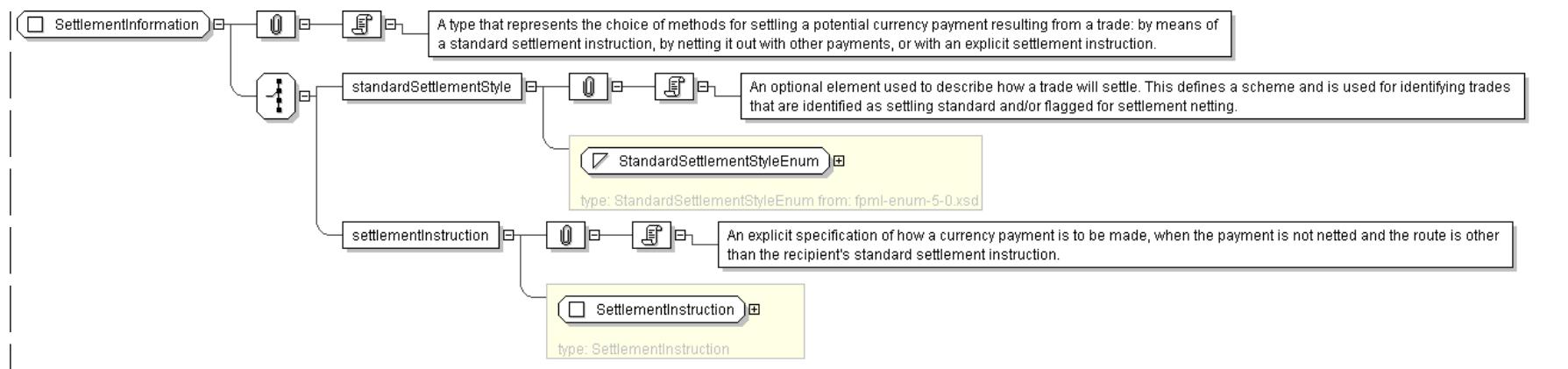
<b>Name</b>	SettlementInformation
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">Payment</a> , Complex Type <a href="#">PaymentDetails</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type that represents the choice of methods for settling a potential currency payment resulting from a trade: by means of a standard settlement instruction, by netting it out with other payments, or with an explicit settlement instruction.

**XML Instance Representation**

```

<...>
Start Choice [1]
  <standardSettlementStyle> StandardSettlementStyleEnum </standardSettlementStyle> [1]
  'An optional element used to describe how a trade will settle. This defines a scheme and
  is used for identifying trades that are identified as settling standard and/or flagged
  for settlement netting.'
<settlementInstruction> SettlementInstruction </settlementInstruction> [1]
  'An explicit specification of how a currency payment is to be made, when the payment is
  not netted and the route is other than the recipient\'s standard settlement instruction.'
End Choice
</...>
  
```

**Diagram**



#### Schema Component Representation

```

<xsd:complexType name="SettlementInformation">
  <xsd:choice>
    <xsd;element name="standardSettlementStyle" type=" StandardSettlementStyleEnum " />
    <xsd;element name="settlementInstruction" type=" SettlementInstruction " />
  </xsd:choice>
</xsd:complexType>
  
```

top

#### Complex Type: `SettlementInstruction`

Super-types:	None
Sub-types:	None

Name	SettlementInstruction
Used by (from the same schema document)	Complex Type <a href="#">SettlementInformation</a>
Abstract	no
Documentation	A type that models a complete instruction for settling a currency payment, including the settlement method to be used, the correspondent bank, any intermediary banks and the ultimate beneficiary.

#### XML Instance Representation

```

<...>
<settlementMethod> SettlementMethod </settlementMethod> [0..1]
  'The mechanism by which settlement is to be made. The scheme of domain values will
  include standard mechanisms such as CLS, Fedwire, Chips ABA, Chips UID, SWIFT, CHAPS and DDA.'
<correspondentInformation> CorrespondentInformation </correspondentInformation> [0..1]
  'The information required to identify the correspondent bank that will make delivery of
  the funds on the paying bank\'s behalf in the country where the payment is to be made'
<intermediaryInformation> IntermediaryInformation </intermediaryInformation> [0..*]
  'Information to identify an intermediary through which payment will be made by
  the correspondent bank to the ultimate beneficiary of the funds.'
<beneficiaryBank> Beneficiary </beneficiaryBank> [0..1]
  'The bank that acts for the ultimate beneficiary of the funds in receiving payments.'
<beneficiary> Beneficiary </beneficiary> [1]
  'The ultimate beneficiary of the funds. The beneficiary can be identified either by an
  account at the beneficiaryBank (qv) or by explicit routingInformation. This element
  provides for the latter.'
<depositoryPartyReference> PartyReference </depositoryPartyReference> [0..1]
  '
  
```

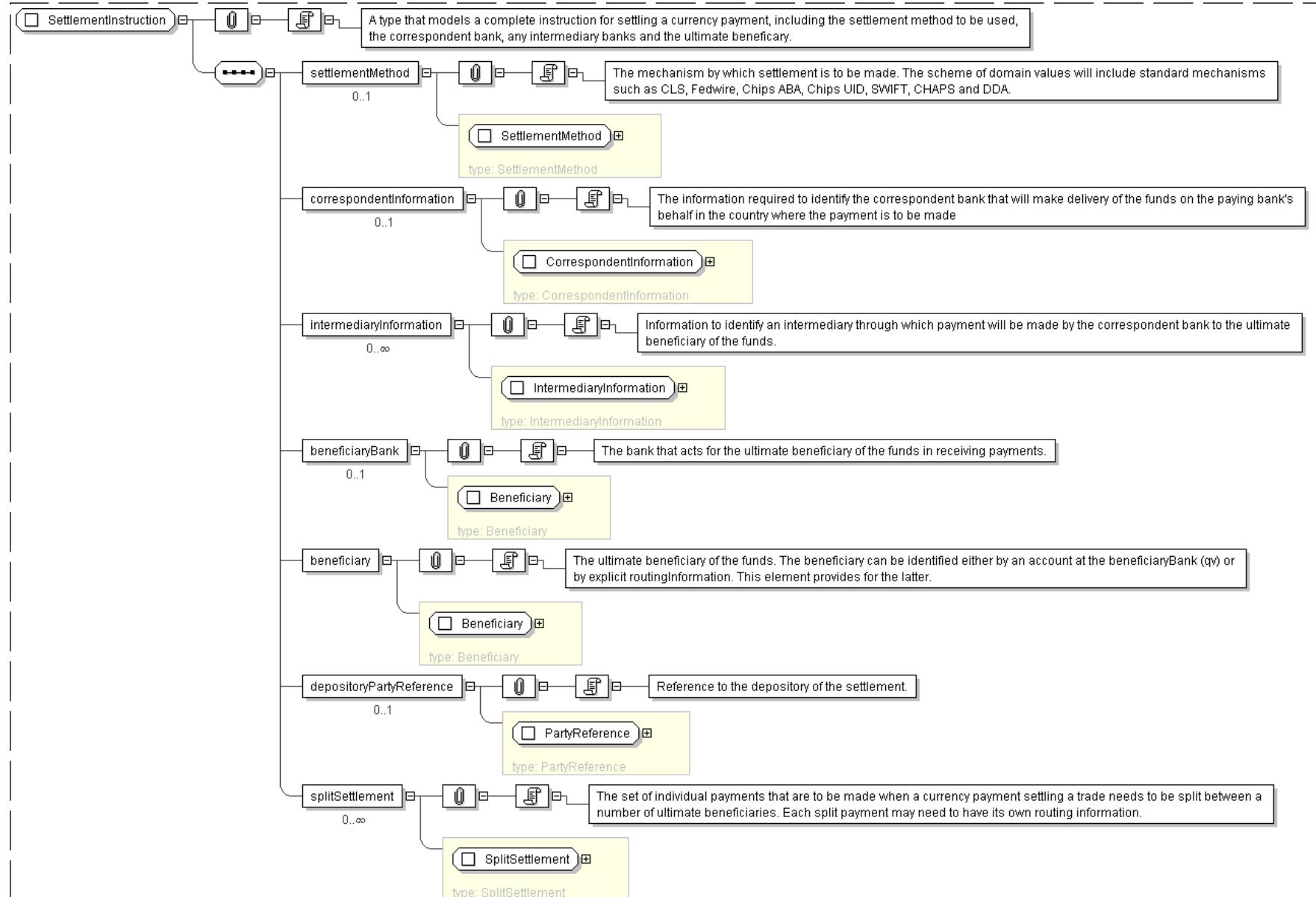
'Reference to the depository of the settlement.'

<splitSettlement> **SplitSettlement** </splitSettlement> [0..\*]

'The set of individual payments that are to be made when a currency payment settling a trade needs to be split between a number of ultimate beneficiaries. Each split payment may need to have its own routing information.'

</...>

#### Diagram



#### Schema Component Representation

```
<xsd:complexType name="SettlementInstruction">
```

```

<xsd:sequence>
  <xsd:element name="settlementMethod" type=" SettlementMethod " minOccurs="0" />
  <xsd:element name="correspondentInformation" type=" CorrespondentInformation " minOccurs="0" />
  <xsd:element name="intermediaryInformation" type=" IntermediaryInformation "
    minOccurs="0" maxOccurs="unbounded"/>
  <xsd:element name="beneficiaryBank" type=" Beneficiary " minOccurs="0" />
  <xsd:element name="beneficiary" type=" Beneficiary " />
  <xsd:element name="depositoryPartyReference" type=" PartyReference " minOccurs="0" />
  <xsd:element name="splitSettlement" type=" SplitSettlement "
    minOccurs="0" maxOccurs="unbounded"/>
</xsd:sequence>
</xsd:complexType>

```

[top](#)

## Complex Type: SettlementMethod

Super-types:	<a href="#">xsd:normalizedString</a> < <a href="#">Scheme</a> (by restriction) < <b>SettlementMethod</b> (by extension)
Sub-types:	None

Name	SettlementMethod
Used by (from the same schema document)	Complex Type <a href="#">SettlementInstruction</a>
Abstract	no

### XML Instance Representation

```

<...
  settlementMethodScheme=" xsd:anyURI [0..1]">
  Scheme
</...>

```

### Diagram



### Schema Component Representation

```

<xsd:complexType name="SettlementMethod">
  <xsd:simpleContent>
    <xsd:extension base=" Scheme ">
      <xsd:attribute name="settlementMethodScheme" type=" xsd:anyURI " default="http://www.fpml.
        org/coding-scheme/settlement-method"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>

```

[top](#)

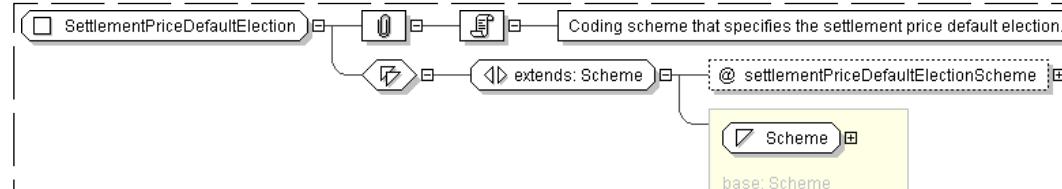
## Complex Type: SettlementPriceDefaultElection

Super-types:	<a href="#">xsd:normalizedString</a> < <a href="#">Scheme</a> (by restriction) < <b>SettlementPriceDefaultElection</b> (by extension)
Sub-types:	None

Name	SettlementPriceDefaultElection
Abstract	no
Documentation	Coding scheme that specifies the settlement price default election.

### XML Instance Representation

```
<...
  settlementPriceDefaultElectionScheme=" xsd:anyURI [0..1]">
  Scheme
</...>
```

**Diagram****Schema Component Representation**

```
<xsd:complexType name="SettlementPriceDefaultElection">
  <xsd:simpleContent>
    <xsd:extension base=" Scheme ">
      <xsd:attribute name="settlementPriceDefaultElectionScheme" type=" xsd:anyURI "
        default="http://www.fpml.org/coding-scheme/settlement-price-default-election"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
```

top

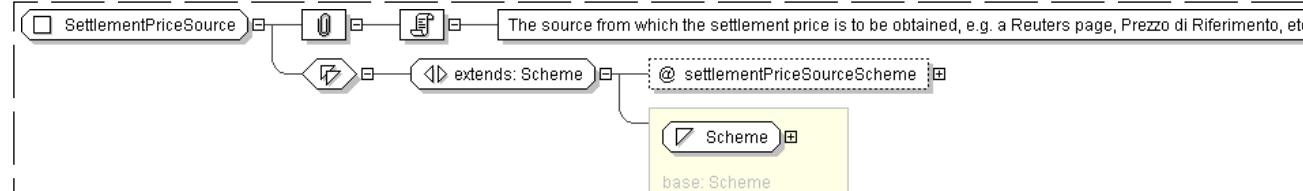
**Complex Type: SettlementPriceSource**

**Super-types:** xsd:normalizedString < [Scheme](#) (by restriction) < **SettlementPriceSource** (by extension)  
**Sub-types:** None

<b>Name</b>	SettlementPriceSource
<b>Abstract</b>	no
<b>Documentation</b>	The source from which the settlement price is to be obtained, e.g. a Reuters page, Prezzo di Riferimento, etc.

**XML Instance Representation**

```
<...
  settlementPriceSourceScheme=" xsd:anyURI [0..1]">
  Scheme
</...>
```

**Diagram****Schema Component Representation**

```
<xsd:complexType name="SettlementPriceSource">
  <xsd:simpleContent>
    <xsd:extension base=" Scheme ">
      <xsd:attribute name="settlementPriceSourceScheme" type=" xsd:anyURI "
        default="http://www.
          fpml.org/coding-scheme/settlement-price-source"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
```

**Complex Type: SettlementRateSource**

**Super-types:** None  
**Sub-types:** None

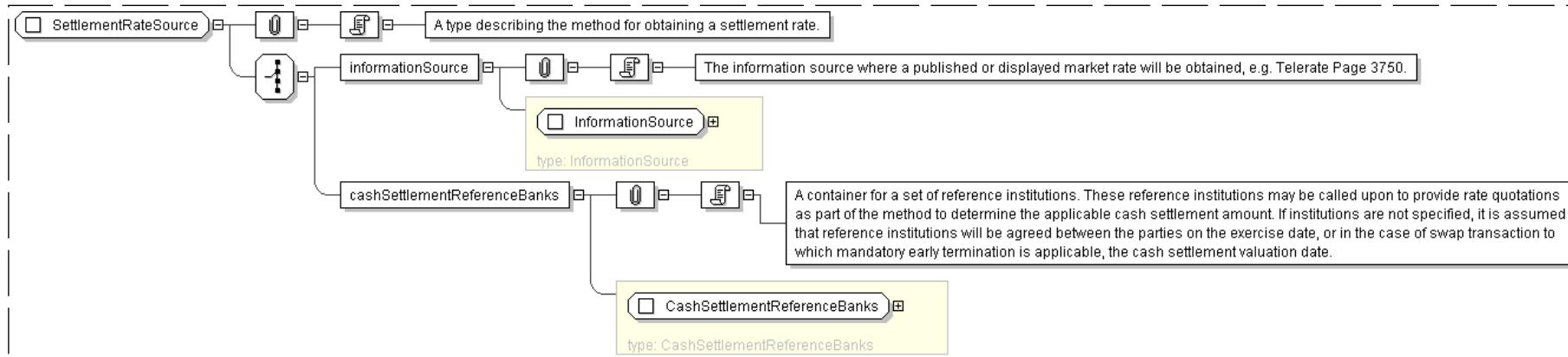
<b>Name</b>	SettlementRateSource
<b>Abstract</b>	no
<b>Documentation</b>	A type describing the method for obtaining a settlement rate.

**XML Instance Representation**

```
<...>
Start Choice [1]
<informationSource> InformationSource </informationSource> [1]
'The information source where a published or displayed market rate will be obtained, e.
g. Telerate Page 3750.'

<cashSettlementReferenceBanks> CashSettlementReferenceBanks </cashSettlementReferenceBanks> [1]
'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. If institutions are not specified, it is assumed that
reference institutions will be agreed between the parties on the exercise date, or in the
case of swap transaction to which mandatory early termination is applicable, the
cash settlement valuation date.'

End Choice
</...>
```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="SettlementRateSource">
  <xsd:choice>
    <xsd:element name="informationSource" type="InformationSource" />
    <xsd:element name="cashSettlementReferenceBanks" type="CashSettlementReferenceBanks" />
  </xsd:choice>
</xsd:complexType>
  
```

**Complex Type: SharedAmericanExercise**

<b>Super-types:</b>	<a href="#">Exercise</a> < <b>SharedAmericanExercise</b> (by extension)
<b>Sub-types:</b>	None
<b>Name</b>	SharedAmericanExercise
<b>Abstract</b>	no
<b>Documentation</b>	TBA

## XML Instance Representation

```
<...>
  id="xsd:ID [0..1]">
    <commencementDate> AdjustableOrRelativeDate </commencementDate> [1]
    'The first day of the exercise period for an American style option.'

    <expirationDate> AdjustableOrRelativeDate </expirationDate> [1]
    'The last day within an exercise period for an American style option. For a European
    style option it is the only day within the exercise period.'

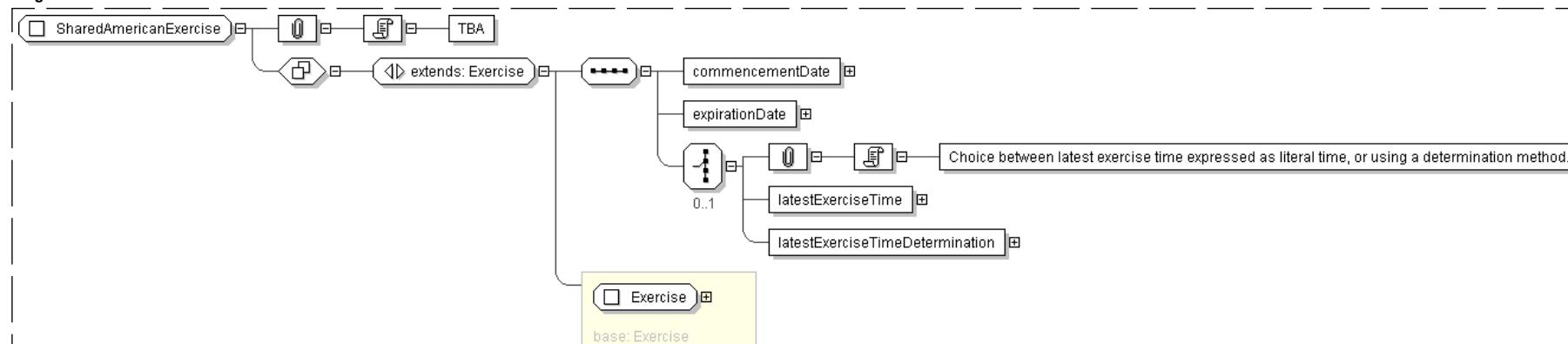
  Start Choice [0..1]
  'Choice between latest exercise time expressed as literal time, or using a
  determination method.'

    <latestExerciseTime> BusinessCenterTime </latestExerciseTime> [1]
    'For a Bermuda or American style option, the latest time on an exercise business day
    (excluding the expiration date) within the exercise period that notice can be given
    by the buyer to the seller or seller\'s agent. Notice of exercise given after this time
    is deemed to have been given on the next exercise business day.'

    <latestExerciseTimeDetermination> DeterminationMethod </latestExerciseTimeDetermination>
    'Latest exercise time determination method.'

  End Choice
</...>
```

## Diagram



## Schema Component Representation

```
<xsd:complexType name="SharedAmericanExercise">
  <xsd:complexContent>
    <xsd:extension base=" Exercise ">
      <xsd:sequence>
        <xsd:element name="commencementDate" type=" AdjustableOrRelativeDate " />
        <xsd:element name="expirationDate" type=" AdjustableOrRelativeDate " />
        <xsd:choice minOccurs="0">
          <xsd:element name="latestExerciseTime" type=" BusinessCenterTime " />
          <xsd:element name="latestExerciseTimeDetermination" type=" DeterminationMethod " />
        </xsd:choice>
```

```

</xsd:sequence>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>

```

## Complex Type: SimplePayment

**Super-types:** [PaymentBase](#) < **SimplePayment** (by extension)

**Sub-types:** None

<b>Name</b>	SimplePayment
-------------	---------------

<b>Abstract</b>	no
-----------------	----

<b>Documentation</b>	A complex type to specified payments in a simpler fashion than the Payment type. This construct should be used from the version 4.3 onwards.
----------------------	--

### XML Instance Representation

```

<...
  id="# ID [0..1]">
  <payerPartyReference> PartyReference </payerPartyReference> [1]
  'A reference to the party responsible for making the payments defined by this structure.'

  <payerAccountReference> AccountReference </payerAccountReference> [0..1]
  'A reference to the account responsible for making the payments defined by this structure.'

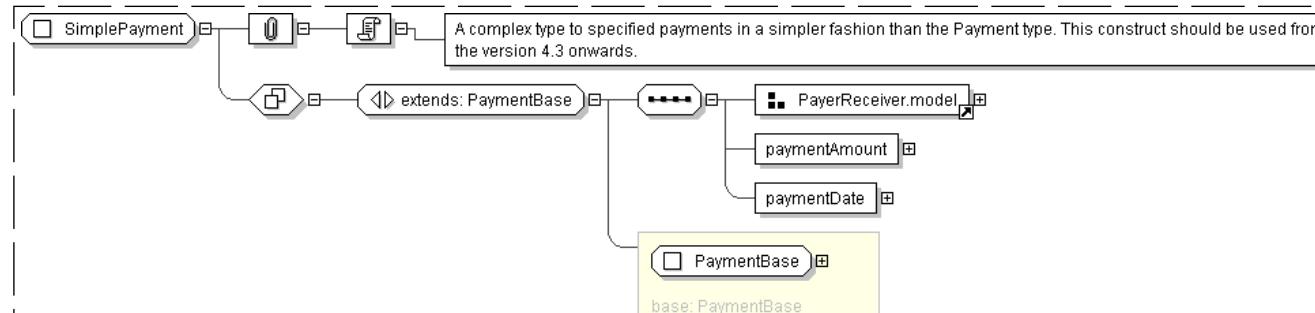
  <receiverPartyReference> PartyReference </receiverPartyReference> [1]
  'A reference to the party that receives the payments corresponding to this structure.'

  <receiverAccountReference> AccountReference </receiverAccountReference> [0..1]
  'A reference to the account that receives the payments corresponding to this structure.'

  <paymentAmount> Money </paymentAmount> [1]
  <paymentDate> AdjustableOrRelativeDate </paymentDate> [1]
  'The payment date. This date is subject to adjustment in accordance with any
  applicable business day convention.'

```

### Diagram



### Schema Component Representation

```

<xsd:complexType name="SimplePayment">
  <xsd:complexContent>
    <xsd:extension base=" PaymentBase ">
      <xsd:sequence>
        <xsd:group ref=" PayerReceiver.model "/>
        <xsd:element name="paymentAmount" type=" Money "/>
        <xsd:element name="paymentDate" type=" AdjustableOrRelativeDate "/>
      
```

```

</xsd:sequence>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>

```

## Complex Type: SplitSettlement

Super-types:	None
Sub-types:	None

Name	SplitSettlement
Used by (from the same schema document)	Complex Type <a href="#">SettlementInstruction</a>
Abstract	no
Documentation	A type that supports the division of a gross settlement amount into a number of split settlements, each requiring its own settlement instruction.

### XML Instance Representation

```

<...>
<splitSettlementAmount> Money </splitSettlementAmount> [1]
'One of the monetary amounts in a split settlement payment.'

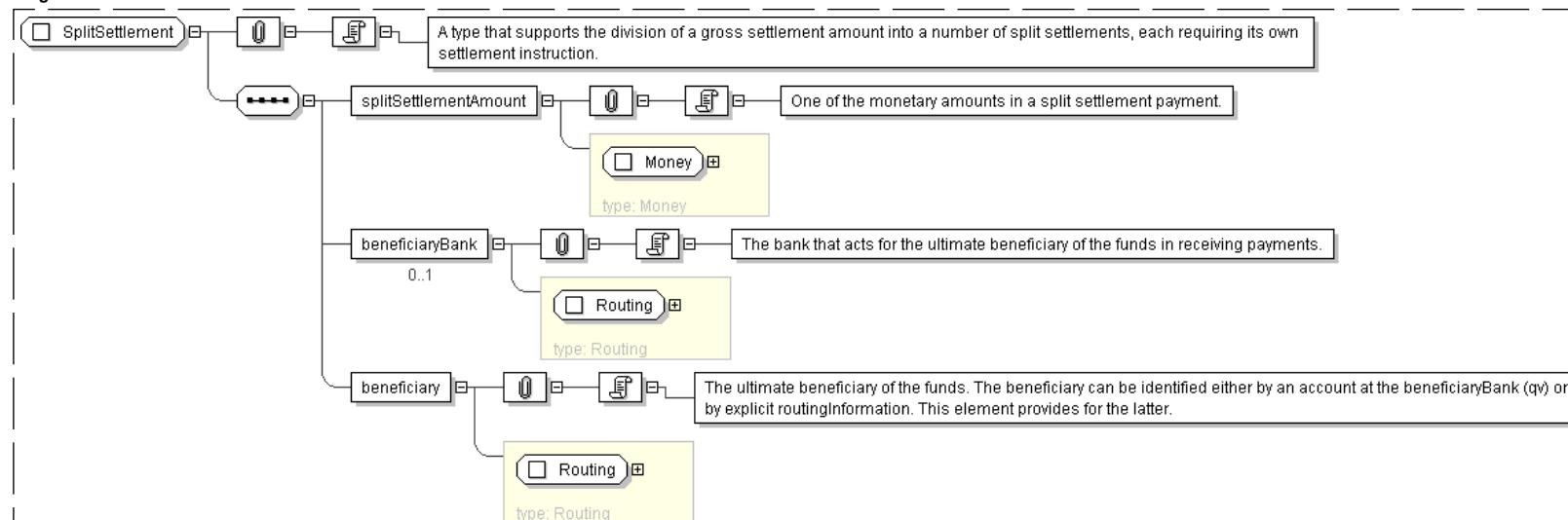
<beneficiaryBank> Routing </beneficiaryBank> [0..1]
'The bank that acts for the ultimate beneficiary of the funds in receiving payments.'

<beneficiary> Routing </beneficiary> [1]
'The ultimate beneficiary of the funds. The beneficiary can be identified either by an
account at the beneficiaryBank (qv) or by explicit routingInformation. This element
provides for the latter.'

</...>

```

### Diagram



### Schema Component Representation

```

<xsd:complexType name="SplitSettlement">
<xsd:sequence>
  <xsd:element name="splitSettlementAmount" type="Money" />
  <xsd:element name="beneficiaryBank" type="Routing" minOccurs="0" />

```

```

<xsd:element name="beneficiary" type=" Routing " />
</xsd:sequence>
</xsd:complexType>

```

## Complex Type: SpreadSchedule

Super-types:	<a href="#">Schedule</a> < <b>SpreadSchedule</b> (by extension)
Sub-types:	None

Name	SpreadSchedule
Used by (from the same schema document)	Complex Type <a href="#">FloatingRate</a>
Abstract	no
Documentation	Adds an optional spread type element to the Schedule to identify a long or short spread value.

### XML Instance Representation

```

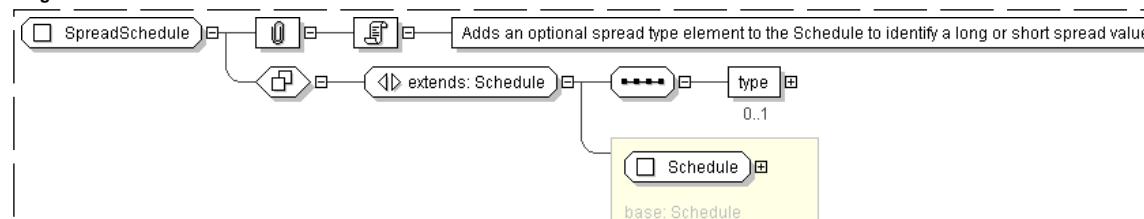
<...>
  id=" xsd:ID [0..1]">
    <initialValue> xsd:decimal </initialValue> [1]
    'The initial rate or amount, as the case may be. An initial rate of 5% would be represented
    as 0.05.'

    <step> Step </step> [0..*]
    '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.'

    <type> SpreadScheduleType </type> [0..1]
</...>

```

### Diagram



### Schema Component Representation

```

<xsd:complexType name="SpreadSchedule">
  <xsd:complexContent>
    <xsd:extension base=" Schedule ">
      <xsd:sequence>
        <xsd:element name="type" type=" SpreadScheduleType " minOccurs="0" />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

## Complex Type: SpreadScheduleReference

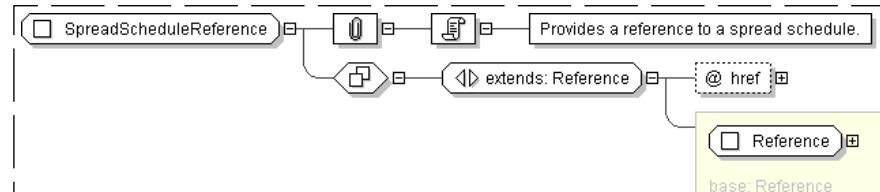
Super-types:	<a href="#">Reference</a> < <b>SpreadScheduleReference</b> (by extension)
Sub-types:	None

Name	SpreadScheduleReference
Abstract	no
Documentation	Provides a reference to a spread schedule.

## XML Instance Representation

```
<...>
  href=" xsd:IDREF [1]" />
```

## Diagram



## Schema Component Representation

```

<xsd:complexType name="SpreadScheduleReference">
  <xsd:complexContent>
    <xsd:extension base=" Reference ">
      <xsd:attribute name="href" type=" xsd:IDREF " use="required" reference="SpreadSchedule" />
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
  
```

top

## Complex Type: SpreadScheduleType

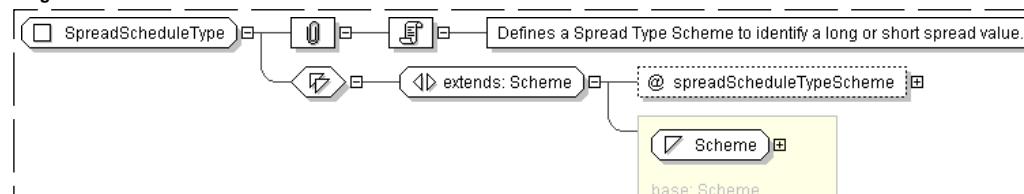
Super-types:	xsd:normalizedString < <a href="#">Scheme</a> (by restriction) < <b>SpreadScheduleType</b> (by extension)
Sub-types:	None

Name	SpreadScheduleType
Used by (from the same schema document)	Complex Type <a href="#">SpreadSchedule</a>
Abstract	no
Documentation	Defines a Spread Type Scheme to identify a long or short spread value.

## XML Instance Representation

```
<...>
  spreadScheduleTypeScheme=" xsd:anyURI [0..1]">
  Scheme
</...>
```

## Diagram



## Schema Component Representation

```

<xsd:complexType name="SpreadScheduleType">
  <xsd:simpleContent>
    <xsd:extension base=" Scheme ">
      <xsd:attribute name="spreadScheduleTypeScheme" type=" xsd:anyURI " default="http://www.fpml.
  
```

```

    org/coding-scheme/spread-schedule-type" />
  </xsd:extension>
</xsd:simpleContent>
</xsd:complexType>

```

top

## Complex Type: Step

**Super-types:** [StepBase](#) < **Step** (by extension)

**Sub-types:** None

**Name** Step

**Used by (from the same schema document)** Complex Type [Schedule](#)

**Abstract** no

**Documentation** A type defining a step date and step value pair. This step definitions are used to define varying rate or amount schedules, e.g. a notional amortization or a step-up coupon schedule.

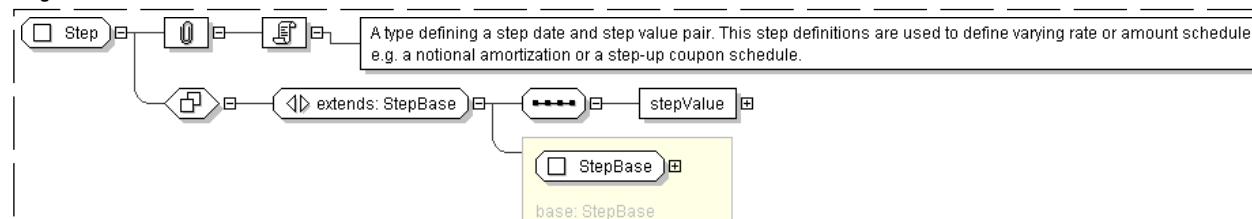
### XML Instance Representation

```

<...>
  id=" xsd:ID [0..1]">
    <stepDate> xsd:date </stepDate> [1]
      'The date on which the associated stepValue becomes effective. This day may be subject
      to adjustment in accordance with a business day convention.'
    <stepValue> xsd:decimal </stepValue> [1]
      'The rate or amount which becomes effective on the associated stepDate. A rate of 5% would
      be represented as 0.05.'
  </...>

```

### Diagram



### Schema Component Representation

```

<xsd:complexType name="Step">
  <xsd:complexContent>
    <xsd:extension base=" StepBase ">
      <xsd:sequence>
        <xsd:element name="stepValue" type=" xsd:decimal " />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

top

## Complex Type: StepBase

**Super-types:** None

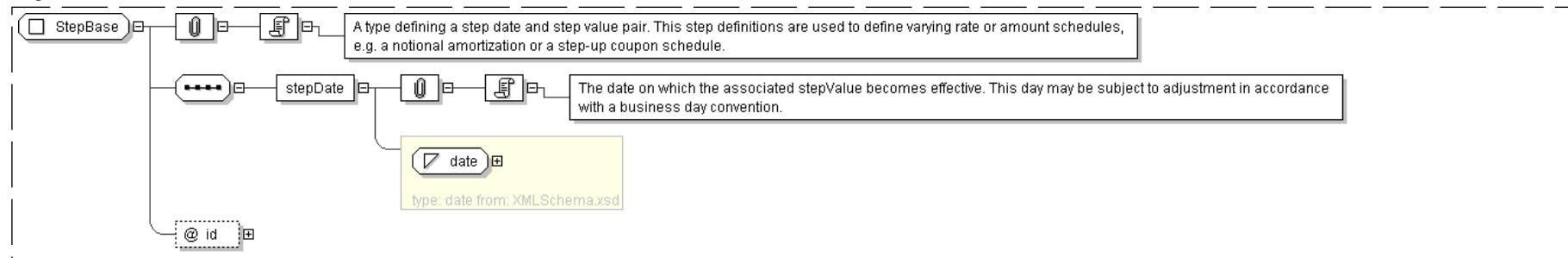
**Sub-types:**

- [NonNegativeStep](#) (by extension)
- [PositiveStep](#) (by extension)
- [Step](#) (by extension)

<b>Name</b>	StepBase
<b>Abstract</b>	yes
<b>Documentation</b>	A type defining a step date and step value pair. This step definitions are used to define varying rate or amount schedules, e.g. a notional amortization or a step-up coupon schedule.

**XML Instance Representation**

```
<...>
<id=" xsd:ID [0..1]">
<stepDate> xsd:date </stepDate> [1]
' The date on which the associated stepValue becomes effective. This day may be subject
to adjustment in accordance with a business day convention.'
</...>
```

**Diagram****Schema Component Representation**

```
<xsd:complexType name="StepBase" abstract="true">
  <xsd:sequence>
    <xsd:element name="stepDate" type=" xsd:date " />
  </xsd:sequence>
  <xsd:attribute name="id" type=" xsd:ID " />
</xsd:complexType>
```

top

**Complex Type: StreetAddress**

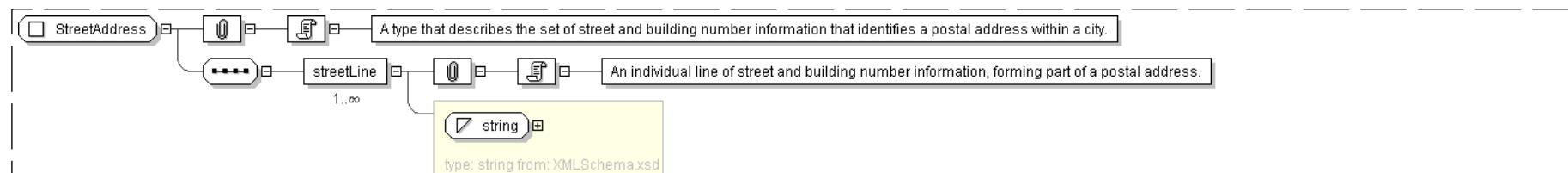
<b>Super-types:</b>	None
<b>Sub-types:</b>	None

<b>Name</b>	StreetAddress
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">Address</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type that describes the set of street and building number information that identifies a postal address within a city.

**XML Instance Representation**

```
<...>
<streetLine> xsd:string </streetLine> [1...*]
' An individual line of street and building number information, forming part of a
postal address.'
</...>
```

**Diagram**

**Schema Component Representation**

```

<xsd:complexType name="StreetAddress">
  <xsd:sequence>
    <xsd:element name="streetLine" type="xsd:string" maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>
  
```

[top](#)**Complex Type: Strike**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None

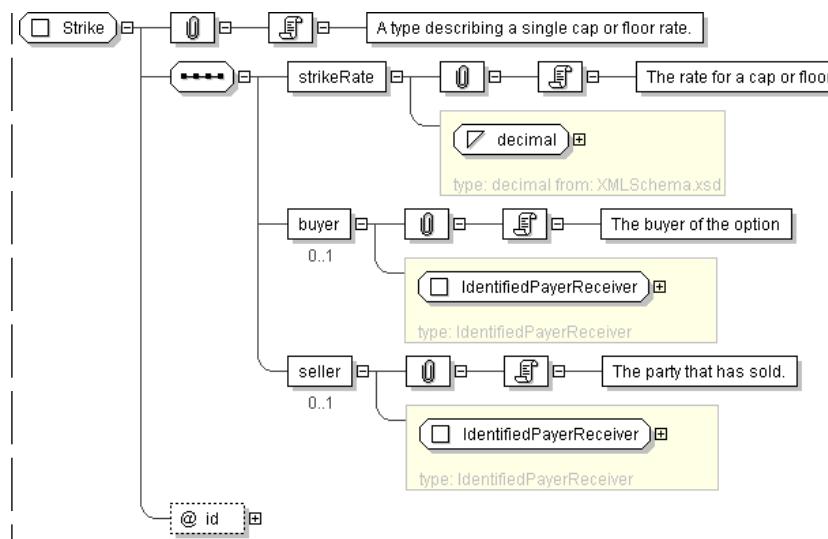
<b>Name</b>	Strike
<b>Abstract</b>	no
<b>Documentation</b>	A type describing a single cap or floor rate.

**XML Instance Representation**

```

<...
  id="xsd:ID [0..1]">
  <strikeRate> xsd:decimal </strikeRate> [1]
  'The rate for a cap or floor.'
  <buyer> IdentifiedPayerReceiver </buyer> [0..1]
  'The buyer of the option'
  <seller> IdentifiedPayerReceiver </seller> [0..1]
  'The party that has sold.'
</...>
  
```

**Diagram**



#### Schema Component Representation

```

<xsd:complexType name="Strike">
  <xsd:sequence>
    <xsd:element name="strikeRate" type="xsd:decimal" />
    <xsd:element name="buyer" type="IdentifiedPayerReceiver" minOccurs="0"/>
    <xsd:element name="seller" type="IdentifiedPayerReceiver" minOccurs="0"/>
  </xsd:sequence>
  <xsd:attribute name="id" type="xsd:ID" />
</xsd:complexType>
  
```

top

#### Complex Type: StrikeSchedule

Super-types:	<a href="#">Schedule</a> < <b>StrikeSchedule</b> (by extension)
Sub-types:	None

Name	StrikeSchedule
Used by (from the same schema document)	Complex Type <a href="#">FloatingRate</a> , Complex Type <a href="#">FloatingRate</a>
Abstract	no
Documentation	A type describing a schedule of cap or floor rates.

#### XML Instance Representation

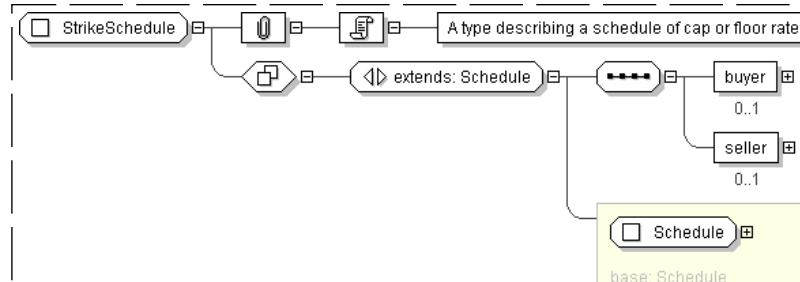
```

<...
  id="xsd:ID [0..1]">
  <initialValue> xsd:decimal </initialValue> [1]
  'The initial rate or amount, as the case may be. An initial rate of 5% would be represented
  as 0.05.'
<step> Step </step> [0..*]
  '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.'
<buyer> IdentifiedPayerReceiver </buyer> [0..1]
  'The buyer of the option'
  
```

```
|   <seller> IdentifiedPayerReceiver </seller> [0..1]
```

'The party that has sold.'

<...>

**Diagram****Schema Component Representation**

```

<xsd:complexType name="StrikeSchedule">
  <xsd:complexContent>
    <xsd:extension base=" Schedule ">
      <xsd:sequence>
        <xsd:element name="buyer" type=" IdentifiedPayerReceiver" minOccurs="0"/>
        <xsd:element name="seller" type=" IdentifiedPayerReceiver" minOccurs="0"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
  
```

top

**Complex Type: Stub**

**Super-types:** [StubValue](#) < **Stub** (by extension)

**Sub-types:** None

**Name**

Stub

**Abstract**

no

**Documentation**

A type defining how a stub calculation period amount is calculated and the start and end date of the stub. 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 many 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.

**XML Instance Representation**

```

<...>
Start Choice [1]
  <floatingRate> FloatingRate </floatingRate> [1..2]
  
```

'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.'

```
<stubRate> xsd:decimal </stubRate> [1]
```

'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.'

<stubAmount> Money </stubAmount> [1]

'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.'

End Choice

<stubStartDate> AdjustableOrRelativeDate </stubStartDate> [0..1]

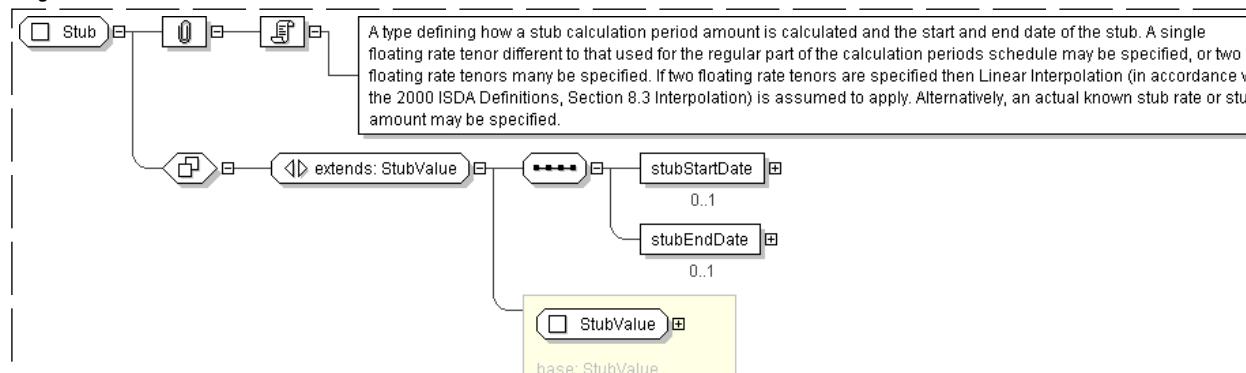
'Start date of stub period. This was created to support use of the InterestRateStream within the Equity Derivative sphere, and this element is not expected to be produced in the representation of Interest Rate products.'

<stubEndDate> AdjustableOrRelativeDate </stubEndDate> [0..1]

'End date of stub period. This was created to support use of the InterestRateStream within the Equity Derivative sphere, and this element is not expected to be produced in the representation of Interest Rate products.'

</...>

#### Diagram



#### Schema Component Representation

```

<xsd:complexType name="Stub">
  <xsd:complexContent>
    <xsd:extension base=" StubValue ">
      <xsd:sequence>
        <xsd:element name="stubStartDate" type=" AdjustableOrRelativeDate " minOccurs="0"/>
        <xsd:element name="stubEndDate" type=" AdjustableOrRelativeDate " minOccurs="0"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
  
```

[top](#)

#### Complex Type: StubValue

Super-types:

None

Sub-types:

- [Stub](#) (by extension)

Name

StubValue

Abstract

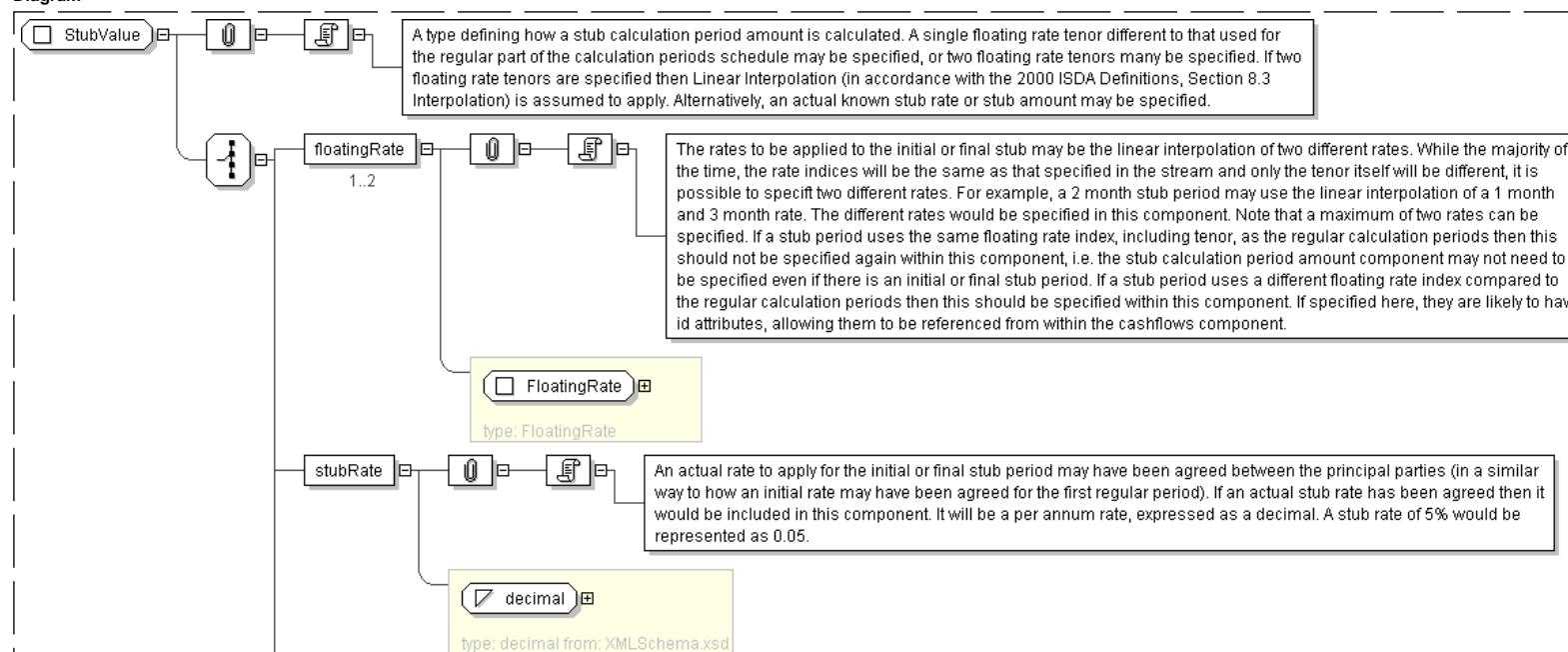
no

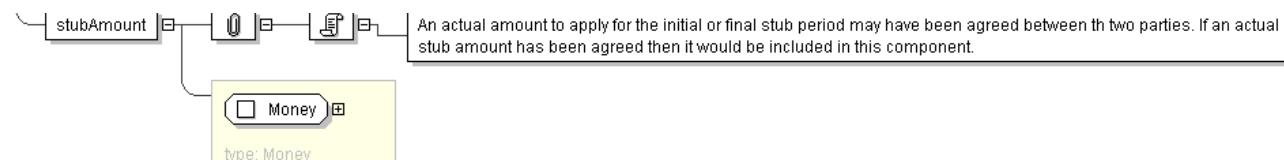
**Documentation**

A type 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.

**XML Instance Representation**

```
<...>
Start Choice [1]
<floatingRate> FloatingRate </floatingRate> [1..2]
'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.'
<stubRate> xsd:decimal </stubRate> [1]
'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.'
<stubAmount> Money </stubAmount> [1]
'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.'
End Choice
</...>
```

**Diagram**

**Schema Component Representation**

```
<xsd:complexType name="StubValue">
  <xsd:choice>
    <xsd:element name="floatingRate" type="#FloatingRate" maxOccurs="2"/>
    <xsd:element name="stubRate" type="xsd:decimal"/>
    <xsd:element name="stubAmount" type="Money"/>
  </xsd:choice>
</xsd:complexType>
```

[top](#)**Complex Type: TimezoneLocation**

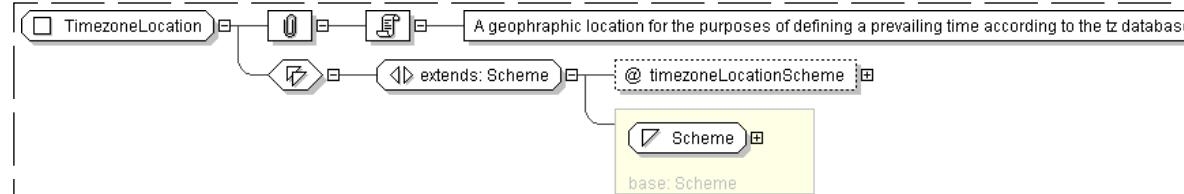
**Super-types:** [xsd:normalizedString](#) < [Scheme](#) (by restriction) < **TimezoneLocation** (by extension)

**Sub-types:** None

<b>Name</b>	TimezoneLocation
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">PrevailingTime</a>
<b>Abstract</b>	no
<b>Documentation</b>	A geographic location for the purposes of defining a prevailing time according to the tz database.

**XML Instance Representation**

```
<...
  timezoneLocationScheme="xsd:anyURI [0..1]">
  Scheme
</...>
```

**Diagram****Schema Component Representation**

```
<xsd:complexType name="TimezoneLocation">
  <xsd:simpleContent>
    <xsd:extension base="Scheme">
      <xsd:attribute name="timezoneLocationScheme" type="xsd:anyURI" default="http://www.fpml.org/coding-scheme/external/tzdatabase"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
```

[top](#)**Model Group: AdjustableDate.model**

<b>Name</b>	AdjustableDate.model
-------------	----------------------

## Used by (from the same schema document)

Complex Type [AdjustableDate](#), Complex Type [AdjustableOrAdjustedDate](#)

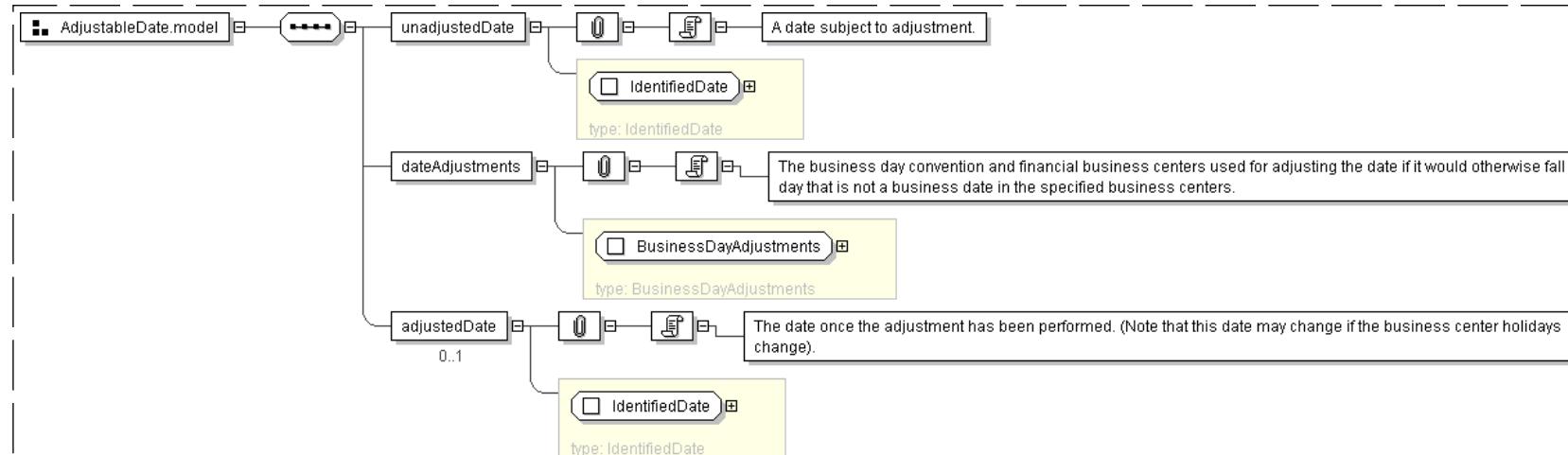
## XML Instance Representation

```
<unadjustedDate> IdentifiedDate </unadjustedDate> [1]
'A date subject to adjustment.'
```

<dateAdjustments> [BusinessDayAdjustments](#) </dateAdjustments> [1]
'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 date in the specified business centers.'

```
<adjustedDate> IdentifiedDate </adjustedDate> [0..1]
'The date once the adjustment has been performed. (Note that this date may change if the business center holidays change).'
```

## Diagram



## Schema Component Representation

```
<xsd:group name="AdjustableDate.model">
  <xsd:sequence>
    <xsd:element name="unadjustedDate" type=" IdentifiedDate " />
    <xsd:element name="dateAdjustments" type=" BusinessDayAdjustments " />
    <xsd:element name="adjustedDate" type=" IdentifiedDate " minOccurs="0"/>
  </xsd:sequence>
</xsd:group>
```

top

Model Group: [BusinessCentersOrReference.model](#)

## Name

BusinessCentersOrReference.model

## Used by (from the same schema document)

Complex Type [BusinessDateRange](#), Complex Type [BusinessDayAdjustments](#), Complex Type [RelativeDateOffset](#), Complex Type [RelativeDateSequence](#)

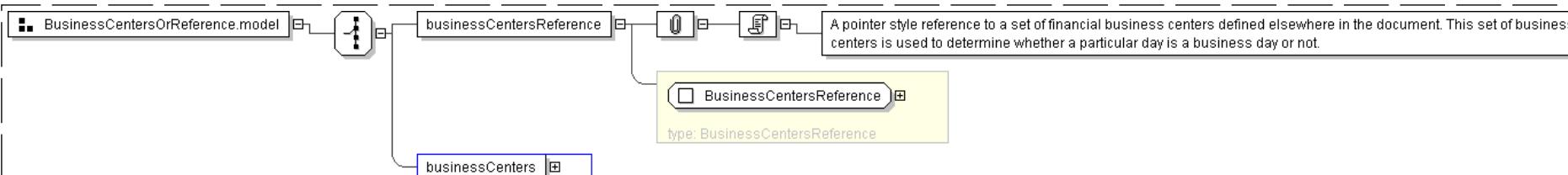
## XML Instance Representation

```
Start Choice [1]
<businessCentersReference> BusinessCentersReference </businessCentersReference> [1]
'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.'
```

```

<businessCenters> BusinessCenters </businessCenters> [1]
End Choice

```

**Diagram****Schema Component Representation**

```

<xsd:group name="BusinessCentersOrReference.model">
  <xsd:choice>
    <xsd:element name="businessCentersReference" type=" BusinessCentersReference "/>
    <xsd:element name="businessCenters" type=" BusinessCenters "/>
  </xsd:choice>
</xsd:group>

```

top

**Model Group: BuyerSeller.model****Name**

BuyerSeller.model

**XML Instance Representation**

```
<buyerPartyReference> PartyReference </buyerPartyReference> [1]
```

'A reference to the party that buys this instrument, i.e. pays for this instrument and receives the rights defined by it. See 2000 ISDA definitions Article 11.1 (b). In the case of FRAs this is the fixed rate payer.'

```
<buyerAccountReference> AccountReference </buyerAccountReference> [0..1]
```

'A reference to the account that buys this instrument.'

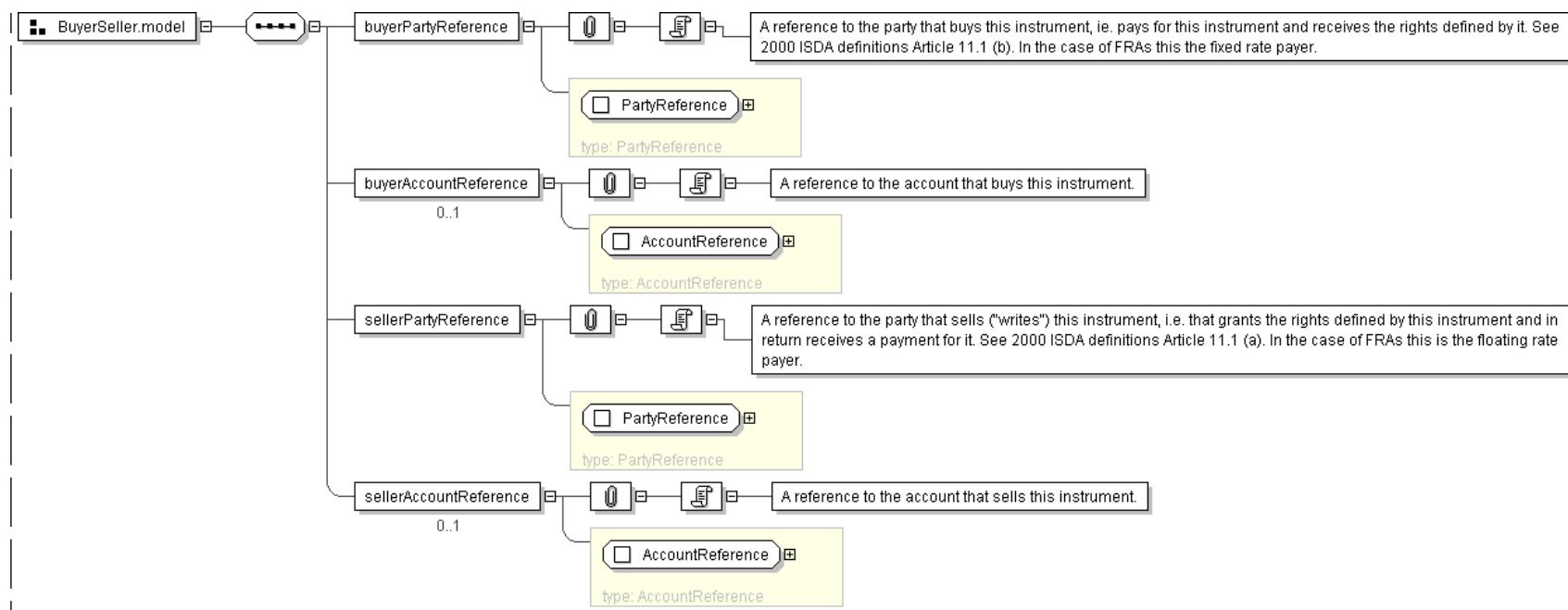
```
<sellerPartyReference> PartyReference </sellerPartyReference> [1]
```

'A reference to the party that sells ("writes") this instrument, i.e. that grants the rights defined by this instrument and in return receives a payment for it. See 2000 ISDA definitions Article 11.1 (a). In the case of FRAs this is the floating rate payer.'

```
<sellerAccountReference> AccountReference </sellerAccountReference> [0..1]
```

'A reference to the account that sells this instrument.'

**Diagram**

**Schema Component Representation**

```
<xsd:group name="BuyerSeller.model">
  <xsd:sequence>
    <xsd:element name="buyerPartyReference" type=" PartyReference "/>
    <xsd:element name="buyerAccountReference" type=" AccountReference " minOccurs="0"/>
    <xsd:element name="sellerPartyReference" type=" PartyReference "/>
    <xsd:element name="sellerAccountReference" type=" AccountReference " minOccurs="0"/>
  </xsd:sequence>
</xsd:group>
```

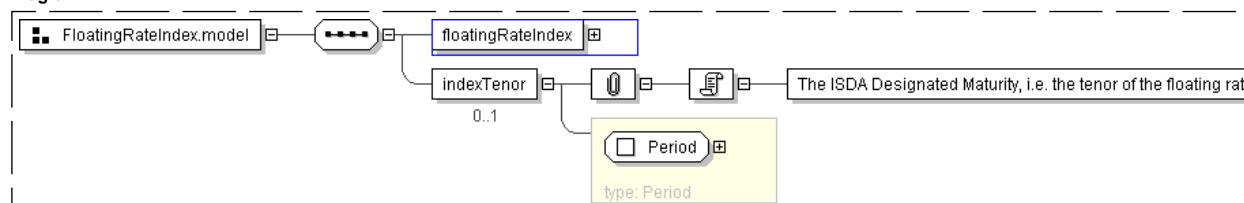
top

**Model Group: FloatingRateIndex.model**

<b>Name</b>	FloatingRateIndex.model
<b>Used by (from the same schema document)</b>	Complex Type <b>FloatingRate</b>

**XML Instance Representation**

```
<floatingRateIndex> FloatingRateIndex </floatingRateIndex> [1]
<indexTenor> Period </indexTenor> [0..1]
'The ISDA Designated Maturity, i.e. the tenor of the floating rate.'
```

**Diagram****Schema Component Representation**

```

<xsd:group name="FloatingRateIndex.model">
  <xsd:sequence>
    <xsd:element name="floatingRateIndex" type="# FloatingRateIndex " />
    <xsd:element name="indexTenor" type="# Period " minOccurs="0"/>
  </xsd:sequence>
</xsd:group>

```

[top](#)

## Model Group: PartialExercise.model

Name	PartialExercise.model
Used by (from the same schema document)	Complex Type <a href="#">MultipleExercise</a> , Complex Type <a href="#">PartialExercise</a>

### XML Instance Representation

```
<notionalReference> NotionalReference </notionalReference> [0..*]
```

'A pointer style reference to the associated notional schedule defined elsewhere in the document. This element has been made optional as part of its integration in the OptionBaseExtended, because not required for the options on securities.'

```
<integralMultipleAmount> xsd:decimal </integralMultipleAmount> [0..1]
```

'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.'

```
Start Choice [1]
```

```
<minimumNotionalAmount> xsd:decimal </minimumNotionalAmount> [1]
```

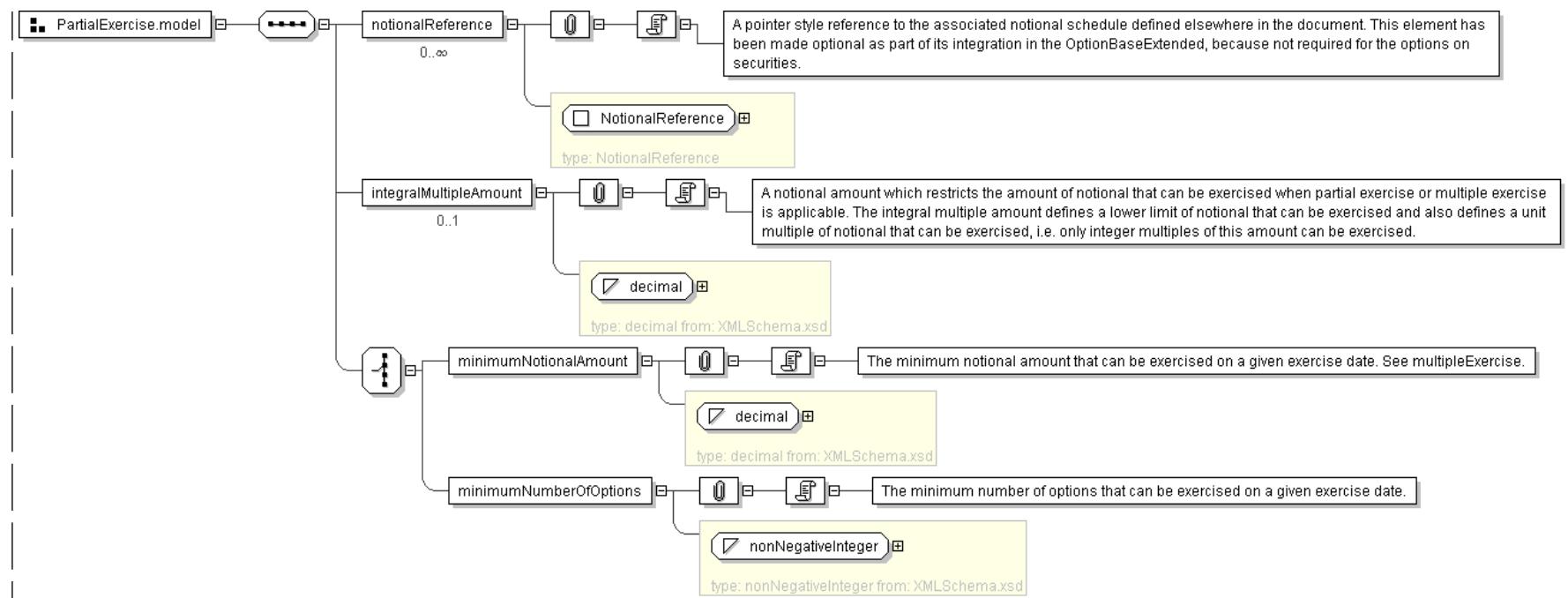
'The minimum notional amount that can be exercised on a given exercise date.  
See [multipleExercise](#).'

```
<minimumNumberOfOptions> xsd:nonNegativeInteger </minimumNumberOfOptions> [1]
```

'The minimum number of options that can be exercised on a given exercise date.'

```
End Choice
```

### Diagram

**Schema Component Representation**

```
<xsd:group name="PartialExercise.model">
  <xsd:sequence>
    <xsd:element name="notionalReference" type=" NotionalReference "
      minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element name="integralMultipleAmount" type=" xsd:decimal " minOccurs="0" />
    <xsd:choice>
      <xsd:element name="minimumNotionalAmount" type=" xsd:decimal "/>
      <!--Corrected the type from NonNegativeInteger-->
      <xsd:element name="minimumNumberOfOptions" type=" xsd:nonNegativeInteger " />
    </xsd:choice>
  </xsd:sequence>
</xsd:group>
```

top

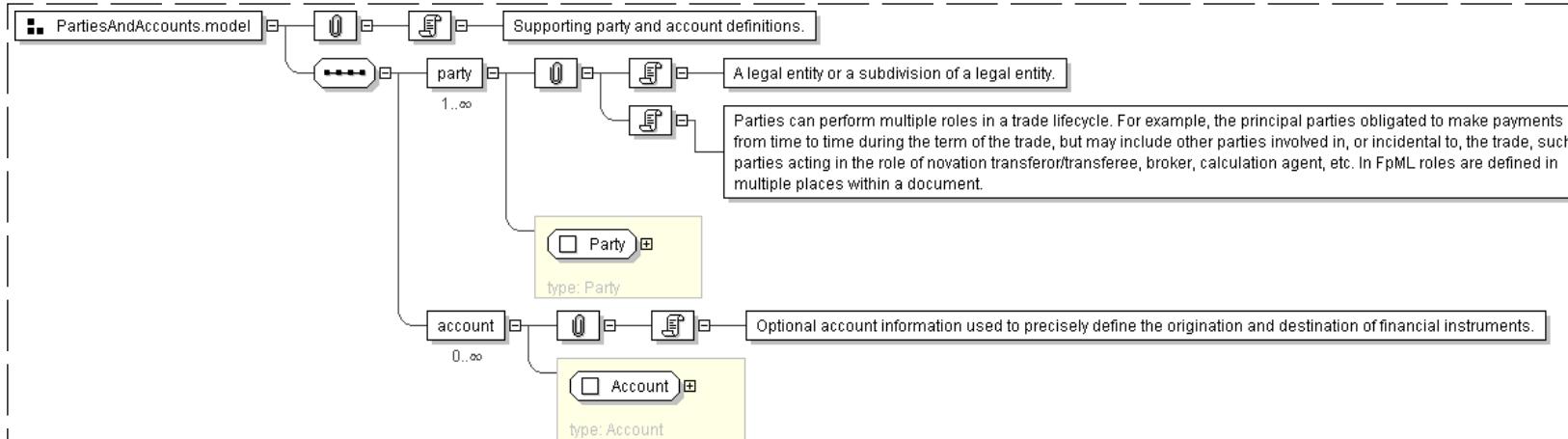
**Model Group: PartiesAndAccounts.model**

Name	PartiesAndAccounts.model
Documentation	Supporting party and account definitions.

**XML Instance Representation**

```
<party> Party </party> [1..*]
'A legal entity or a subdivision of a legal entity.', 'Parties can perform multiple roles in
a trade lifecycle. For example, the principal parties obligated to make payments from time
to time during the term of the trade, but may include other parties involved in, or
incidental to, the trade, such as parties acting in the role of novation transferor/
transferee, broker, calculation agent, etc. In FpML roles are defined in multiple places
within a document.'
```

```
<account> Account </account> [0..*]
'Optional account information used to precisely define the origination and destination
of financial instruments.'
```

**Diagram****Schema Component Representation**

```
<xsd:group name="PartiesAndAccounts.model">
  <xsd:sequence>
    <xsd:element name="party" type=" Party " maxOccurs="unbounded"/>
    <xsd:element name="account" type=" Account " minOccurs="0" maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:group>
```

top

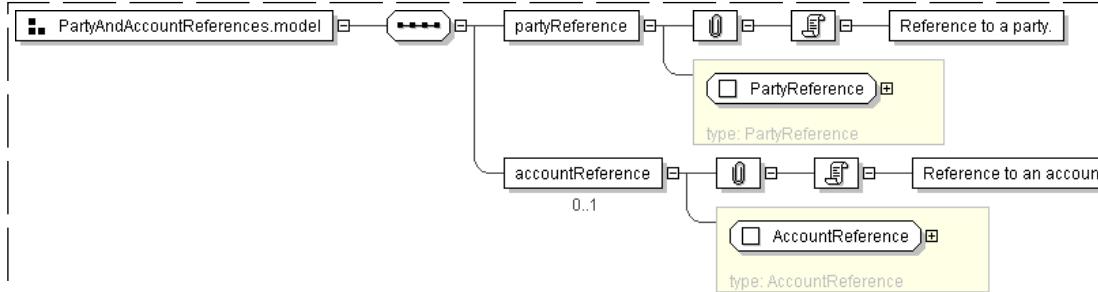
**Model Group: PartyAndAccountReferences.model**

<b>Name</b>	PartyAndAccountReferences.model
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">PartyRelationship</a> , Complex Type <a href="#">RelatedParty</a>

**XML Instance Representation**

```
<partyReference> PartyReference </partyReference> [1]
'Reference to a party.'

<accountReference> AccountReference </accountReference> [0..1]
'Reference to an account.'
```

**Diagram****Schema Component Representation**

```
<xsd:group name="PartyAndAccountReferences.model">
  <xsd:sequence>
```

```

<xsd:element name="partyReference" type=" PartyReference " />
<xsd:element name="accountReference" type=" AccountReference " minOccurs="0" />
</xsd:sequence>
</xsd:group>

```

**Model Group: PartyInformation.model**

<b>Name</b>	PartyInformation.model
<b>Documentation</b>	Information about a party for reporting purposes.

**XML Instance Representation**

```

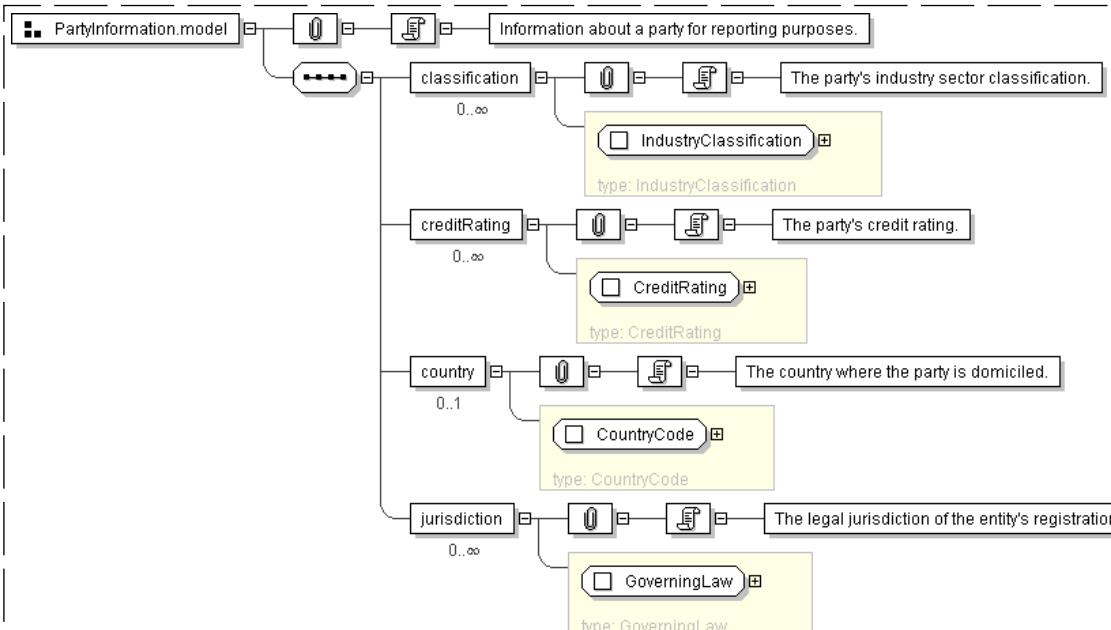
<classification> IndustryClassification </classification> [0..*]
'The party\'s industry sector classification.'

<creditRating> CreditRating </creditRating> [0..*]
'The party\'s credit rating.'

<country> CountryCode </country> [0..1]
'The country where the party is domiciled.'

<jurisdiction> GoverningLaw </jurisdiction> [0..*]
'The legal jurisdiction of the entity\'s registration.'

```

**Diagram****Schema Component Representation**

```

<xsd:group name="PartyInformation.model">
<xsd:sequence>
  <xsd:element name="classification" type=" IndustryClassification "
  minOccurs="0" maxOccurs="unbounded"/>
  <xsd:element name="creditRating" type=" CreditRating " minOccurs="0" maxOccurs="unbounded" />
  <xsd:element name="country" type=" CountryCode " minOccurs="0" />

```

```

<xsd:element name="jurisdiction" type=" GouverningLaw " minOccurs="0" maxOccurs="unbounded"/>
<!-- <xsd:element name="creditStatus" type="CreditStatus" minOccurs="0" > <xsd:annotation>
<xsd:documentation xml:lang="en">The legal jurisdiction of the entity's registration.</
  xsd:documentation> </xsd:annotation> </xsd:element> -->
</xsd:sequence>
</xsd:group>

```

**Model Group: PayerReceiver.model**

<b>Name</b>	PayerReceiver.model
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">ExerciseFee</a> , Complex Type <a href="#">ExerciseFeeSchedule</a> , Complex Type <a href="#">GrossCashflow</a> , Complex Type <a href="#">Payment</a> , Complex Type <a href="#">PaymentBaseExtended</a> , Complex Type <a href="#">SimplePayment</a>

**XML Instance Representation**

```

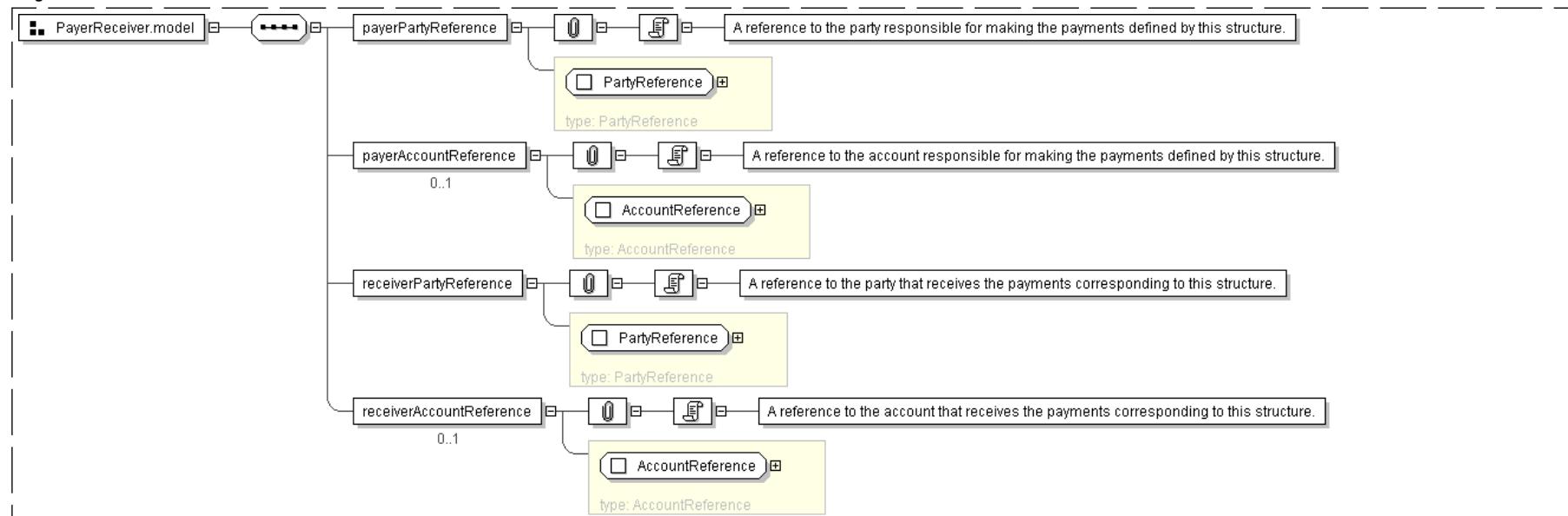
<payerPartyReference> PartyReference </payerPartyReference> [1]
'A reference to the party responsible for making the payments defined by this structure.'

<payerAccountReference> AccountReference </payerAccountReference> [0..1]
'A reference to the account responsible for making the payments defined by this structure.'

<receiverPartyReference> PartyReference </receiverPartyReference> [1]
'A reference to the party that receives the payments corresponding to this structure.'

<receiverAccountReference> AccountReference </receiverAccountReference> [0..1]
'A reference to the account that receives the payments corresponding to this structure.'

```

**Diagram****Schema Component Representation**

```

<xsd:group name="PayerReceiver.model">
  <xsd:sequence>
    <xsd:element name="payerPartyReference" type=" PartyReference " />
    <xsd:element name="payerAccountReference" type=" AccountReference " minOccurs="0"/>
    <xsd:element name="receiverPartyReference" type=" PartyReference " />
    <xsd:element name="receiverAccountReference" type=" AccountReference " minOccurs="0"/>
  </xsd:sequence>
</xsd:group>

```

```
</xsd:sequence>
</xsd:group>
```

## Model Group: PaymentDiscounting.model

### Name

PaymentDiscounting.model

### Documentation

A model group for representing the discounting elements that can be associated with a payment.

### XML Instance Representation

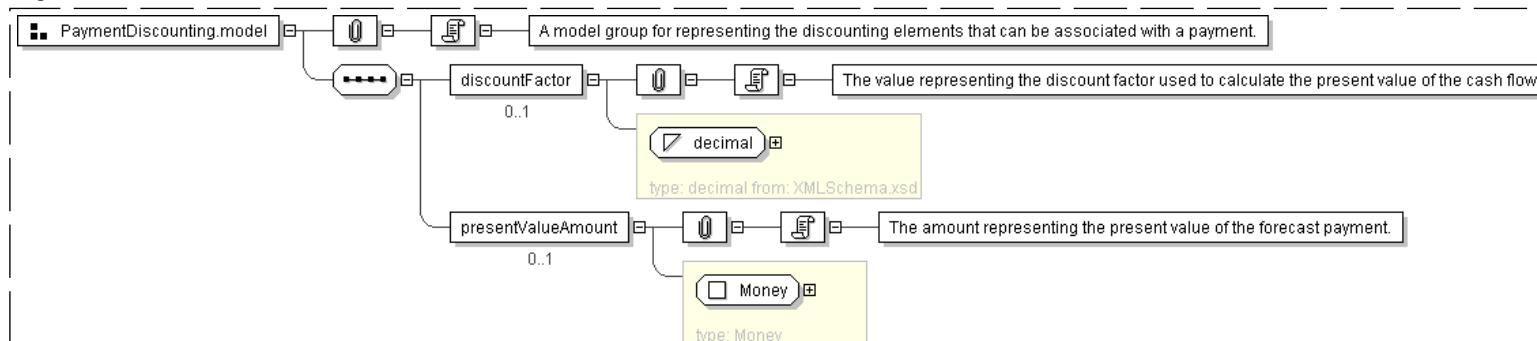
```
<discountFactor> xsd:decimal </discountFactor> [0..1]
```

'The value representing the discount factor used to calculate the present value of the cash flow.'

```
<presentValueAmount> Money </presentValueAmount> [0..1]
```

'The amount representing the present value of the forecast payment.'

### Diagram



### Schema Component Representation

```

<xsd:group name="PaymentDiscounting.model">
  <xsd:sequence>
    <xsd:element name="discountFactor" type="xsd:decimal" minOccurs="0"/>
    <xsd:element name="presentValueAmount" type="Money" minOccurs="0"/>
  </xsd:sequence>
</xsd:group>

```

## Model Group: Period.model

### Name

Period.model

### XML Instance Representation

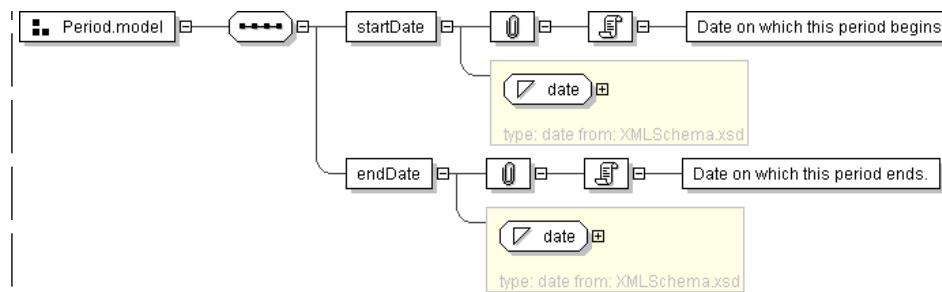
```
<startDate> xsd:date </startDate> [1]
```

'Date on which this period begins.'

```
<endDate> xsd:date </endDate> [1]
```

'Date on which this period ends.'

### Diagram

**Schema Component Representation**

```

<xsd:group name="Period.model">
  <xsd:sequence>
    <xsd:element name="startDate" type="xsd:date" />
    <xsd:element name="endDate" type="xsd:date" />
  </xsd:sequence>
</xsd:group>
  
```

[top](#)**Model Group: Premium.model**

<b>Name</b>	Premium.model
<b>Documentation</b>	A model group for representing the option premium when expressed in a way other than an amount.

**XML Instance Representation**

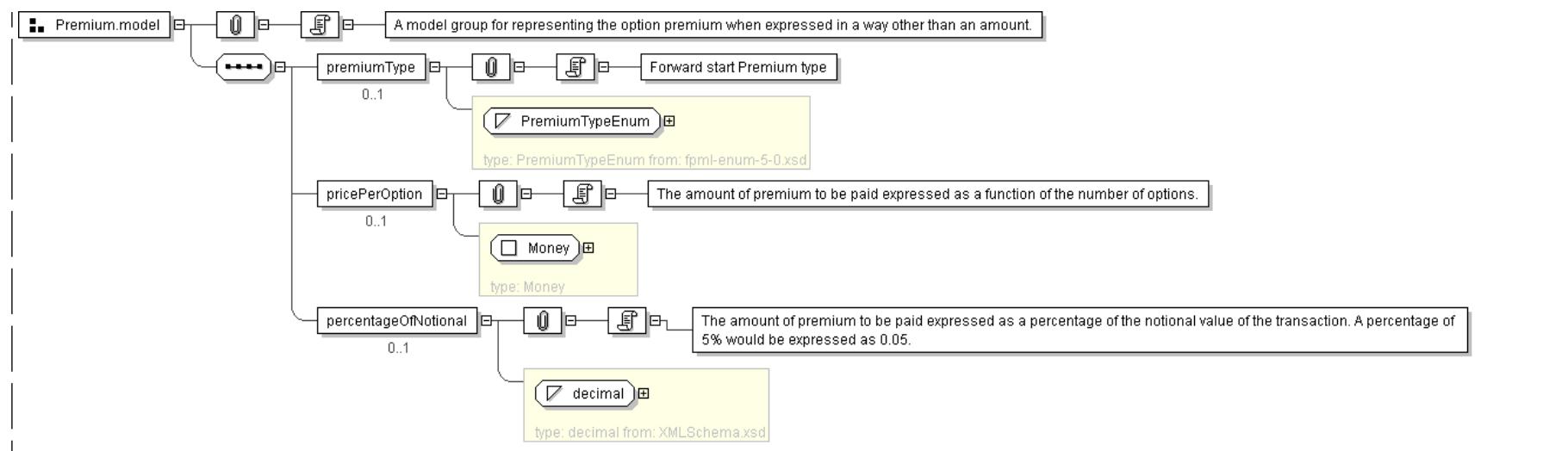
```

<premiumType> PremiumTypeEnum </premiumType> [0..1]
'Forward start Premium type'

<pricePerOption> Money </pricePerOption> [0..1]
'The amount of premium to be paid expressed as a function of the number of options.'

<percentageOfNotional> xsd:decimal </percentageOfNotional> [0..1]
'The amount of premium to be paid expressed as a percentage of the notional value of
the transaction. A percentage of 5% would be expressed as 0.05.'
  
```

**Diagram**



#### Schema Component Representation

```

<xsd:group name="Premium.model">
  <xsd:sequence>
    <xsd:element name="premiumType" type="PremiumTypeEnum" minOccurs="0"/>
    <xsd:element name="pricePerOption" type="Money" minOccurs="0"/>
    <xsd:element name="percentageOfNotional" type="xsd:decimal" minOccurs="0"/>
  </xsd:sequence>
</xsd:group>
  
```

top

#### Model Group: Product.model

Name	Product.model
Used by (from the same schema document)	Complex Type <a href="#">Product</a>

#### XML Instance Representation

```

<productType> ProductType </productType> [0..*]
'A classification of the type of product. FpML defines a simple product categorization using a coding scheme.'

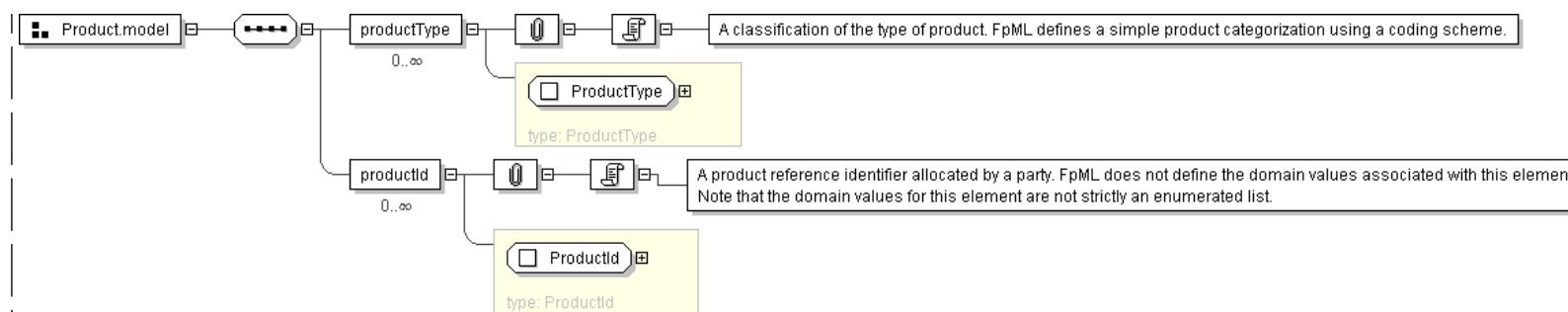
```

```

<productId> ProductId </productId> [0..*]
'A product 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.'
  
```

#### Diagram

**Schema Component Representation**

```

<xsd:group name="Product.model">
  <xsd:sequence>
    <xsd:element name="productType" type="#ProductType" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element name="productId" type="#ProductId" minOccurs="0" maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:group>

```

[top](#)**Model Group: RoutingExplicitDetails.model**

<b>Name</b>	RoutingExplicitDetails.model
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">RoutingExplicitDetails</a> , Complex Type <a href="#">RoutingIdsAndExplicitDetails</a>

**XML Instance Representation**

```

<routingName> xsd:string </routingName> [1]
'A real name that is used to identify a party involved in the routing of a payment.'

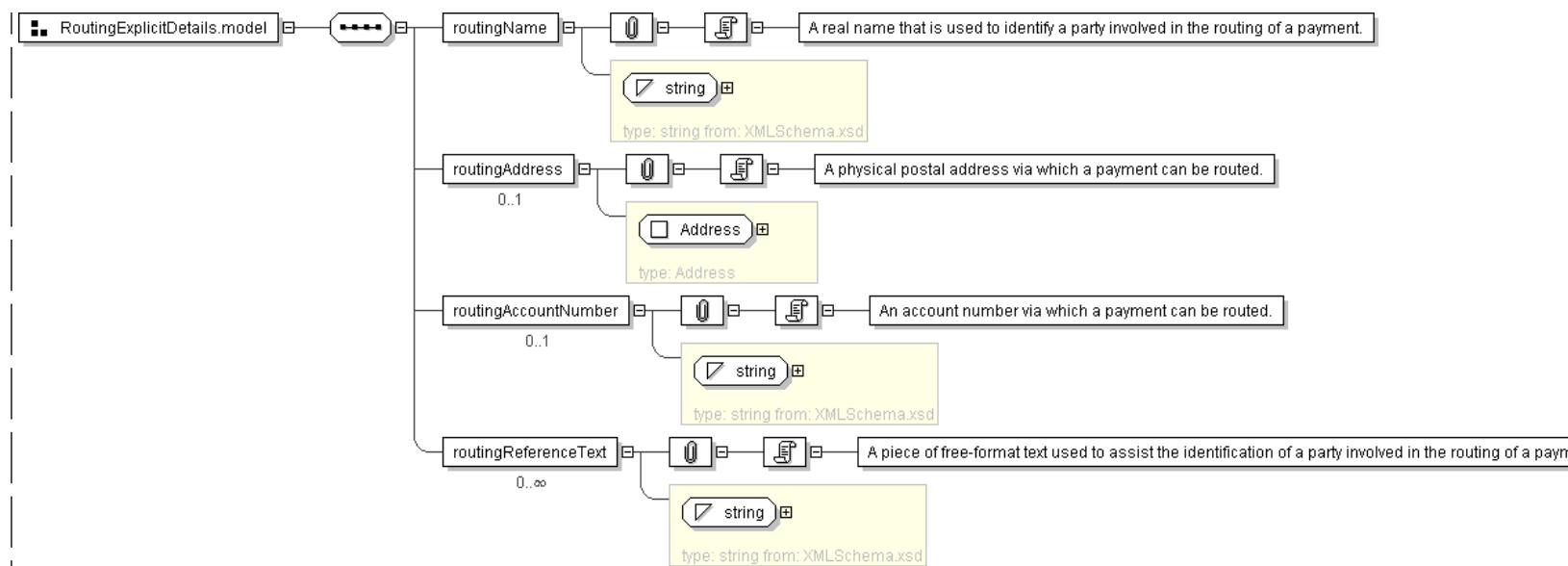
<routingAddress> Address </routingAddress> [0..1]
'A physical postal address via which a payment can be routed.'

<routingAccountNumber> xsd:string </routingAccountNumber> [0..1]
'An account number via which a payment can be routed.'

<routingReferenceText> xsd:string </routingReferenceText> [0..*]
'A piece of free-format text used to assist the identification of a party involved in
the routing of a payment.'

```

**Diagram**



#### Schema Component Representation

```

<xsd:group name="RoutingExplicitDetails.model">
  <xsd:sequence>
    <xsd:element name="routingName" type="xsd:string" />
    <xsd:element name="routingAddress" type="Address" minOccurs="0"/>
    <xsd:element name="routingAccountNumber" type="xsd:string" minOccurs="0"/>
    <xsd:element name="routingReferenceText" type="xsd:string" 
      minOccurs="0" maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:group>
  
```

top

#### Model Group: RoutingIdentification.model

Name	RoutingIdentification.model
Used by (from the same schema document)	Complex Type <a href="#">Beneficiary</a> , Complex Type <a href="#">CorrespondentInformation</a> , Complex Type <a href="#">IntermediaryInformation</a> , Complex Type <a href="#">Routing</a>

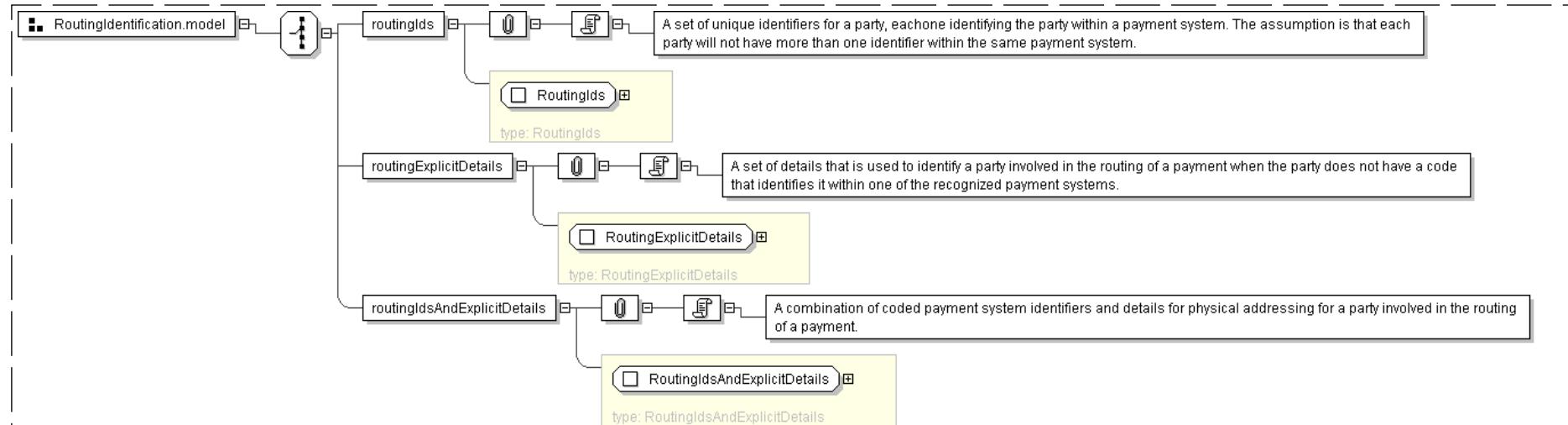
#### XML Instance Representation

```

Start Choice [1]
<routingIds> RoutingIds </routingIds> [1]
  'A set of unique identifiers for a party, each one identifying the party within a payment system. The assumption is that each party will not have more than one identifier within the same payment system.'

  <routingExplicitDetails> RoutingExplicitDetails </routingExplicitDetails> [1]
  'A set of details that is used to identify a party involved in the routing of a payment when the party does not have a code that identifies it within one of the recognized payment systems.'

  <routingIdsAndExplicitDetails> RoutingIdsAndExplicitDetails </routingIdsAndExplicitDetails> [1]
  'A combination of coded payment system identifiers and details for physical addressing for a party involved in the routing of a payment.'
End Choice
  
```

**Diagram****Schema Component Representation**

```

<xsd:group name="RoutingIdentification.model">
  <xsd:choice>
    <xsd:element name="routingIds" type=" RoutingIds "/>
    <xsd:element name="routingExplicitDetails" type=" RoutingExplicitDetails "/>
    <xsd:element name="routingIdsAndExplicitDetails" type=" RoutingIdsAndExplicitDetails "/>
  </xsd:choice>
</xsd:group>
  
```

[top](#)**Model Group: SettlementAmountOrCurrency.model**

Name	SettlementAmountOrCurrency.model
------	----------------------------------

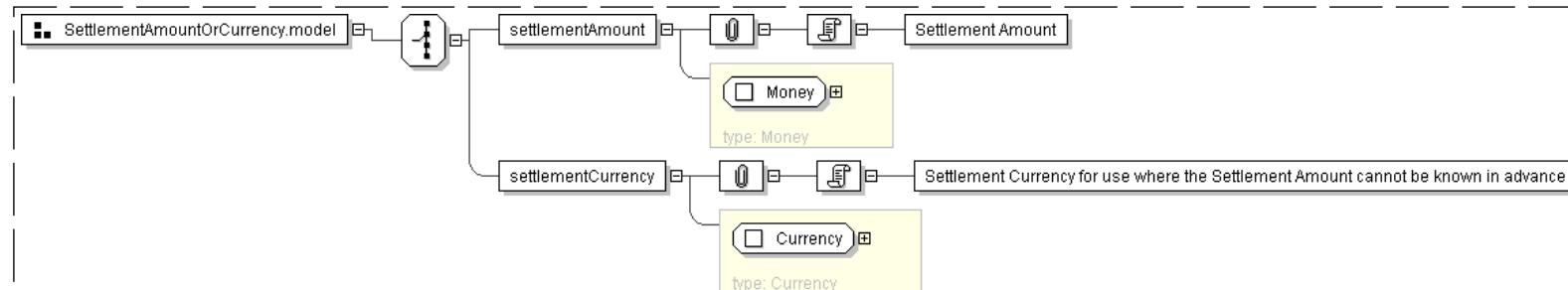
**XML Instance Representation**

```

Start Choice [1]
<settlementAmount> Money </settlementAmount> [1]
'Settlement Amount'

<settlementCurrency> Currency </settlementCurrency> [1]
'Settlement Currency for use where the Settlement Amount cannot be known in advance'

End Choice
  
```

**Diagram**

**Schema Component Representation**

```
<xsd:group name="SettlementAmountOrCurrency.model">
  <xsd:choice>
    <xsd:element name="settlementAmount" type=" Money " />
    <xsd:element name="settlementCurrency" type=" Currency " />
  </xsd:choice>
</xsd:group>
```

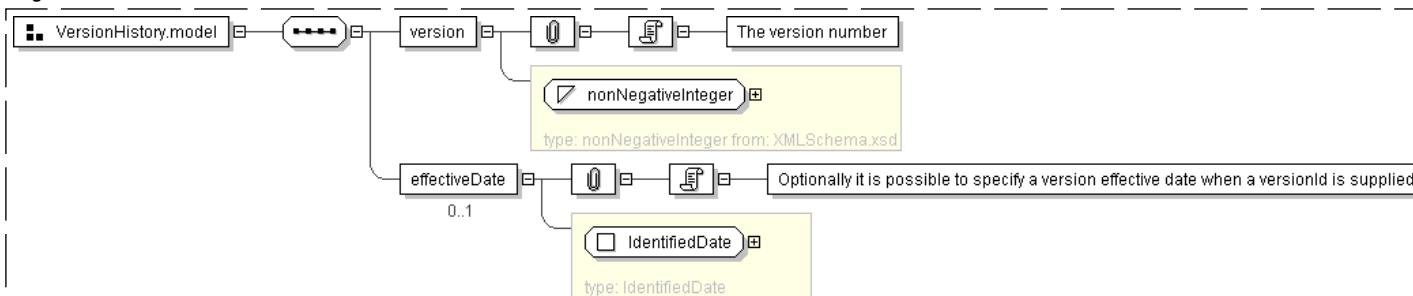
[top](#)**Model Group: VersionHistory.model****Name**

VersionHistory.model

**XML Instance Representation**

```
<version> xsd:nonNegativeInteger </version> [1]
'The version number'

<effectiveDate> IdentifiedDate </effectiveDate> [0..1]
'Optionally it is possible to specify a version effective date when a versionId is supplied.'
```

**Diagram****Schema Component Representation**

```
<xsd:group name="VersionHistory.model">
  <xsd:sequence>
    <xsd:element name="version" type=" xsd:nonNegativeInteger " />
    <xsd:element name="effectiveDate" type=" IdentifiedDate " minOccurs="0"/>
  </xsd:sequence>
</xsd:group>
```

[top](#)**Simple Type: CorrelationValue****Super-types:**xsd:decimal < **CorrelationValue** (by restriction)**Sub-types:**

None

**Name**

CorrelationValue

**Content**

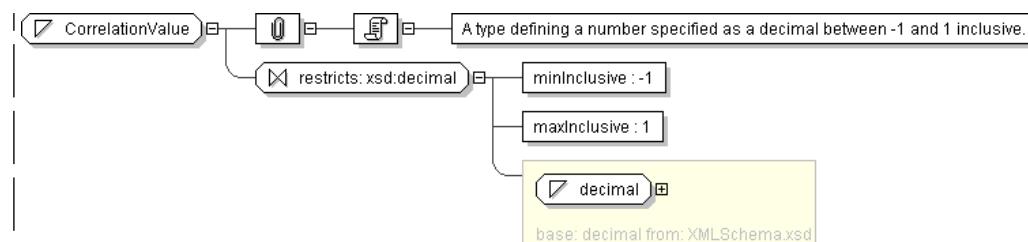
- Base XSD Type: decimal
- $-1 \leq value \leq 1$

**Documentation**

A type defining a number specified as a decimal between -1 and 1 inclusive.

**Diagram**

|

**Schema Component Representation**

```

<xsd:simpleType name="CorrelationValue">
  <xsd:restriction base=" xsd:decimal ">
    <xsd:minInclusive value="-1"/>
    <xsd:maxInclusive value="1"/>
  </xsd:restriction>
</xsd:simpleType>

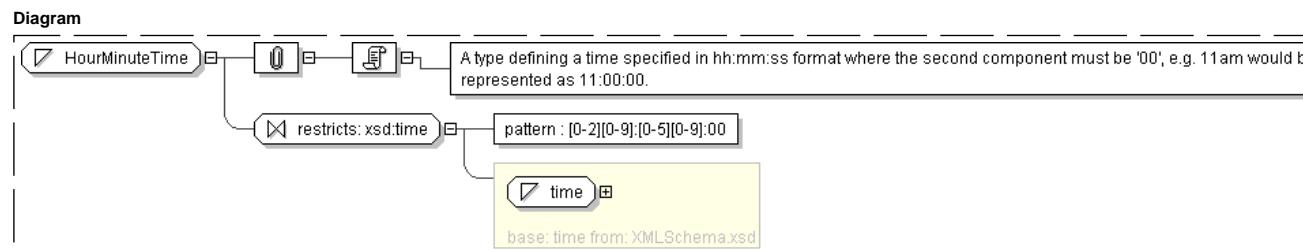
```

top

**Simple Type: HourMinuteTime**

<b>Super-types:</b>	<a href="#">xsd:time</a> < <b>HourMinuteTime</b> (by restriction)
<b>Sub-types:</b>	None

<b>Name</b>	HourMinuteTime
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">BusinessCenterTime</a> , Complex Type <a href="#">PrevailingTime</a>
<b>Content</b>	<ul style="list-style-type: none"> <li>• Base XSD Type: time</li> <li>• <code>pattern = [0-2][0-9]:[0-5][0-9]:00</code></li> </ul>
<b>Documentation</b>	A type defining 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.

**Schema Component Representation**

```

<xsd:simpleType name="HourMinuteTime">
  <xsd:restriction base=" xsd:time ">
    <xsd:pattern value="[0-2][0-9]:[0-5][0-9]:00"/>
  </xsd:restriction>
</xsd:simpleType>

```

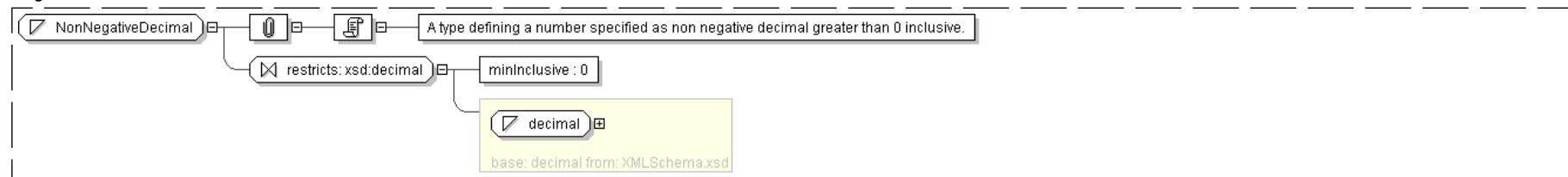
top

**Simple Type: NonNegativeDecimal**

<b>Super-types:</b>	<a href="#">xsd:decimal</a> < <b>NonNegativeDecimal</b> (by restriction)
<b>Sub-types:</b>	None

<b>Name</b>	NonNegativeDecimal
-------------	--------------------

<b>Used by (from the same schema document)</b>	Complex Type <a href="#">MultipleExercise</a> , Complex Type <a href="#">NonNegativeMoney</a> , Complex Type <a href="#">NonNegativeSchedule</a> , Complex Type <a href="#">NonNegativeStep</a>
<b>Content</b>	<ul style="list-style-type: none"> <li>• Base XSD Type: decimal</li> <li>• <i>value</i> &gt;= 0</li> </ul>
<b>Documentation</b>	A type defining a number specified as non negative decimal greater than 0 inclusive.

**Diagram****Schema Component Representation**

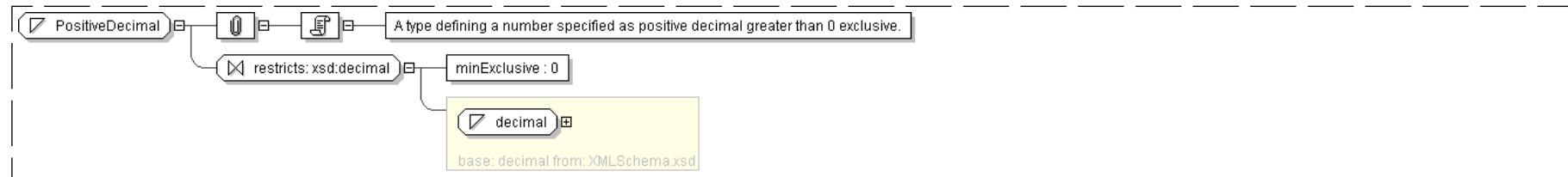
```

<xsd:simpleType name="NonNegativeDecimal">
  <xsd:restriction base=" xsd:decimal ">
    <xsd:minInclusive value="0"/>
  </xsd:restriction>
</xsd:simpleType>
  
```

[top](#)**Simple Type: PositiveDecimal**

<b>Super-types:</b>	<a href="#">xsd:decimal</a> < <b>PositiveDecimal</b> (by restriction)
<b>Sub-types:</b>	None

<b>Name</b>	PositiveDecimal
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">PositiveMoney</a> , Complex Type <a href="#">PositiveSchedule</a> , Complex Type <a href="#">PositiveStep</a>
<b>Content</b>	<ul style="list-style-type: none"> <li>• Base XSD Type: decimal</li> <li>• <i>value</i> &gt; 0</li> </ul>
<b>Documentation</b>	A type defining a number specified as positive decimal greater than 0 exclusive.

**Diagram****Schema Component Representation**

```

<xsd:simpleType name="PositiveDecimal">
  <xsd:restriction base=" xsd:decimal ">
    <xsd:minExclusive value="0"/>
  </xsd:restriction>
</xsd:simpleType>
  
```

[top](#)**Simple Type: RestrictedPercentage**

**Super-types:** xsd:decimal < **RestrictedPercentage** (by restriction)

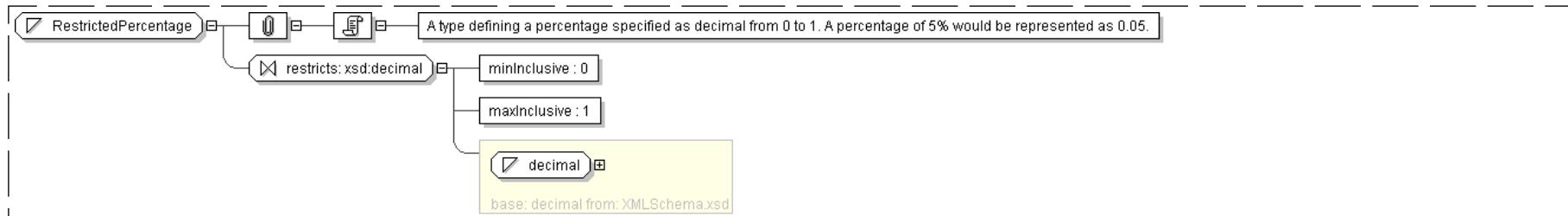
**Sub-types:** None

<b>Name</b>	RestrictedPercentage
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">AverageDailyTradingVolumeLimit</a>

<b>Content</b>	<ul style="list-style-type: none"> <li>• Base XSD Type: decimal</li> <li>• <math>0 \leq value \leq 1</math></li> </ul>
----------------	--

<b>Documentation</b>	A type defining a percentage specified as decimal from 0 to 1. A percentage of 5% would be represented as 0.05.
----------------------	---

#### Diagram



#### Schema Component Representation

```

<xsd:simpleType name="RestrictedPercentage">
  <xsd:restriction base="xsd:decimal" >
    <xsd:minInclusive value="0" />
    <xsd:maxInclusive value="1" />
  </xsd:restriction>
</xsd:simpleType>
  
```

[top](#)

## Simple Type: Scheme

**Super-types:** xsd:normalizedString < **Scheme** (by restriction)

**Sub-types:**

- [AccountId](#) (by extension)
- [AccountName](#) (by extension)
- [AgreementType](#) (by extension)
- [AgreementVersion](#) (by extension)
- [AssetClass](#) (by extension)
- [BrokerConfirmationType](#) (by extension)
- [BusinessCenter](#) (by extension)
- [CashflowId](#) (by extension)
- [CashflowType](#) (by extension)
- [ClearanceSystem](#) (by extension)
- [ContractualDefinitions](#) (by extension)
- [ContractualSupplement](#) (by extension)
- [CreditSeniority](#) (by extension)
- [CreditSupportAgreementIdentifier](#) (by extension)
- [CreditSupportAgreementType](#) (by extension)
- [Currency](#) (by extension)
- [DayCountFraction](#) (by extension)
- [DeterminationMethod](#) (by extension)
- [EntityId](#) (by extension)
- [EntityName](#) (by extension)
- [ExchangeId](#) (by extension)
- [FloatingRateIndex](#) (by extension)
- [GoverningLaw](#) (by extension)
- [InformationProvider](#) (by extension)
- [InstrumentId](#) (by extension)
- [InterpolationMethod](#) (by extension)
- [MainPublication](#) (by extension)

- [MasterAgreementType](#) (by extension)
  - [MasterAgreementVersion](#) (by extension)
  - [MasterConfirmationAnnexType](#) (by extension)
  - [MasterConfirmationType](#) (by extension)
  - [MatchId](#) (by extension)
  - [MatrixType](#) (by extension)
  - [MatrixTerm](#) (by extension)
  - [MimeType](#) (by extension)
  - [PartyId](#) (by extension)
  - [PartyName](#) (by extension)
  - [PartyRole](#) (by extension)
  - [PartyRoleType](#) (by extension)
  - [IndustryClassification](#) (by extension)
  - [CreditRating](#) (by extension)
  - [PaymentType](#) (by extension)
  - [ProductId](#) (by extension)
  - [ProductType](#) (by extension)
  - [RateSourcePage](#) (by extension)
  - [ReferenceAmount](#) (by extension)
  - [ReferenceBankId](#) (by extension)
  - [RoutingId](#) (by extension)
  - [SettlementMethod](#) (by extension)
  - [SettlementPriceDefaultElection](#) (by extension)
  - [SettlementPriceSource](#) (by extension)
  - [SpreadScheduleType](#) (by extension)
  - [TimezoneLocation](#) (by extension)

<b>Name</b>	Scheme
<b>Content</b>	<ul style="list-style-type: none"><li>• Base XSD Type: normalizedString</li><li>• <i>length</i> &lt;= 255</li></ul>
<b>Documentation</b>	The base class for all types which define coding schemes.



## Schema Component Representation

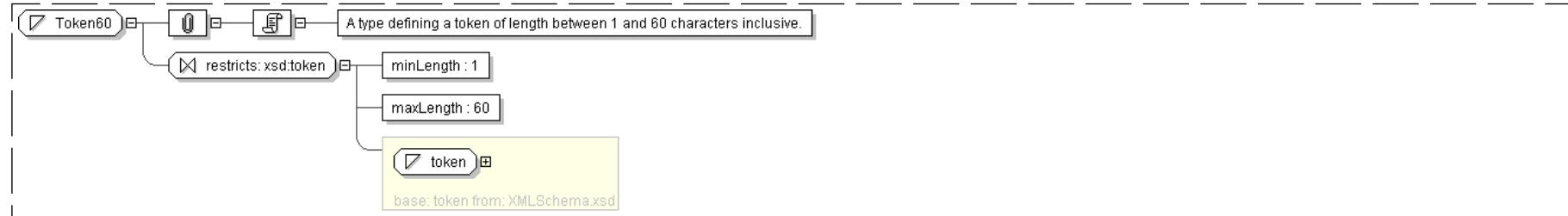
```
<xsd:simpleType name="Schema">
  <xsd:restriction base="xsd:normalizedString">
    <xsd:maxLength value="255" />
  </xsd:restriction>
</xsd:simpleType>
```

top

## Simple Type: Token60

**Super-types:** [xsd:token](#) < **Token60** (by restriction)  
**Sub-types:** None

Name	Token60
Content	<ul style="list-style-type: none"><li>• Base XSD Type: token</li><li>• <math>length \geq 1</math></li></ul>
Documentation	A type defining a token of length between 1 and 60 characters inclusive.

**Diagram****Schema Component Representation**

```

<xsd:simpleType name="Token60">
  <xsd:restriction base=" xsd:token ">
    <xsd:minLength value="1"/>
    <xsd:maxLength value="60"/>
  </xsd:restriction>
</xsd:simpleType>
  
```

top

**Legend**

**Complex Type:**  
Schema Component Type

**AusAddress**  
Schema Component Name

<b>Super-types:</b>	<a href="#">Address</a> < AusAddress (by extension)
<b>Sub-types:</b>	<ul style="list-style-type: none"> <li>• <a href="#">QLDAddress</a> (by restriction)</li> </ul>

If this schema component is a type definition, its type hierarchy is shown in a gray-bordered box.

<b>Name</b>	AusAddress
<b>Abstract</b>	no

The table above displays the properties of this schema component.

**XML Instance Representation**

```

<... country="Australia" >
<unitNo> string </unitNo> [0..1]
<houseNo> string </houseNo> [1]
<street> string </street> [1]
Start Choice [1]
<city> string </city> [1]
<town> string </town> [1]
End Choice
<state> AusStates </state> [1]
<postcode> string <><pattern = [1-9][0-9]{3}></postcode> [1]
</...>
  
```

The XML Instance Representation table above shows the schema component's content as an XML instance.

- The minimum and maximum occurrence of elements and attributes are provided in square brackets, e.g. [0..1].
- Model group information are shown in gray, e.g. Start Choice ... End Choice.
- For type derivations, the elements and attributes that have been added to or changed from the base type's content are shown in **bold**.
- If an element/attribute has a fixed value, the fixed value is shown in green, e.g. `country="Australia"`.
- Otherwise, the type of the element/attribute is displayed.
  - If the element/attribute's type is in the schema, a link is provided to it.
  - For local simple type definitions, the constraints are displayed in angle brackets, e.g. `<><pattern = [1-9][0-9]{3}></postcode>`.

**Schema Component Representation**

```

<complexType name="AusAddress">
<complexContent>
<extension base=" Address ">
<sequence>
<element name="state" type=" AusStates " />
<element name="postcode">
<simpleType>
<restriction base=" string ">
<pattern value="[1-9][0-9]{3}" />
</restriction>
</simpleType>
</element>
</sequence>
<attribute name="country" type=" string " fixed="Australia" />
</extension>
</complexContent>
</complexType>

```

The Schema Component Representation table above displays the underlying XML representation of the schema component. (Annotations are not shown.)

[top](#)

## Glossary

**Abstract** (Applies to complex type definitions and element declarations). An abstract element or complex type cannot be used to validate an element instance. If there is a reference to an abstract element, only element declarations that can substitute the abstract element can be used to validate the instance. For references to abstract type definitions, only derived types can be used.

**All Model Group** Child elements can be provided *in any order* in instances. See: <http://www.w3.org/TR/xmlschema-1/#element-all>.

**Choice Model Group** *Only one* from the list of child elements and model groups can be provided in instances. See: <http://www.w3.org/TR/xmlschema-1/#element-choice>.

**Collapse Whitespace Policy** Replace tab, line feed, and carriage return characters with space character (Unicode character 32). Then, collapse contiguous sequences of space characters into single space character, and remove leading and trailing space characters.

**Disallowed Substitutions** (Applies to element declarations). If *substitution* is specified, then *substitution group* members cannot be used in place of the given element declaration to validate element instances. If *derivation methods*, e.g. extension, restriction, are specified, then the given element declaration will not validate element instances that have types derived from the element declaration's type using the specified derivation methods. Normally, element instances can override their declaration's type by specifying an *xsi:type* attribute.

**Key Constraint** Like *Uniqueness Constraint*, but additionally requires that the specified value(s) must be provided. See: [http://www.w3.org/TR/xmlschema-1/#clidentity-constraint\\_Definitions](http://www.w3.org/TR/xmlschema-1/#clidentity-constraint_Definitions).

**Key Reference Constraint** Ensures that the specified value(s) must match value(s) from a *Key Constraint* or *Uniqueness Constraint*. See: [http://www.w3.org/TR/xmlschema-1/#clidentity-constraint\\_Definitions](http://www.w3.org/TR/xmlschema-1/#clidentity-constraint_Definitions).

**Model Group** Groups together element content, specifying the order in which the element content can occur and the number of times the group of element content may be repeated. See: [http://www.w3.org/TR/xmlschema-1/#Model\\_Groups](http://www.w3.org/TR/xmlschema-1/#Model_Groups).

**Nillable** (Applies to element declarations). If an element declaration is nillable, instances can use the *xsi:nil* attribute. The *xsi:nil* attribute is the boolean attribute, *nil*, from the <http://www.w3.org/2001/XMLSchema-instance> namespace. If an element instance has an *xsi:nil* attribute set to true, it can be left empty, even though its element declaration may have required content.

**Notation** A notation is used to identify the format of a piece of data. Values of elements and attributes that are of type, NOTATION, must come from the names of declared notations. See: [http://www.w3.org/TR/xmlschema-1/#cNotation\\_Declarations](http://www.w3.org/TR/xmlschema-1/#cNotation_Declarations).

**Preserve Whitespace Policy** Preserve whitespaces exactly as they appear in instances.

**Prohibited Derivations** (Applies to type definitions). Derivation methods that cannot be used to create sub-types from a given type definition.

**Prohibited Substitutions** (Applies to complex type definitions). Prevents sub-types that have been derived using the specified derivation methods from validating element instances in place of the given type definition.

**Replace Whitespace Policy** Replace tab, line feed, and carriage return characters with space character (Unicode character 32).

**Sequence Model Group** Child elements and model groups must be provided *in the specified order* in instances. See: <http://www.w3.org/TR/xmlschema-1/#element-sequence>.

**Substitution Group** Elements that are *members* of a substitution group can be used wherever the *head* element of the substitution group is referenced.

**Substitution Group Exclusions** (Applies to element declarations). Prohibits element declarations from nominating themselves as being able to substitute a given element declaration, if they have types that are derived from the original element's type using the specified derivation methods.

**Target Namespace** The target namespace identifies the namespace that components in this schema belongs to. If no target namespace is provided, then the schema components do not belong to any namespace.

**Uniqueness Constraint** Ensures uniqueness of an element/attribute value, or a combination of values, within a specified scope. See: [http://www.w3.org/TR/xmlschema-1/#clidentity-constraint\\_Definitions](http://www.w3.org/TR/xmlschema-1/#clidentity-constraint_Definitions).

---

top

Generated by [eXoYgen](#) / [XML Editor](#) using a modified version of [xs3p](#) that adds schema diagrams and chunking support.

# XML Schema Documentation

## Table of Contents

- [Schema Document Properties](#)
- [Global Declarations](#)
  - [Element: varianceOptionTransactionSupplement](#)
  - [Element: varianceSwap](#)
  - [Element: varianceSwapTransactionSupplement](#)
- [Global Definitions](#)
  - [Complex Type: VarianceAmount](#)
  - [Complex Type: VarianceLeg](#)
  - [Complex Type: VarianceOptionTransactionSupplement](#)
  - [Complex Type: VarianceSwap](#)
  - [Complex Type: VarianceSwapTransactionSupplement](#)
- [Legend](#)
- [Glossary](#)

[top](#)

## Schema Document Properties

<b>Target Namespace</b>	<a href="http://www.fpml.org/FpML-5/confirmation">http://www.fpml.org/FpML-5/confirmation</a>
<b>Version</b>	\$Revision: 2527 \$
<b>Element and Attribute Namespaces</b>	<ul style="list-style-type: none"> <li>• Global element and attribute declarations belong to this schema's target namespace.</li> <li>• By default, local element declarations belong to this schema's target namespace.</li> <li>• By default, local attribute declarations have no namespace.</li> </ul>
<b>Schema Composition</b>	<ul style="list-style-type: none"> <li>• This schema includes components from the following schema document(s):           <ul style="list-style-type: none"> <li>◦ <a href="#">fpml-eqd-5-0.xsd</a></li> </ul> </li> </ul>

## Declared Namespaces

Prefix	Namespace
Default namespace	<a href="http://www.fpml.org/FpML-5/confirmation">http://www.fpml.org/FpML-5/confirmation</a>
xml	<a href="http://www.w3.org/XML/1998/namespace">http://www.w3.org/XML/1998/namespace</a>
fpml-annotation	<a href="http://www.fpml.org/annotation">http://www.fpml.org/annotation</a>
xsd	<a href="http://www.w3.org/2001/XMLSchema">http://www.w3.org/2001/XMLSchema</a>
ecore	<a href="http://www.eclipse.org/emf/2002/Ecore">http://www.eclipse.org/emf/2002/Ecore</a>

## Schema Component Representation

```

<xsd:schema targetNamespace="http://www.fpml.org/FpML-5/confirmation"
documentRoot="FpML" nsPrefix="conf" package="org.fpml.confirmation" version="$Revision: 2527 "
$ elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xsd:include schemaLocation="fpml-eqd-5-0.xsd"/>
  ...
</xsd:schema>

```

[top](#)

## Global Declarations

### Element: varianceOptionTransactionSupplement

- This element can be used wherever the following element is referenced:
  - [product](#)

<b>Name</b>	varianceOptionTransactionSupplement
<b>Type</b>	<a href="#">VarianceOptionTransactionSupplement</a>

**Nillable**

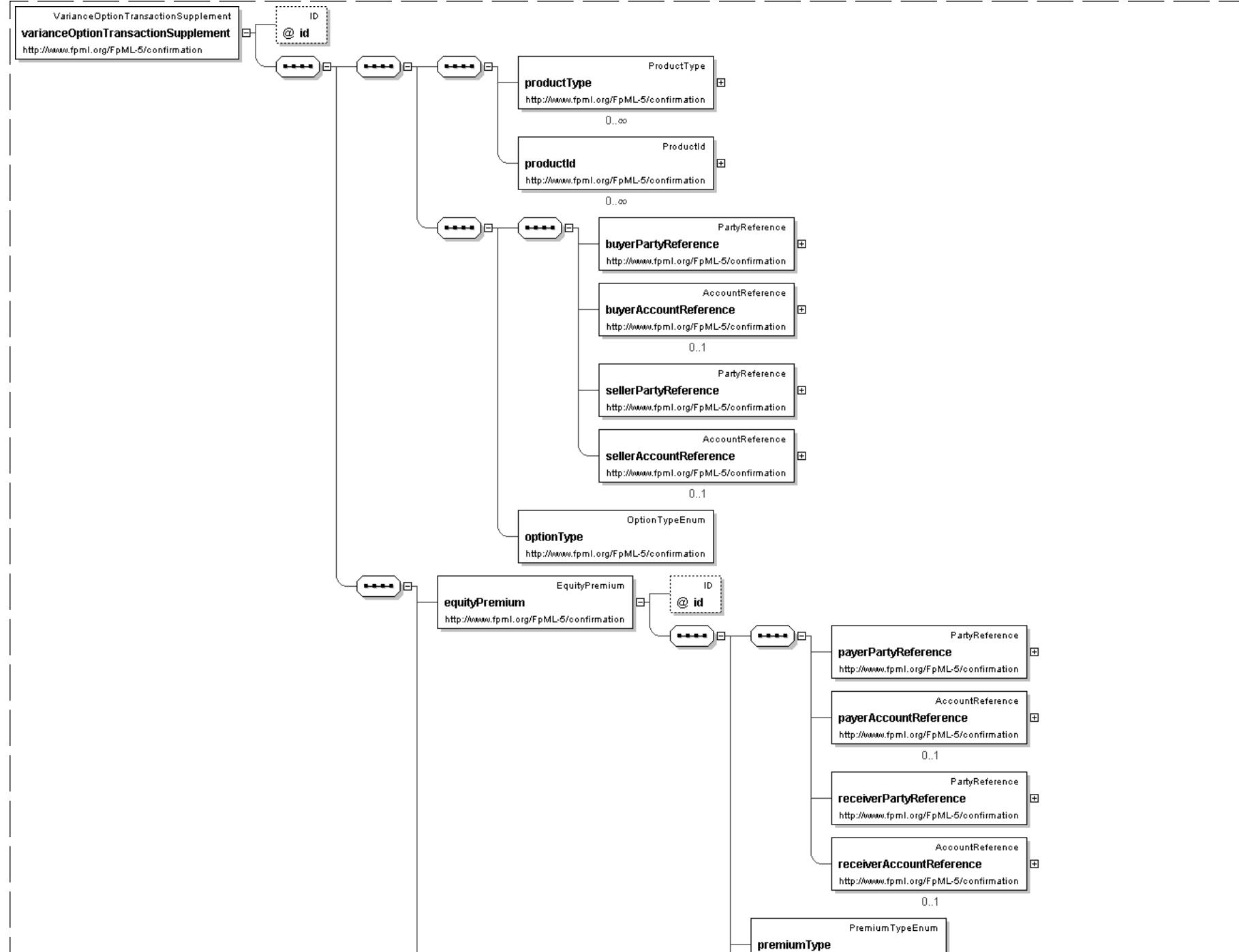
no

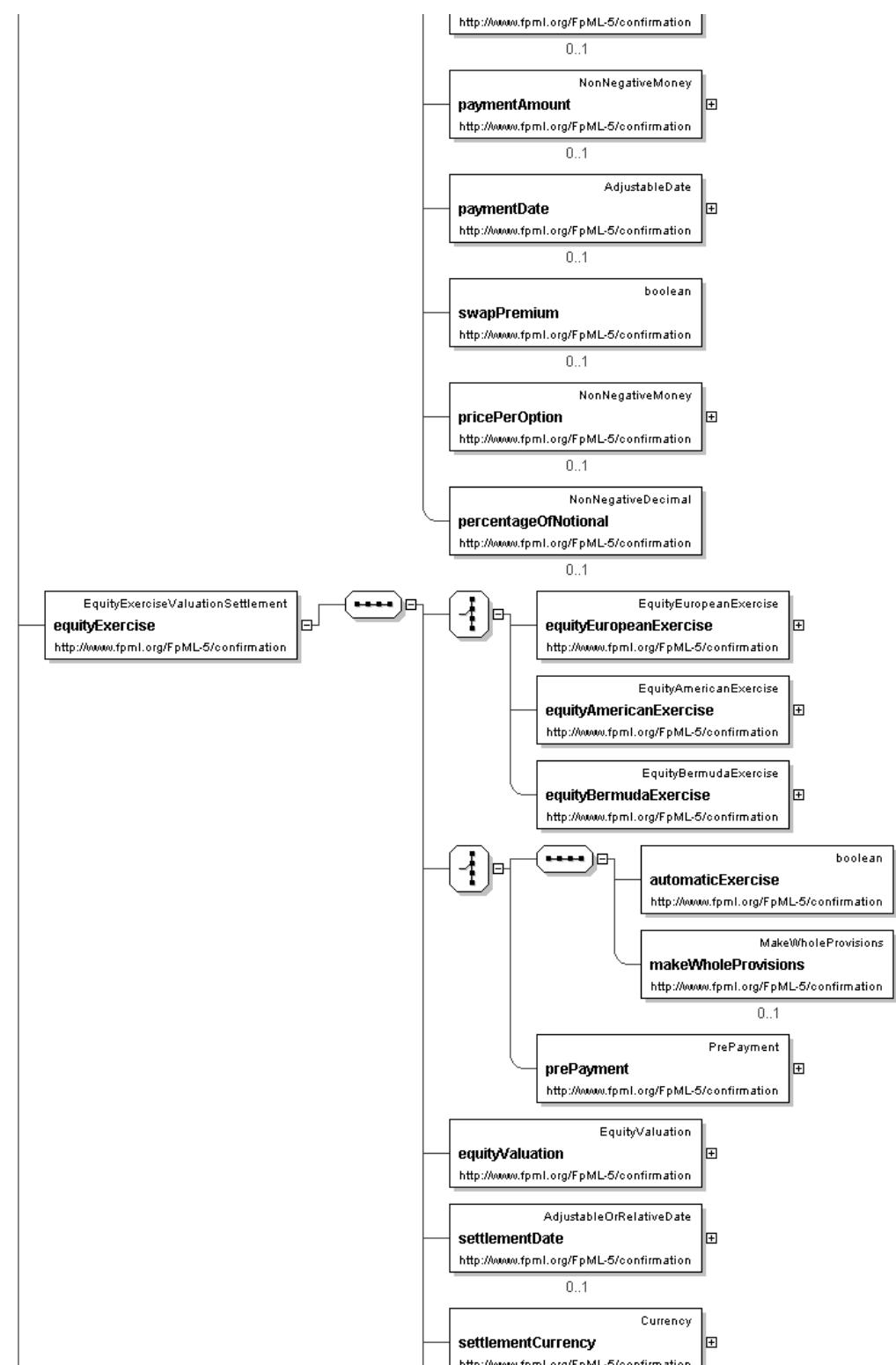
**Abstract**

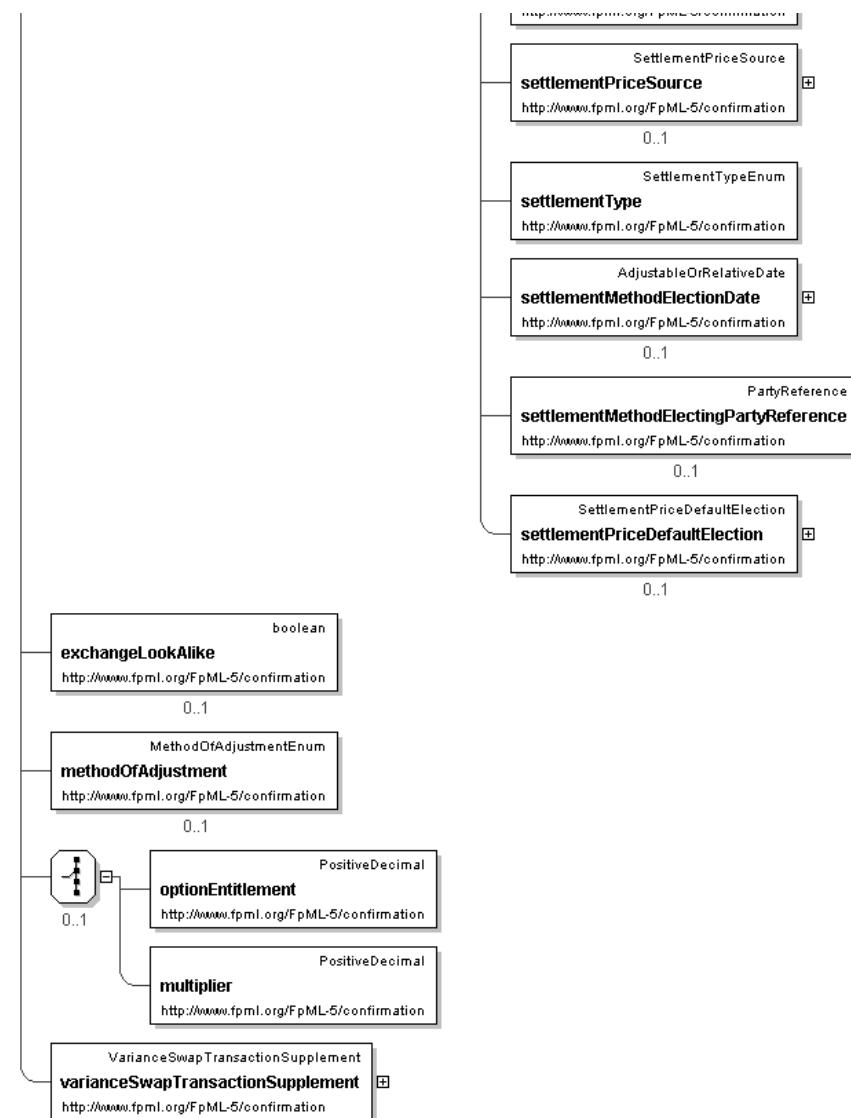
no

**Documentation**

Specifies the structure of a variance option.

**Logical Diagram**



**XML Instance Representation**

```

<varianceOptionTransactionSupplement
id=" xsd:ID [0..1]>
  <productType> ProductType </productType> [0..*]
  'A classification of the type of product. FpML defines a simple product categorization using
  a coding scheme.'
  <productId> ProductId </productId> [0..*]
  'A product 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.'
  <buyerPartyReference> PartyReference </buyerPartyReference> [1]
  'A reference to the party that buys this instrument, ie. pays for this instrument and
  receives the rights defined by it. See 2000 ISDA definitions Article 11.1 (b). In the case
  of FRAs this the fixed rate payer.'

```

```

<buyerAccountReference> AccountReference </buyerAccountReference> [0..1]
'A reference to the account that buys this instrument.'

<sellerPartyReference> PartyReference </sellerPartyReference> [1]
'A reference to the party that sells ("writes") this instrument, i.e. that grants the
rights defined by this instrument and in return receives a payment for it. See 2000
ISDA definitions Article 11.1 (a). In the case of FRAs this is the floating rate payer.'

<sellerAccountReference> AccountReference </sellerAccountReference> [0..1]
'A reference to the account that sells this instrument.'

<optionType> OptionTypeEnum </optionType> [1]
'The type of option transaction. From a usage standpoint, put/call is the default option
type, while payer/receiver indicator is used for options index credit default
swaps, consistently with the industry practice. Straddle is used for the case of
straddle strategy, that combines a call and a put with the same strike.'

<equityPremium> EquityPremium </equityPremium> [1]
'The variance option premium payable by the buyer to the seller.'

<equityExercise> EquityExerciseValuationSettlement </equityExercise> [1]
'The parameters for defining how the equity option can be exercised, how it is valued and
how it is settled.'

<exchangeLookAlike> xsd:boolean </exchangeLookAlike> [0..1]
'For a share option transaction, a flag used to indicate whether the transaction is to
be treated as an \'exchange look-alike\'. This designation has significance for how
share adjustments (arising from corporate actions) will be determined for the transaction.
For an \'exchange look-alike\' transaction the relevant share adjustments will follow that
for a corresponding designated contract listed on the related exchange (referred to as
Options Exchange Adjustment (ISDA defined term)), otherwise the share adjustments will
be determined by the calculation agent (referred to as Calculation Agent Adjustment
(ISDA defined term)).'

<methodOfAdjustment> MethodOfAdjustmentEnum </methodOfAdjustment> [0..1]
'Defines how adjustments will be made to the contract should one or more of the
extraordinary events occur.'

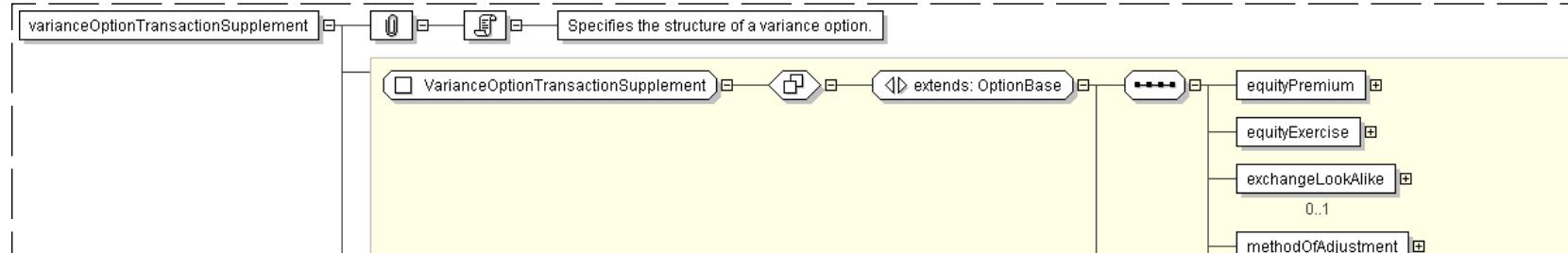
Start Choice [0..1]
  <optionEntitlement> PositiveDecimal </optionEntitlement> [1]
    'The number of shares per option comprised in the option transaction supplement.'

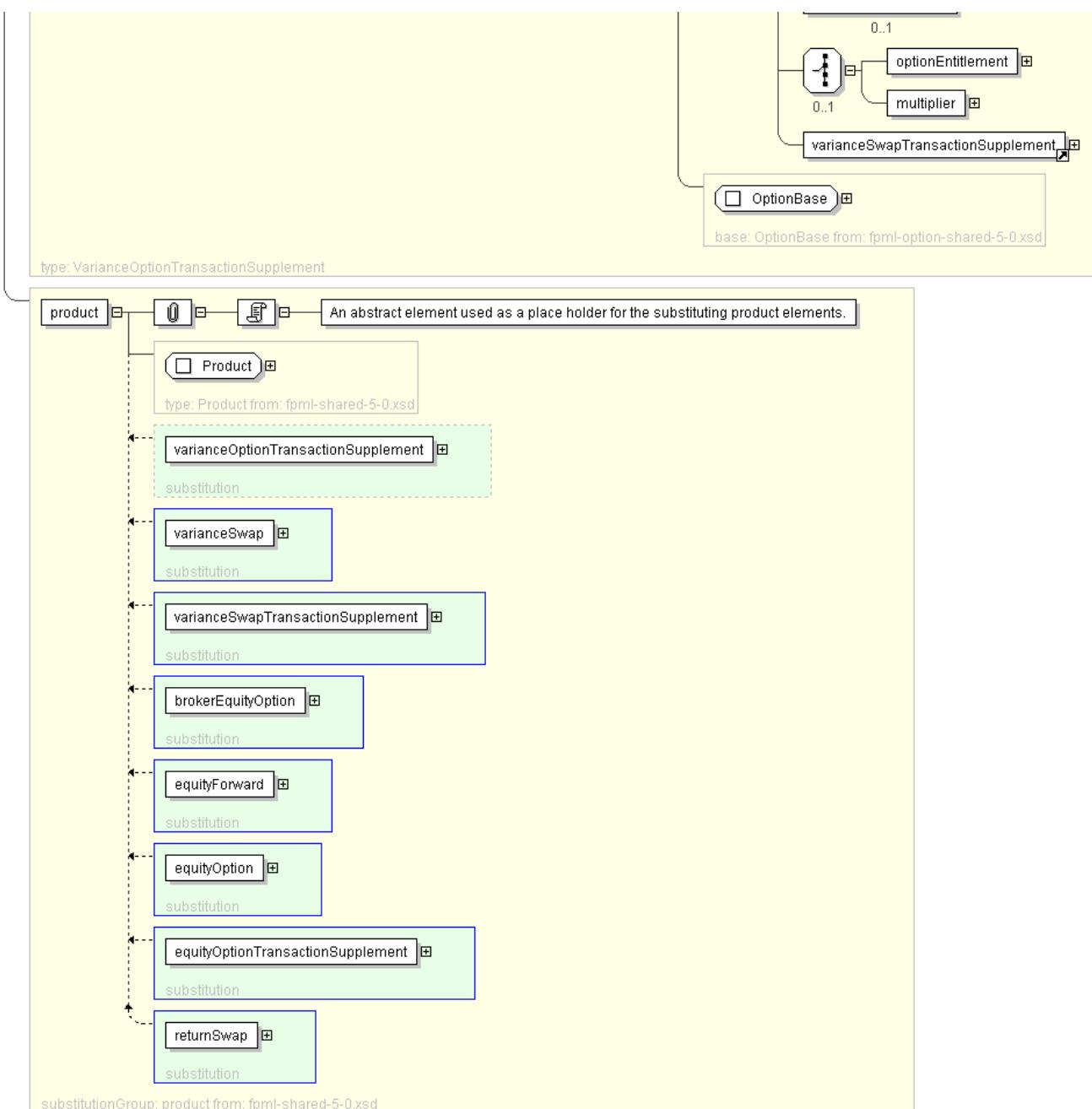
    <multiplier> PositiveDecimal </multiplier> [1]
      'Specifies the contract multiplier that can be associated with an index option.'

End Choice
<varianceSwapTransactionSupplement> ... </varianceSwapTransactionSupplement> [1]
'The variance swap details.'

</varianceOptionTransactionSupplement>

```

**Diagram**

**Schema Component Representation**

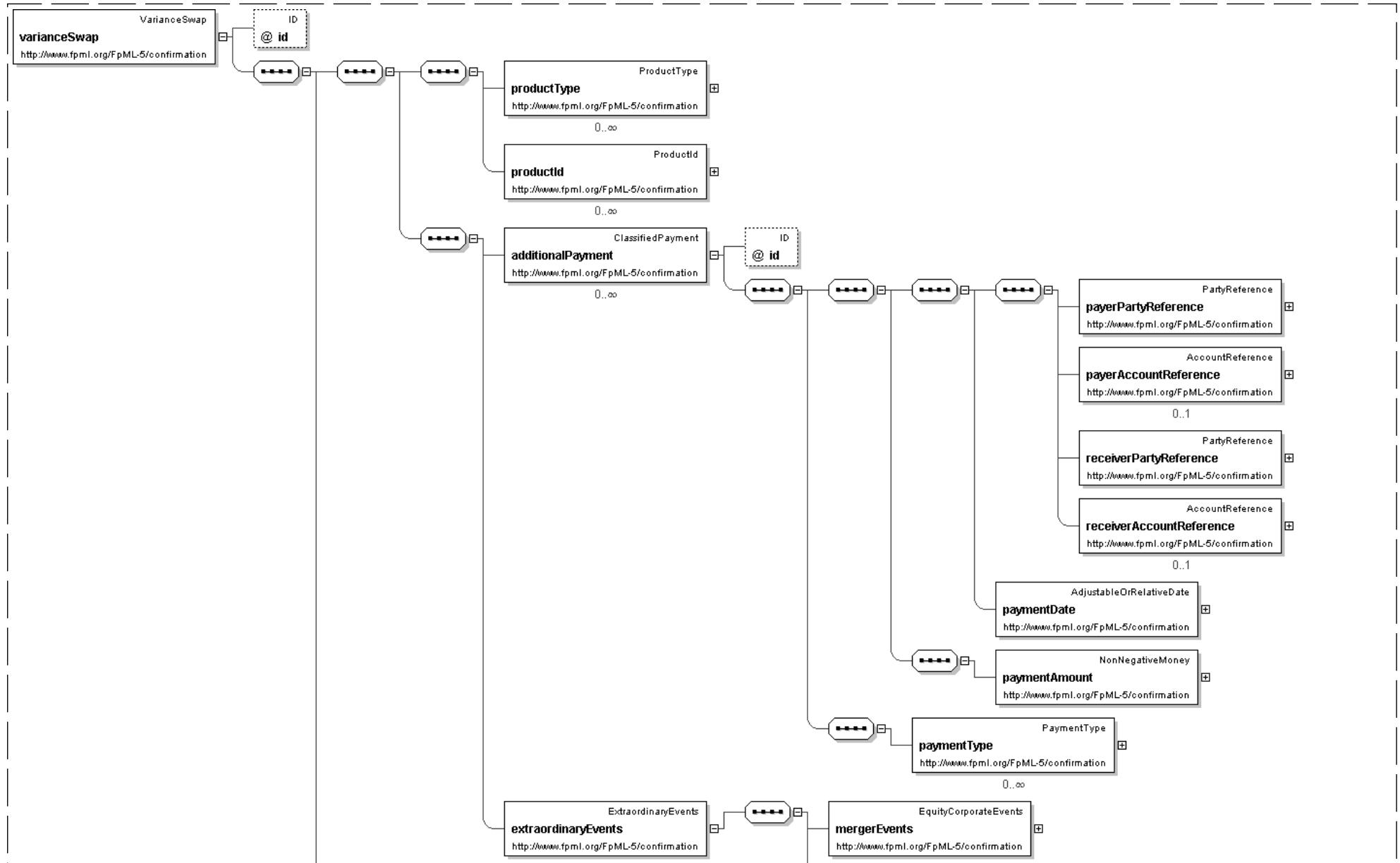
```

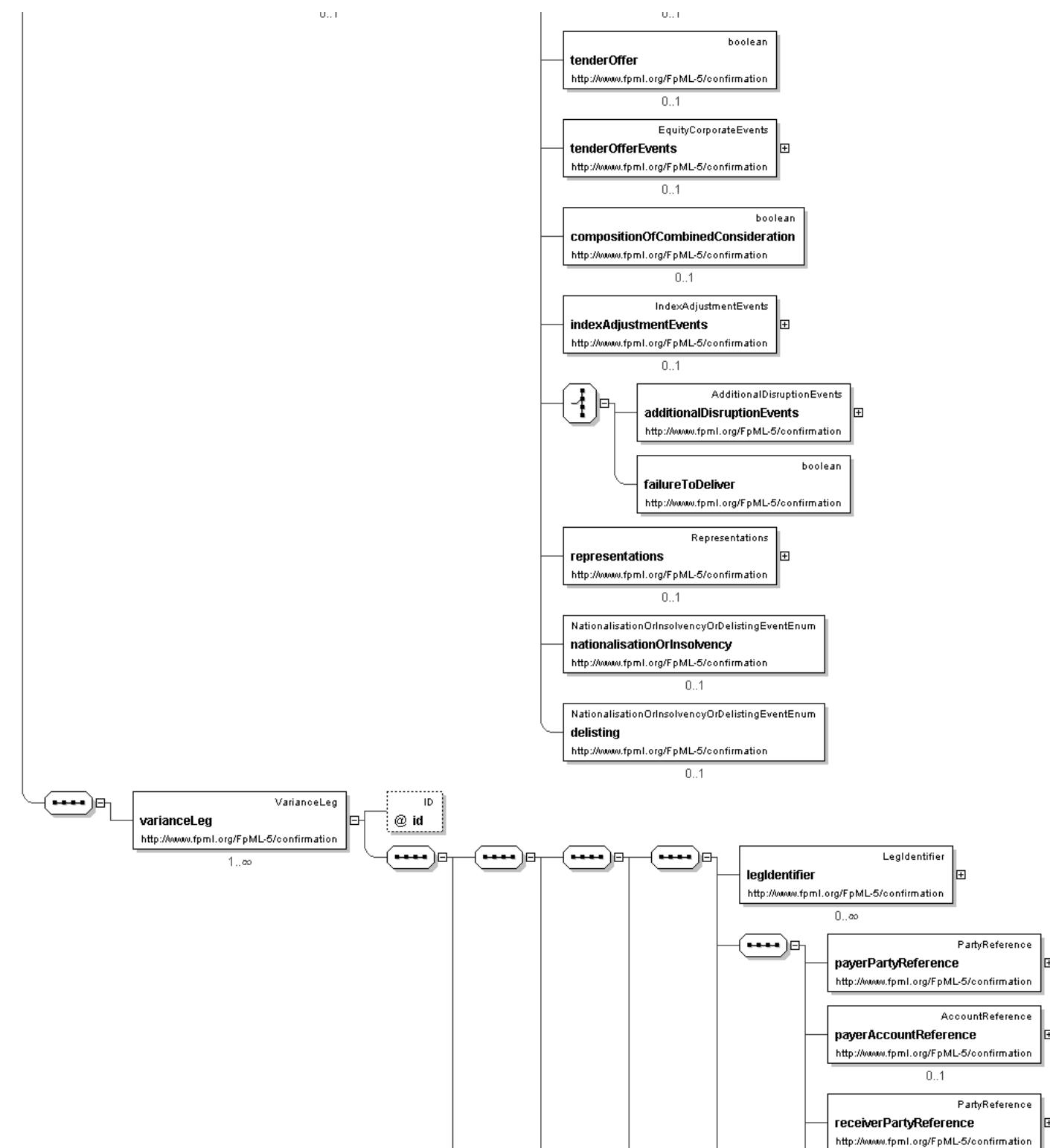
<xsd:element name="varianceOptionTransactionSupplement"
  type=" VarianceOptionTransactionSupplement " substitutionGroup="product" />
  
```

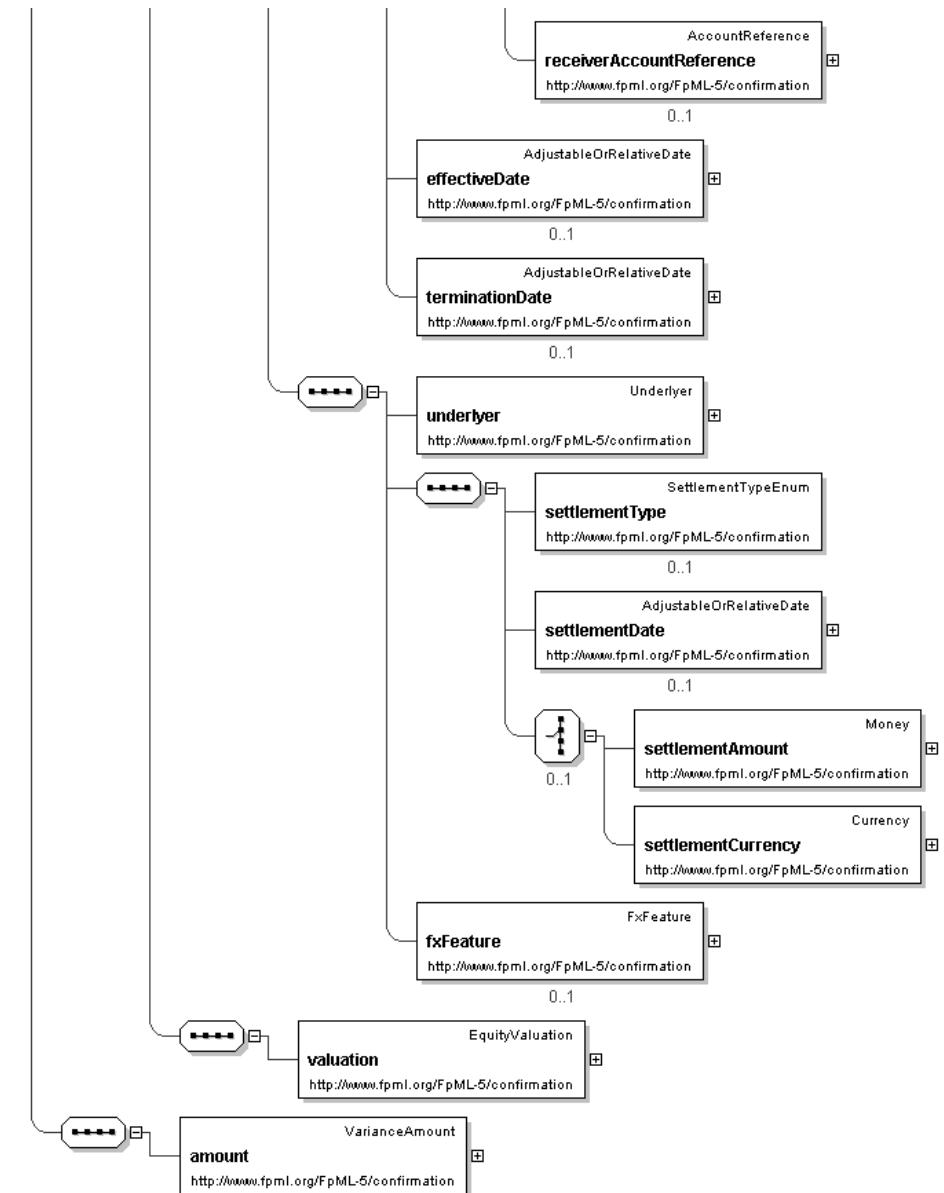
**Element: varianceSwap**

- This element can be used wherever the following element is referenced:
  - product

<b>Name</b>	varianceSwap
<b>Type</b>	VarianceSwap
<b>Nillable</b>	no
<b>Abstract</b>	no
<b>Documentation</b>	Specifies the structure of a variance swap.

**Logical Diagram**



**XML Instance Representation**

```

<varianceSwap
id=" xsd:ID [0..1]">
<productType> ProductType </productType> [0..*]
  'A classification of the type of product. FpML defines a simple product categorization using
  a coding scheme.'
<productId> ProductId </productId> [0..*]
  'A product 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.'

```

```

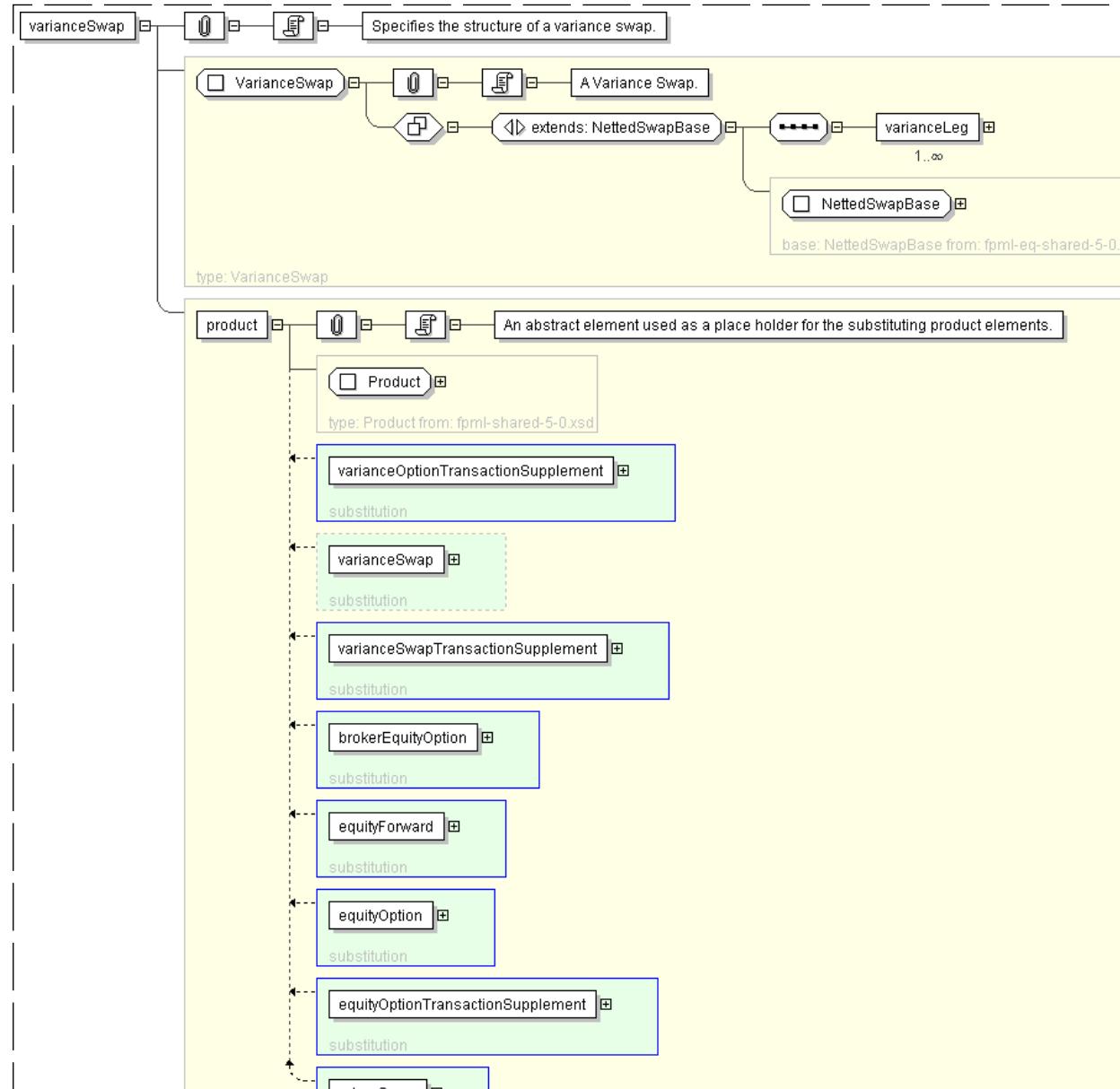
<additionalPayment> ClassifiedPayment </additionalPayment> [0..*]
  'Specifies additional payment(s) between the principal parties to the netted swap.'

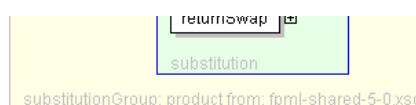
<extraordinaryEvents> ExtraordinaryEvents </extraordinaryEvents> [0..1]
  'Where the underlying is shares, specifies events affecting the issuer of those shares that
  may require the terms of the transaction to be adjusted.'

<varianceLeg> VarianceLeg </varianceLeg> [1..*]
  'Variance Leg.'

</varianceSwap>

```

**Diagram**

**Schema Component Representation**

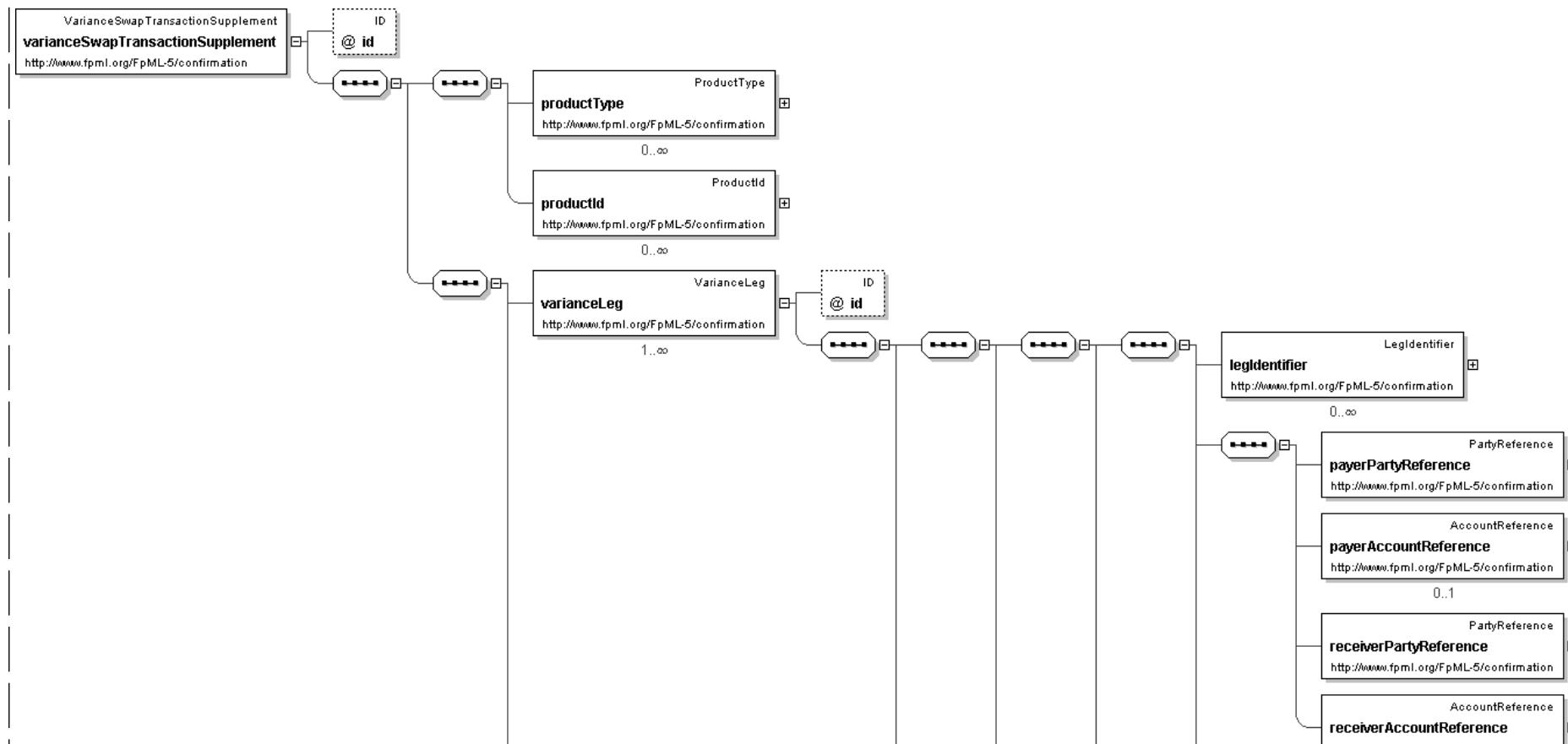
```
<xsd:element name="varianceSwap" type=" VarianceSwap " substitutionGroup="product" />
```

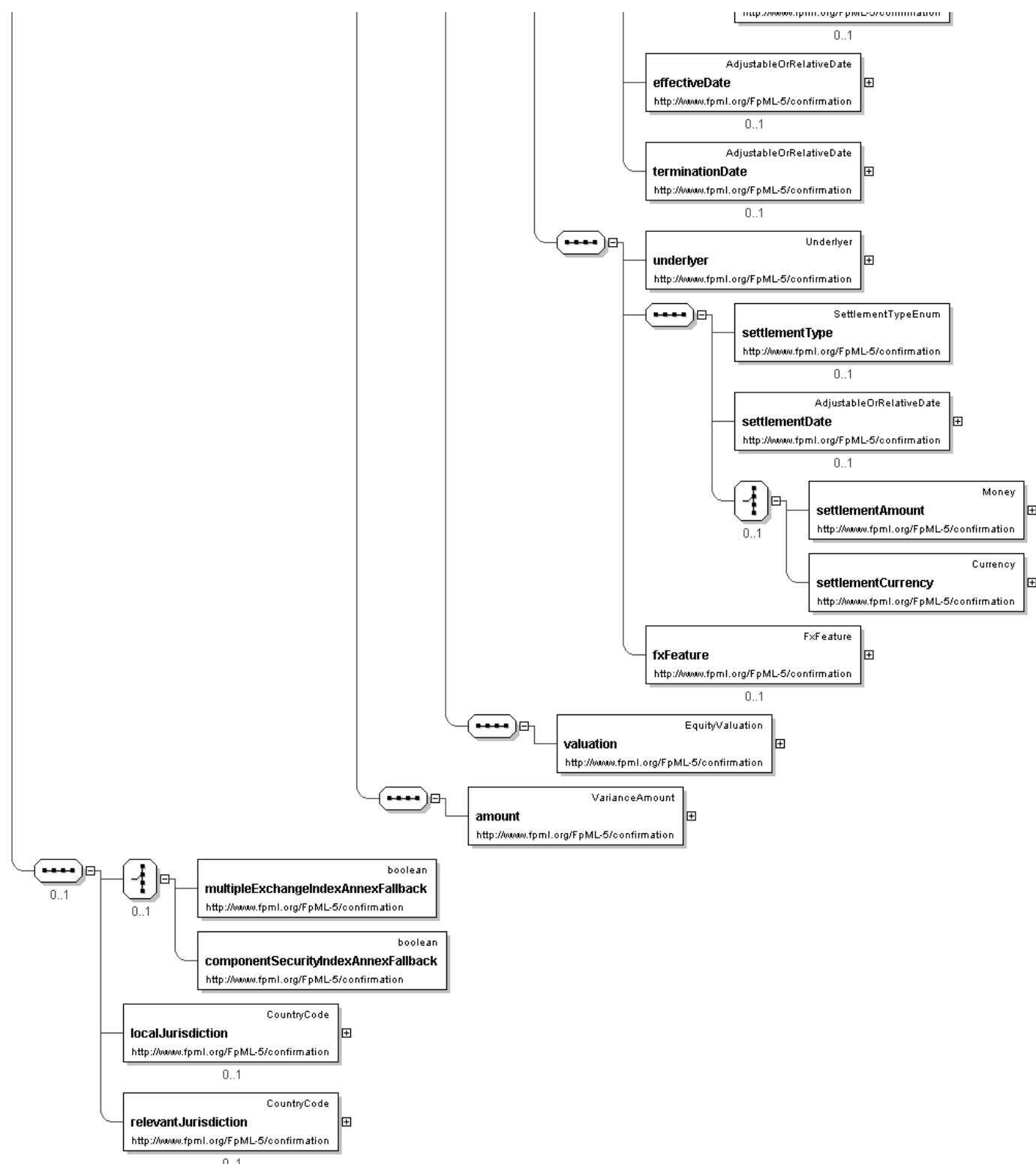
top

**Element: varianceSwapTransactionSupplement**

- This element can be used wherever the following element is referenced:
  - product

<b>Name</b>	varianceSwapTransactionSupplement
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">VarianceOptionTransactionSupplement</a>
<b>Type</b>	<a href="#">VarianceSwapTransactionSupplement</a>
<b>Nillable</b>	no
<b>Abstract</b>	no
<b>Documentation</b>	Specifies the structure of a variance swap transaction supplement.

**Logical Diagram**



**XML Instance Representation**

```

<varianceSwapTransactionSupplement
id=" xsd:ID [0..1]">
  <productType> ProductType </productType> [0..*]
    'A classification of the type of product. FpML defines a simple product categorization using
    a coding scheme.'

  <productId> ProductId </productId> [0..*]
    'A product 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.'

  <varianceLeg> VarianceLeg </varianceLeg> [1..*]
    'Variance Leg.'

Start Group: EquityUnderlyerProvisions.model [0..1]
Start Group: IndexAnnexFallback.model [0..1]
Start Choice [1]
  <multipleExchangeIndexAnnexFallback> xsd:boolean </multipleExchangeIndexAnnexFallback> [1]
    'For an index option transaction, a flag to indicate whether a relevant Multiple Exchange
    Index Annex is applicable to the transaction. This annex defines additional provisions
    which are applicable where an index is comprised of component securities that are traded
    on multiple exchanges.'

  <componentSecurityIndexAnnexFallback> xsd:boolean </componentSecurityIndexAnnexFallback> [1]
    'For an index option transaction, a flag to indicate whether a relevant Component
    Security Index Annex is applicable to the transaction.'

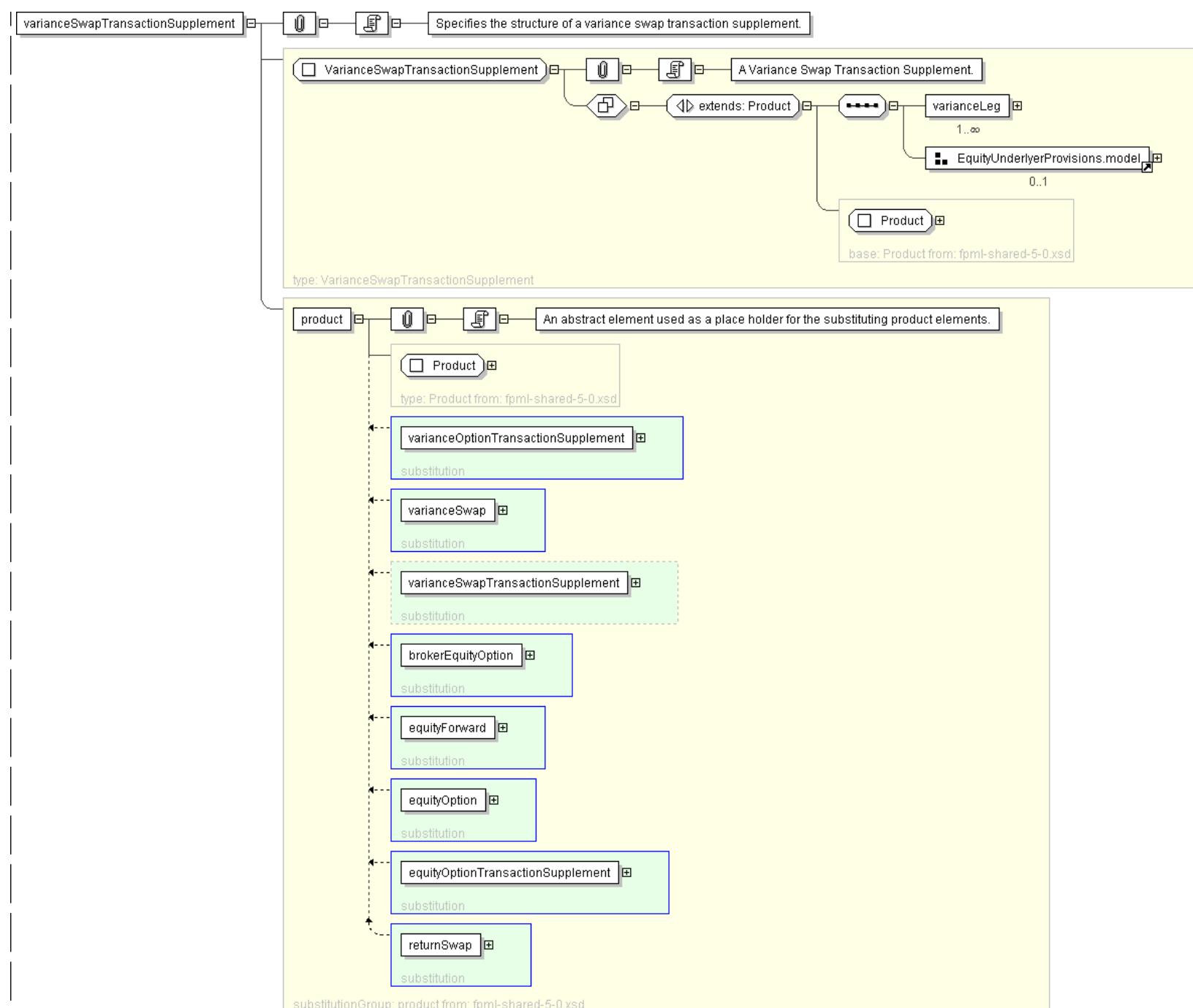
End Choice
End Group: IndexAnnexFallback.model
  <localJurisdiction> CountryCode </localJurisdiction> [0..1]
    'Local Jurisdiction is a term used in the AEJ Master Confirmation, which is used to
    determine local taxes, which shall mean taxes, duties, and similar charges imposed by
    the taxing authority of the Local Jurisdiction If this element is not present
    Local Jurisdiction is Not Applicable.'

  <relevantJurisdiction> CountryCode </relevantJurisdiction> [0..1]
    'Relevant Jurisdiction is a term used in the AEJ Master Confirmation, which is used
    to determine local taxes, which shall mean taxes, duties and similar charges that would
    be imposed by the taxing authority of the Country of Underlyer on a Hypothetical Broker
    Dealer assuming the Applicable Hedge Positions are held by its office in the
    Relevant Jurisdiction. If this element is not present Relevant Jurisdiction is Not Applicable.'

End Group: EquityUnderlyerProvisions.model
</varianceSwapTransactionSupplement>

```

**Diagram**

**Schema Component Representation**

```
<xsd:element name="varianceSwapTransactionSupplement" type="VarianceSwapTransactionSupplement" substitutionGroup="product"/>
```

## Global Definitions

### Complex Type: **VarianceAmount**

Super-types:	<a href="#">CalculatedAmount</a> < <b>VarianceAmount</b> (by extension)
Sub-types:	None

Name	VarianceAmount
Used by (from the same schema document)	Complex Type <a href="#">VarianceLeg</a>
Abstract	no
Documentation	Calculation of a Variance Amount.

#### XML Instance Representation

```

<...>
<calculationDates> AdjustableRelativeOrPeriodicDates </calculationDates> [0..1]
'Specifies the date on which a calculation or an observation will be performed for the
purpose of calculating the amount.'

<observationStartDate> AdjustableOrRelativeDate </observationStartDate> [0..1]
'The start of the period over which observations are made which are used in the
calculation Used when the observation start date differs from the trade date such as
for forward starting swaps.'

<optionsExchangeDividends> xsd:boolean </optionsExchangeDividends> [0..1]
'If present and true, then options exchange dividends are applicable.'

<additionalDividends> xsd:boolean </additionalDividends> [0..1]
'If present and true, then additional dividends are applicable.'

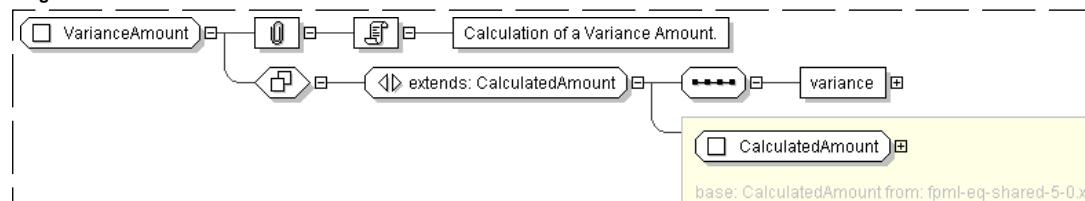
<allDividends> xsd:boolean </allDividends> [0..1]
'Represents the European Master Confirmation value of \'All Dividends\' which, when
applicable, signifies that, for a given Ex-Date, the daily observed Share Price for that day
is adjusted (reduced) by the cash dividend and/or the cash value of any non cash dividend
per Share (including Extraordinary Dividends) declared by the Issuer.'

<variance> Variance </variance> [1]
'Specifies Variance.'

</...>

```

#### Diagram



#### Schema Component Representation

```

<xsd:complexType name="VarianceAmount">
  <xsd:complexContent>
    <xsd:extension base=" CalculatedAmount ">
      <xsd:sequence>
        <xsd:element name="variance" type=" Variance " />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

**Complex Type: VarianceLeg**

**Super-types:** [DirectionalLegUnderlyerValuation](#) < **VarianceLeg** (by extension)

**Sub-types:** None

<b>Name</b>	VarianceLeg
<b>Used by (from the same schema document)</b>	Complex Type <a href="#">VarianceSwap</a> , Complex Type <a href="#">VarianceSwapTransactionSupplement</a>
<b>Abstract</b>	no
<b>Documentation</b>	A type describing return which is driven by a Variance Calculation.

**XML Instance Representation**

```

<...
  id=" xsd:ID [0..1]">
  <legIdentifier> LegIdentifier </legIdentifier> [0..*]
  'Version aware identification of this leg.'

  <payerPartyReference> PartyReference </payerPartyReference> [1]
  'A reference to the party responsible for making the payments defined by this structure.'

  <payerAccountReference> AccountReference </payerAccountReference> [0..1]
  'A reference to the account responsible for making the payments defined by this structure.'

  <receiverPartyReference> PartyReference </receiverPartyReference> [1]
  'A reference to the party that receives the payments corresponding to this structure.'

  <receiverAccountReference> AccountReference </receiverAccountReference> [0..1]
  'A reference to the account that receives the payments corresponding to this structure.'

  <effectiveDate> AdjustableOrRelativeDate </effectiveDate> [0..1]
  'Specifies the effective date of this leg of the swap. When defined in relation to a
  date specified somewhere else in the document (through the relativeDate component),
  this element will typically point to the effective date of the other leg of the swap.'

  <terminationDate> AdjustableOrRelativeDate </terminationDate> [0..1]
  'Specifies the termination date of this leg of the swap. When defined in relation to a
  date specified somewhere else in the document (through the relativeDate component),
  this element will typically point to the termination date of the other leg of the swap.'

  <underlyer> Underlyer </underlyer> [1]
  'Specifies the underlyer of the leg.'

  <settlementType> SettlementTypeEnum </settlementType> [0..1]
  <settlementDate> AdjustableOrRelativeDate </settlementDate> [0..1]
Start Group: SettlementAmountOrCurrency.model [0..1]
Start Choice [1]
  <settlementAmount> Money </settlementAmount> [1]
  'Settlement Amount'

  <settlementCurrency> Currency </settlementCurrency> [1]
  'Settlement Currency for use where the Settlement Amount cannot be known in advance'

End Choice
End Group: SettlementAmountOrCurrency.model
  <fxFeature> FxFeature </fxFeature> [0..1]
  'Quanto, Composite, or Cross Currency FX features.'

```

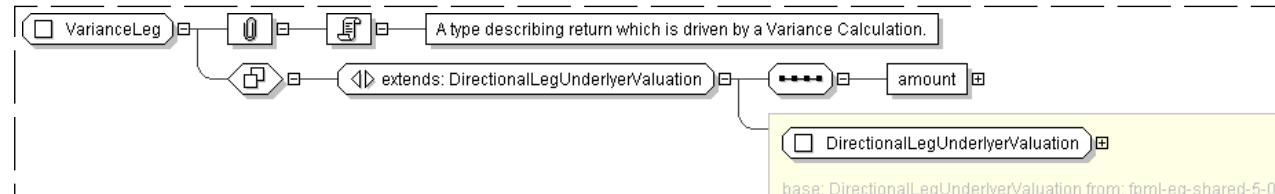
```

<valuation> EquityValuation </valuation> [1]
  'Valuation of the underlyer.'

<amount> VarianceAmount </amount> [1]
  'Specifies, in relation to each Equity Payment Date, the amount to which the Equity
  Payment Date relates. Unless otherwise specified, this term has the meaning defined in the
  ISDA 2002 Equity Derivatives Definitions.'

<...>

```

**Diagram****Schema Component Representation**

```

<xsd:complexType name="VarianceLeg">
  <xsd:complexContent>
    <xsd:extension base=" DirectionalLegUnderlyerValuation ">
      <xsd:sequence>
        <xsd:element name="amount" type=" VarianceAmount " />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

top

**Complex Type: VarianceOptionTransactionSupplement**

<b>Super-types:</b>	<a href="#">OptionBase</a> < <b>VarianceOptionTransactionSupplement</b> (by extension)
---------------------	--

<b>Sub-types:</b>	None
-------------------	------

<b>Name</b>	VarianceOptionTransactionSupplement
<b>Used by (from the same schema document)</b>	Element <a href="#">varianceOptionTransactionSupplement</a>
<b>Abstract</b>	no

**XML Instance Representation**

```

<...>
<id=" xsd:ID [0..1]">
<productType> ProductType </productType> [0..*]
  'A classification of the type of product. FpML defines a simple product categorization using
  a coding scheme.'

<productId> ProductId </productId> [0..*]
  'A product 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.'

<buyerPartyReference> PartyReference </buyerPartyReference> [1]
  'A reference to the party that buys this instrument, ie. pays for this instrument and
  receives the rights defined by it. See 2000 ISDA definitions Article 11.1 (b). In the case
  of FRAs this the fixed rate payer.'

<buyerAccountReference> AccountReference </buyerAccountReference> [0..1]
  'A reference to the account that buys this instrument.'

```

<sellerPartyReference> PartyReference </sellerPartyReference> [1]  
 'A reference to the party that sells ("writes") this instrument, i.e. that grants the rights defined by this instrument and in return receives a payment for it. See 2000 ISDA definitions Article 11.1 (a). In the case of FRAs this is the floating rate payer.'

<sellerAccountReference> AccountReference </sellerAccountReference> [0..1]  
 'A reference to the account that sells this instrument.'

<optionType> OptionTypeEnum </optionType> [1]

'The type of option transaction. From a usage standpoint, put/call is the default option type, while payer/receiver indicator is used for options index credit default swaps, consistently with the industry practice. Straddle is used for the case of straddle strategy, that combine a call and a put with the same strike.'

<equityPremium> EquityPremium </equityPremium> [1]

'The variance option premium payable by the buyer to the seller.'

<equityExercise> EquityExerciseValuationSettlement </equityExercise> [1]

'The parameters for defining how the equity option can be exercised, how it is valued and how it is settled.'

<exchangeLookAlike> xsd:boolean </exchangeLookAlike> [0..1]

'For a share option transaction, a flag used to indicate whether the transaction is to be treated as an '\exchange look-alike'. This designation has significance for how share adjustments (arising from corporate actions) will be determined for the transaction. For an '\exchange look-alike' transaction the relevant share adjustments will follow that for a corresponding designated contract listed on the related exchange (referred to as Options Exchange Adjustment (ISDA defined term)), otherwise the share adjustments will be determined by the calculation agent (referred to as Calculation Agent Adjustment (ISDA defined term)).'

<methodOfAdjustment> MethodOfAdjustmentEnum </methodOfAdjustment> [0..1]

'Defines how adjustments will be made to the contract should one or more of the extraordinary events occur.'

Start Choice [0..1]

<optionEntitlement> PositiveDecimal </optionEntitlement> [1]

'The number of shares per option comprised in the option transaction supplement.'

<multiplier> PositiveDecimal </multiplier> [1]

'Specifies the contract multiplier that can be associated with an index option.'

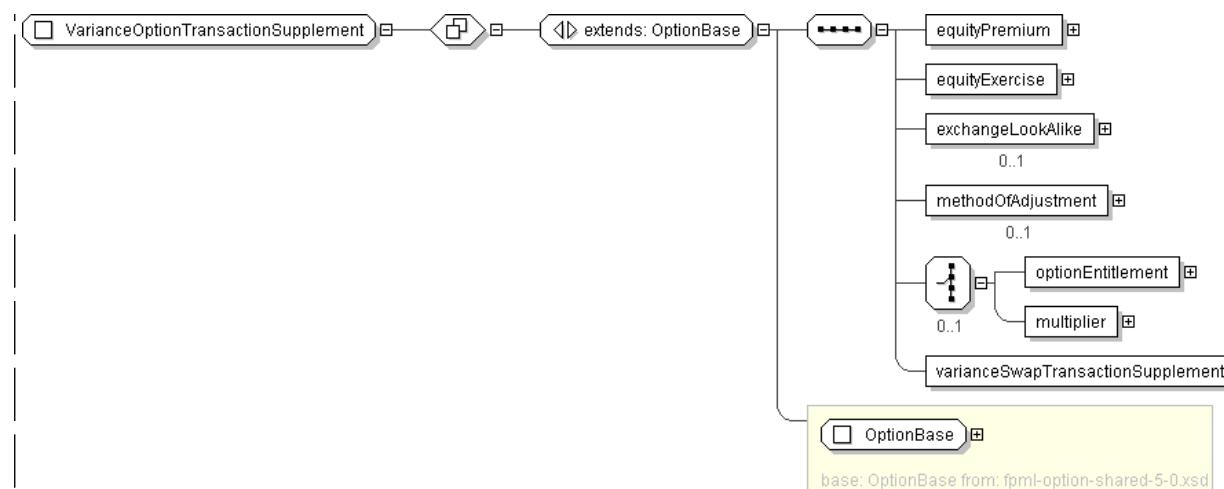
End Choice

<varianceSwapTransactionSupplement> ... </varianceSwapTransactionSupplement> [1]

'The variance swap details.'

</...>

Diagram

**Schema Component Representation**

```

<xsd:complexType name="VarianceOptionTransactionSupplement">
  <xsd:complexContent>
    <xsd:extension base=" OptionBase ">
      <xsd:sequence>
        <xsd:element name="equityPremium" type=" EquityPremium "/>
        <xsd:element name="equityExercise" type=" EquityExerciseValuationSettlement "/>
        <xsd:element name="exchangeLookalike" type=" xsd:boolean " minOccurs="0"/>
        <xsd:element name="methodOfAdjustment" type=" MethodOfAdjustmentEnum " minOccurs="0"/>
        <xsd:choice minOccurs="0">
          <xsd:element name="optionEntitlement" type=" PositiveDecimal "/>
          <xsd:element name="multiplier" type=" PositiveDecimal "/>
        </xsd:choice>
        <xsd:element ref=" varianceSwapTransactionSupplement "/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

[top](#)**Complex Type: [VarianceSwap](#)**

<b>Super-types:</b>	<a href="#">NettedSwapBase</a> < <b>VarianceSwap</b> (by extension)
<b>Sub-types:</b>	None

<b>Name</b>	VarianceSwap
<b>Used by (from the same schema document)</b>	Element <a href="#">varianceSwap</a>
<b>Abstract</b>	no
<b>Documentation</b>	A Variance Swap.

**XML Instance Representation**

```

<...
  id=" xsd:ID [0..1]">
  <productType> ProductType </productType> [0..*]
  'A classification of the type of product. FpML defines a simple product categorization using
  a coding scheme.'
  <productId> ProductId </productId> [0..*]
  'A product 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.'
```

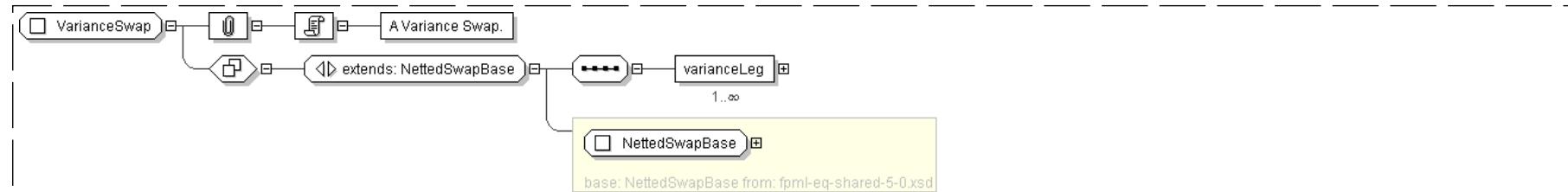
<additionalPayment> ClassifiedPayment </additionalPayment> [0..\*]  
'Specifies additional payment(s) between the principal parties to the netted swap.'

<extraordinaryEvents> ExtraordinaryEvents </extraordinaryEvents> [0..1]  
'Where the underlying is shares, specifies events affecting the issuer of those shares that may require the terms of the transaction to be adjusted.'

<varianceLeg> VarianceLeg </varianceLeg> [1..\*]  
'Variance Leg.'

...>

## Diagram



## Schema Component Representation

```
<xsd:complexType name="VarianceSwap">
  <xsd:complexContent>
    <xsd:extension base="NettedSwapBase">
      <xsd:sequence>
        <xsd:element name="varianceLeg" type="VarianceLeg" maxOccurs="unbounded">
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

top

**Complex Type:** [VarianceSwapTransactionSupplement](#)

<i>Super-types:</i>	<a href="#">Product</a> < <b>VarianceSwapTransactionSupplement</b> (by extension)
<i>Sub-types:</i>	None
<b>Name</b>	VarianceSwapTransactionSupplement
<b>Used by (from the same schema document)</b>	Element <a href="#">varianceSwapTransactionSupplement</a>
<b>Abstract</b>	no
<b>Documentation</b>	A Variance Swan Transaction Supplement

## XML Instance Representation

```
<...  
id=" xsd:ID [0..1]">  
<productType> ProductType </productType> [0..*]  
'A classification of the type of product. FpML defines a simple product categorization using a coding scheme.'  
  
<productId> ProductId </productId> [0..*]  
'A product 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.'  
  
<varianceLeg> VarianceLeg </varianceLeg> [1 .. *]
```

## 'Variance Leg.'

Start Group: [EquityUnderlyerProvisions.model](#) [0..1]  
 Start Group: [IndexAnnexFallback.model](#) [0..1]  
 Start Choice [1]

- <multipleExchangeIndexAnnexFallback> xsd:boolean </multipleExchangeIndexAnnexFallback> [1]
 

'For an index option transaction, a flag to indicate whether a relevant Multiple Exchange Index Annex is applicable to the transaction. This annex defines additional provisions which are applicable where an index is comprised of component securities that are traded on multiple exchanges.'
- <componentSecurityIndexAnnexFallback> xsd:boolean </componentSecurityIndexAnnexFallback> [1]
 

'For an index option transaction, a flag to indicate whether a relevant Component Security Index Annex is applicable to the transaction.'

End Choice  
 End Group: [IndexAnnexFallback.model](#)  
 <localJurisdiction> [CountryCode](#) </localJurisdiction> [0..1]
 

'Local Jurisdiction is a term used in the AEJ Master Confirmation, which is used to determine local taxes, which shall mean taxes, duties, and similar charges imposed by the taxing authority of the Local Jurisdiction If this element is not present Local Jurisdiction is Not Applicable.'

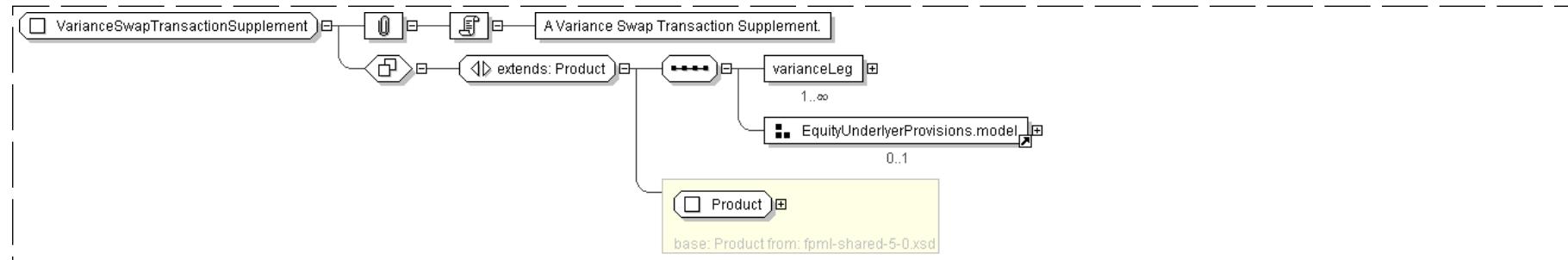
<relevantJurisdiction> [CountryCode](#) </relevantJurisdiction> [0..1]
 

'Relevant Jurisdiction is a term used in the AEJ Master Confirmation, which is used to determine local taxes, which shall mean taxes, duties and similar charges that would be imposed by the taxing authority of the Country of Underlyer on a Hypothetical Broker Dealer assuming the Applicable Hedge Positions are held by its office in the Relevant Jurisdiction. If this element is not present Relevant Jurisdiction is Not Applicable.'

End Group: [EquityUnderlyerProvisions.model](#)

&lt;/...&gt;

## Diagram



## Schema Component Representation

```

<xsd:complexType name="VarianceSwapTransactionSupplement">
  <xsd:complexContent>
    <xsd:extension base=" Product ">
      <xsd:sequence>
        <xsd:element name="varianceLeg" type=" VarianceLeg " maxOccurs="unbounded"/>
        <xsd:group ref=" EquityUnderlyerProvisions.model " minOccurs="0"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
  
```

## Legend

**Complex Type:**  
Schema Component Type**AusAddress**  
Schema Component Name

<b>Super-types:</b>	<a href="#">Address</a> < AusAddress (by extension)
<b>Sub-types:</b>	• <a href="#">OLDAddress</a> (by restriction)

If this schema component is a type definition, its type hierarchy is shown in a gray-bordered box.

<b>Name</b>	AusAddress
<b>Abstract</b>	no

The table above displays the properties of this schema component.

**XML Instance Representation**

```
<... country="Australia" >
<unitNo> string </unitNo> [0..1]
<houseNo> string </houseNo> [1]
<street> string </street> [1]
Start Choice [1]
<city> string </city> [1]
<town> string </town> [1]
End Choice
<state> AusStates </state> [1]
<postcode> string <>pattern = [1-9][0-9]{3}<> </postcode> [1]
</...>
```

The XML Instance Representation table above shows the schema component's content as an XML instance.

- The minimum and maximum occurrence of elements and attributes are provided in square brackets, e.g. [0..1].
- Model group information are shown in gray, e.g. Start Choice ... End Choice.
- For type derivations, the elements and attributes that have been added to or changed from the base type's content are shown in **bold**.
- If an element/attribute has a fixed value, the fixed value is shown in green, e.g. country="Australia".
- Otherwise, the type of the element/attribute is displayed.
  - If the element/attribute's type is in the schema, a link is provided to it.
  - For local simple type definitions, the constraints are displayed in angle brackets, e.g. <>pattern = [1-9][0-9]{3}<>.

**Schema Component Representation**

```
<complexType name="AusAddress">
<complexContent>
<extension base=" Address " >
<sequence>
<element name="state" type=" AusStates " />
<element name="postcode">
<simpleType>
<restriction base=" string " >
<pattern value="[1-9][0-9]{3}" />
</restriction>
</simpleType>
</element>
</sequence>
<attribute name="country" type=" string " fixed="Australia" />
</extension>
</complexContent>
</complexType>
```

The Schema Component Representation table above displays the underlying XML representation of the schema component. (Annotations are not shown.)

**Glossary**

**Abstract** (Applies to complex type definitions and element declarations). An abstract element or complex type cannot be used to validate an element

instance. If there is a reference to an abstract element, only element declarations that can substitute the abstract element can be used to validate the instance. For references to abstract type definitions, only derived types can be used.

**All Model Group** Child elements can be provided *in any order* in instances. See: <http://www.w3.org/TR/xmlschema-1/#element-all>.

**Choice Model Group** *Only one* from the list of child elements and model groups can be provided in instances. See: <http://www.w3.org/TR/xmlschema-1/#element-choice>.

**Collapse Whitespace Policy** Replace tab, line feed, and carriage return characters with space character (Unicode character 32). Then, collapse contiguous sequences of space characters into single space character, and remove leading and trailing space characters.

**Disallowed Substitutions** (Applies to element declarations). If *substitution* is specified, then *substitution group* members cannot be used in place of the given element declaration to validate element instances. If *derivation methods*, e.g. extension, restriction, are specified, then the given element declaration will not validate element instances that have types derived from the element declaration's type using the specified derivation methods. Normally, element instances can override their declaration's type by specifying an *xsi:type* attribute.

**Key Constraint** Like [Uniqueness Constraint](#), but additionally requires that the specified value(s) must be provided. See: [http://www.w3.org/TR/xmlschema-1/#clentity-constraint\\_Definitions](http://www.w3.org/TR/xmlschema-1/#clentity-constraint_Definitions).

**Key Reference Constraint** Ensures that the specified value(s) must match value(s) from a [Key Constraint](#) or [Uniqueness Constraint](#). See: [http://www.w3.org/TR/xmlschema-1/#clentity-constraint\\_Definitions](http://www.w3.org/TR/xmlschema-1/#clentity-constraint_Definitions).

**Model Group** Groups together element content, specifying the order in which the element content can occur and the number of times the group of element content may be repeated. See: [http://www.w3.org/TR/xmlschema-1/#Model\\_Groups](http://www.w3.org/TR/xmlschema-1/#Model_Groups).

**Nillable** (Applies to element declarations). If an element declaration is nillable, instances can use the *xsi:nil* attribute. The *xsi:nil* attribute is the boolean attribute, *nil*, from the [http://www.w3.org/2001/XMLSchema-instance](#) namespace. If an element instance has an *xsi:nil* attribute set to true, it can be left empty, even though its element declaration may have required content.

**Notation** A notation is used to identify the format of a piece of data. Values of elements and attributes that are of type, NOTATION, must come from the names of declared notations. See: [http://www.w3.org/TR/xmlschema-1/#cNotation\\_Declarations](http://www.w3.org/TR/xmlschema-1/#cNotation_Declarations).

**Preserve Whitespace Policy** Preserve whitespaces exactly as they appear in instances.

**Prohibited Derivations** (Applies to type definitions). Derivation methods that cannot be used to create sub-types from a given type definition.

**Prohibited Substitutions** (Applies to complex type definitions). Prevents sub-types that have been derived using the specified derivation methods from validating element instances in place of the given type definition.

**Replace Whitespace Policy** Replace tab, line feed, and carriage return characters with space character (Unicode character 32).

**Sequence Model Group** Child elements and model groups must be provided *in the specified order* in instances. See: <http://www.w3.org/TR/xmlschema-1/#element-sequence>.

**Substitution Group** Elements that are *members* of a substitution group can be used wherever the *head* element of the substitution group is referenced.

**Substitution Group Exclusions** (Applies to element declarations). Prohibits element declarations from nominating themselves as being able to substitute a given element declaration, if they have types that are derived from the original element's type using the specified derivation methods.

**Target Namespace** The target namespace identifies the namespace that components in this schema belongs to. If no target namespace is provided, then the schema components do not belong to any namespace.

**Uniqueness Constraint** Ensures uniqueness of an element/attribute value, or a combination of values, within a specified scope. See: [http://www.w3.org/TR/xmlschema-1/#clentity-constraint\\_Definitions](http://www.w3.org/TR/xmlschema-1/#clentity-constraint_Definitions).

top

# XML Schema Documentation

---

## Table of Contents

- [Schema Document Properties](#)
- [Global Declarations](#)
  - [Element: CanonicalizationMethod](#)
  - [Element: DSAKeyValue](#)
  - [Element: DigestMethod](#)
  - [Element: DigestValue](#)
  - [Element: KeyInfo](#)
  - [Element:KeyName](#)
  - [Element: KeyValue](#)
  - [Element: Manifest](#)
  - [Element: MgmtData](#)
  - [Element: Object](#)
  - [Element: PGPData](#)
  - [Element: RSAKeyValue](#)
  - [Element: Reference](#)
  - [Element: RetrievalMethod](#)
  - [Element: SPKIData](#)
  - [Element: Signature](#)
  - [Element: SignatureMethod](#)
  - [Element: SignatureProperties](#)
  - [Element: SignatureProperty](#)
  - [Element: SignatureValue](#)
  - [Element: SignedInfo](#)
  - [Element: Transform](#)
  - [Element: Transforms](#)
  - [Element: X509Data](#)
- [Global Definitions](#)
  - [Complex Type: CanonicalizationMethodType](#)
  - [Complex Type: DSAKeyValueType](#)
  - [Complex Type: DigestMethodType](#)
  - [Complex Type: KeyInfoType](#)
  - [Complex Type: KeyValue](#)
  - [Complex Type: ManifestType](#)
  - [Complex Type: ObjectType](#)
  - [Complex Type: PGPDataType](#)

- o [Complex Type: RSAKeyValue](#)
- o [Complex Type: ReferenceType](#)
- o [Complex Type: RetrievalMethodType](#)
- o [Complex Type: SPKIDataType](#)
- o [Complex Type: SignatureMethodType](#)
- o [Complex Type: SignaturePropertiesType](#)
- o [Complex Type: SignaturePropertyType](#)
- o [Complex Type: SignatureType](#)
- o [Complex Type: SignatureValueType](#)
- o [Complex Type: SignedInfoType](#)
- o [Complex Type: TransformType](#)
- o [Complex Type: TransformsType](#)
- o [Complex Type: X509DataType](#)
- o [Complex Type: X509IssuerSerialType](#)
- o [Simple Type: CryptoBinary](#)
- o [Simple Type: DigestValueType](#)
- o [Simple Type: HMACOutputLengthType](#)

- [Legend](#)
- [Glossary](#)

[top](#)

## Schema Document Properties

<b>Target Namespace</b>	<a href="http://www.w3.org/2000/09/xmldsig#">http://www.w3.org/2000/09/xmldsig#</a>
<b>Version</b>	0.1
<b>Element and Attribute Namespaces</b>	<ul style="list-style-type: none"> <li>• Global element and attribute declarations belong to this schema's target namespace.</li> <li>• By default, local element declarations belong to this schema's target namespace.</li> <li>• By default, local attribute declarations have no namespace.</li> </ul>

## Declared Namespaces

Prefix	Namespace
Default namespace	<a href="http://www.w3.org/2001/XMLSchema">http://www.w3.org/2001/XMLSchema</a>
xml	<a href="http://www.w3.org/XML/1998/namespace">http://www.w3.org/XML/1998/namespace</a>
ds	<a href="http://www.w3.org/2000/09/xmldsig#">http://www.w3.org/2000/09/xmldsig#</a>

## Schema Component Representation

```
<schema targetNamespace="http://www.w3.org/2000/09/xmldsig#" elementFormDefault="qualified" version="0.1">
```

...

</schema>

## Global Declarations

### Element: CanonicalizationMethod

<b>Name</b>	CanonicalizationMethod
<b>Type</b>	<a href="#">ds:CanonicalizationMethodType</a>
<b>Nillable</b>	no
<b>Abstract</b>	no

#### Logical Diagram

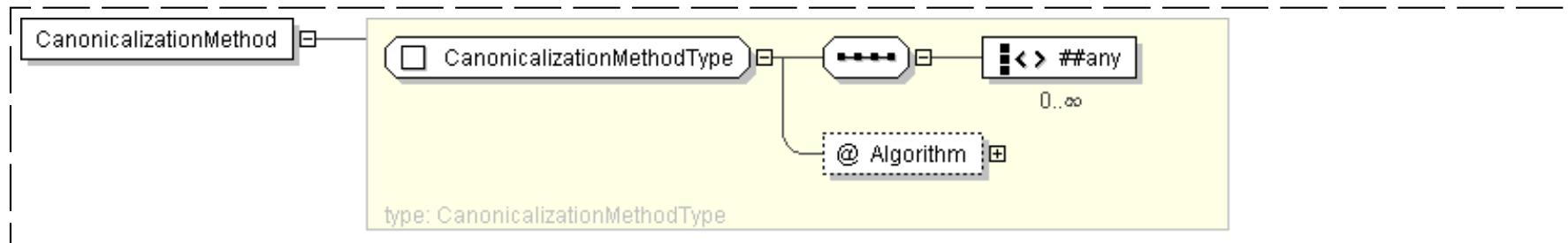


#### XML Instance Representation

```

<ds:CanonicalizationMethod
  Algorithm="anyURI [1]">
  <!-- Mixed content -->
  Allow any elements from any namespace (strict validation). [0..*]
</ds:CanonicalizationMethod>
  
```

#### Diagram



#### Schema Component Representation

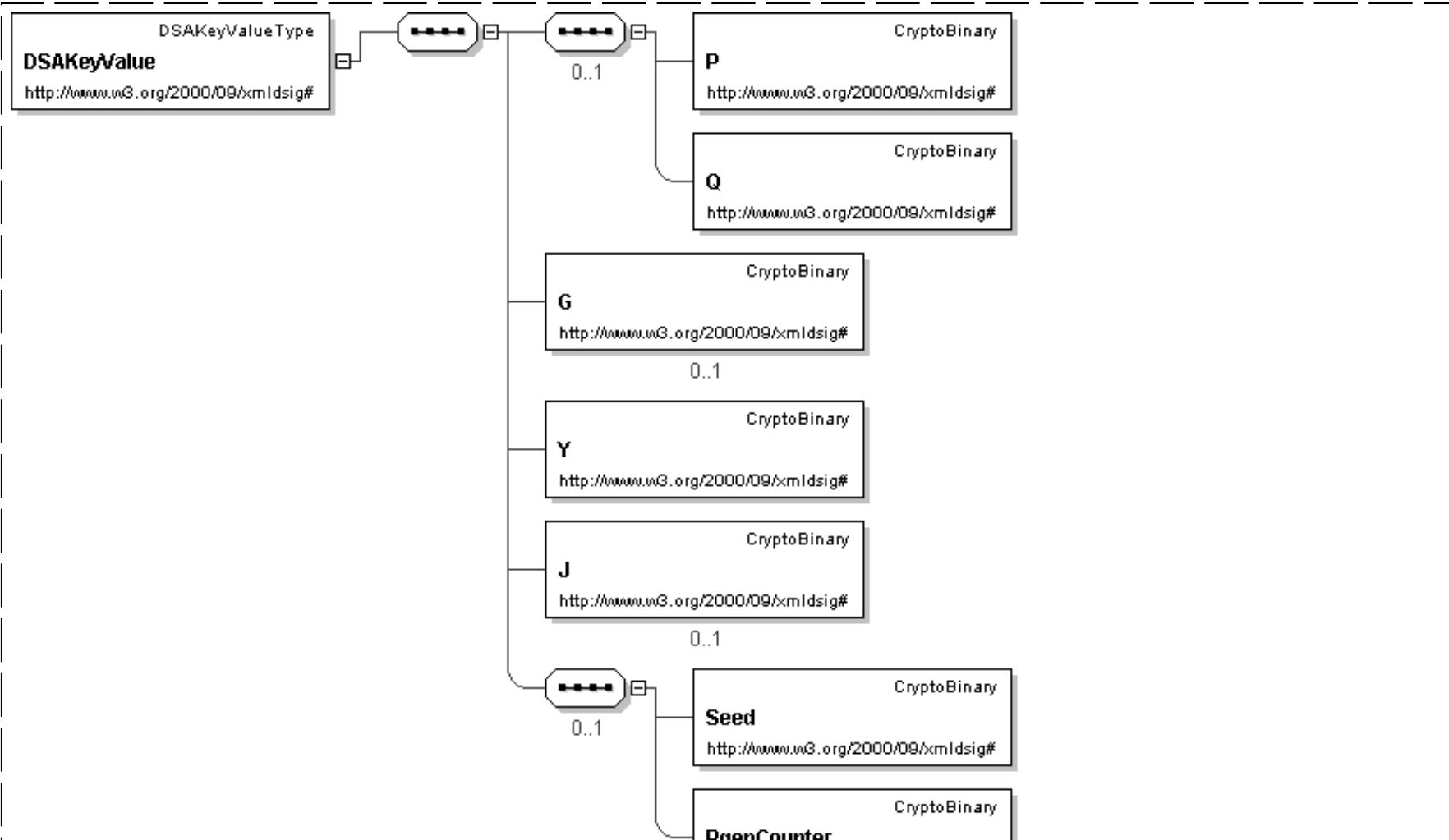
```
<element name="CanonicalizationMethod" type="ds:CanonicalizationMethodType" />
```

[top](#)

## Element: DSAKeyValue

<b>Name</b>	DSAKeyValue
<b>Type</b>	<a href="#">ds:DSAKeyValueType</a>
<b>Nillable</b>	no
<b>Abstract</b>	no

### Logical Diagram



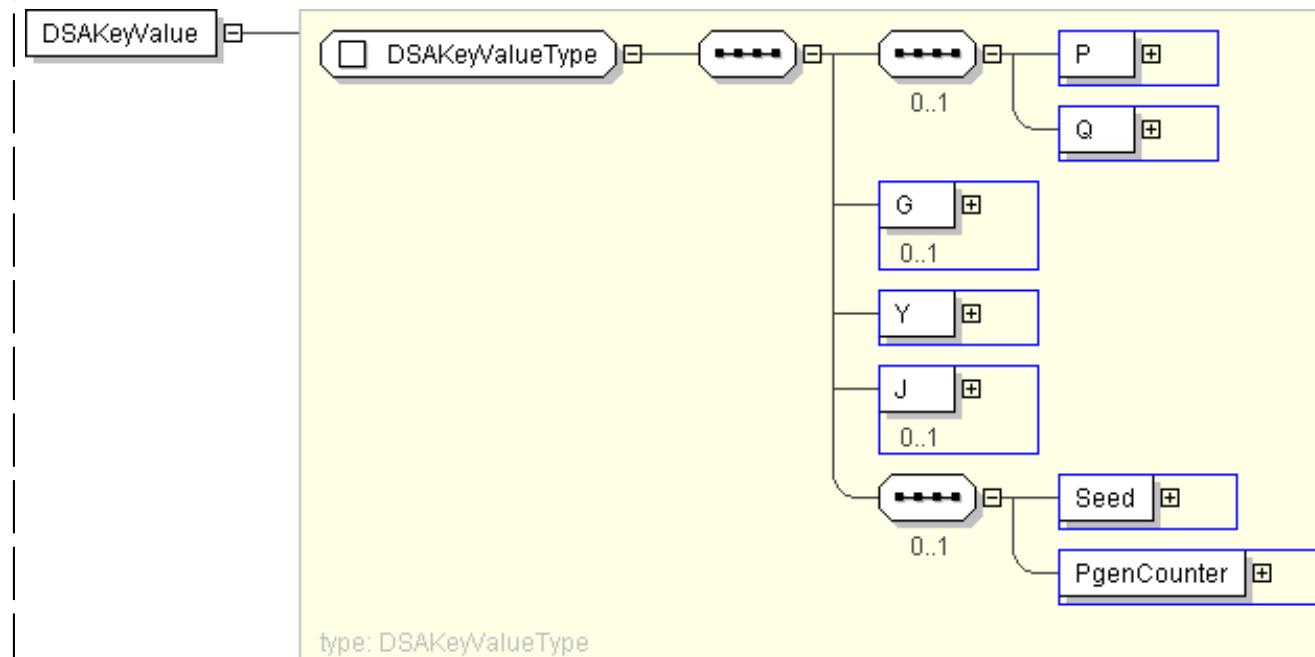
## XML Instance Representation

```

<ds:DSAKeyValue>
  Start Sequence [0..1]
    <ds:P> ds: CryptoBinary </ds:P> [1]
    <ds:Q> ds: CryptoBinary </ds:Q> [1]
  End Sequence
  <ds:G> ds: CryptoBinary </ds:G> [0..1]
  <ds:Y> ds: CryptoBinary </ds:Y> [1]
  <ds:J> ds: CryptoBinary </ds:J> [0..1]
  Start Sequence [0..1]
    <ds:Seed> ds: CryptoBinary </ds:Seed> [1]
    <ds:PgenCounter> ds: CryptoBinary </ds:PgenCounter> [1]
  End Sequence
</ds:DSAKeyValue>

```

## Diagram



## Schema Component Representation

```
<element name="DSAKeyValue" type=" ds:DSAKeyValueType " />
```

## Element: DigestMethod

<b>Name</b>	DigestMethod
<b>Type</b>	<a href="#">ds:DigestMethodType</a>
<b>Nillable</b>	no
<b>Abstract</b>	no

### Logical Diagram

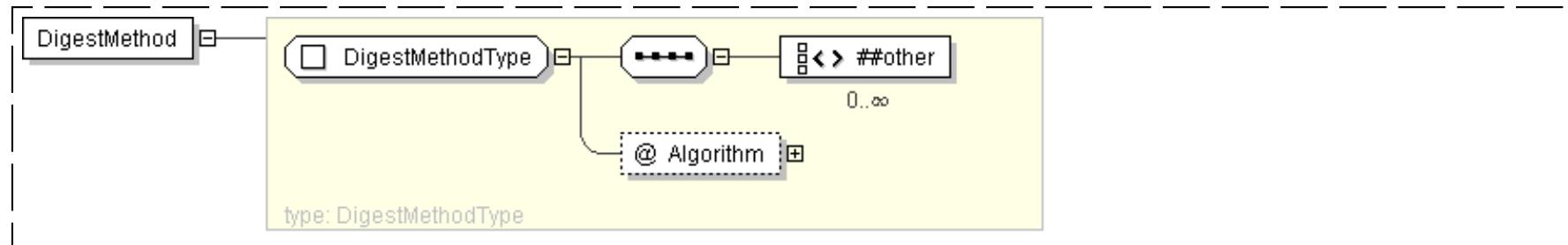


### XML Instance Representation

```

<ds:DigestMethod
Algorithm="anyURI [1]">
<!-- Mixed content -->
  Allow any elements from a namespace other than this schema's namespace (lax validation). [0..*]
</ds:DigestMethod>
  
```

### Diagram



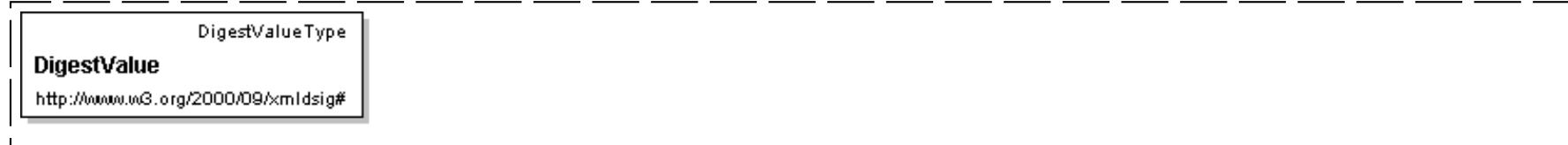
### Schema Component Representation

```
<element name="DigestMethod" type=" ds:DigestMethodType "/>
```

## Element: DigestValue

<b>Name</b>	DigestValue
<b>Type</b>	<a href="#">ds:DigestValueType</a>
<b>Nillable</b>	no
<b>Abstract</b>	no

### Logical Diagram



### XML Instance Representation

```
<ds:DigestValue> ds:DigestValueType </ds:DigestValue>
```

### Diagram



### Schema Component Representation

```
<element name="DigestValue" type=" ds:DigestValueType " />
```

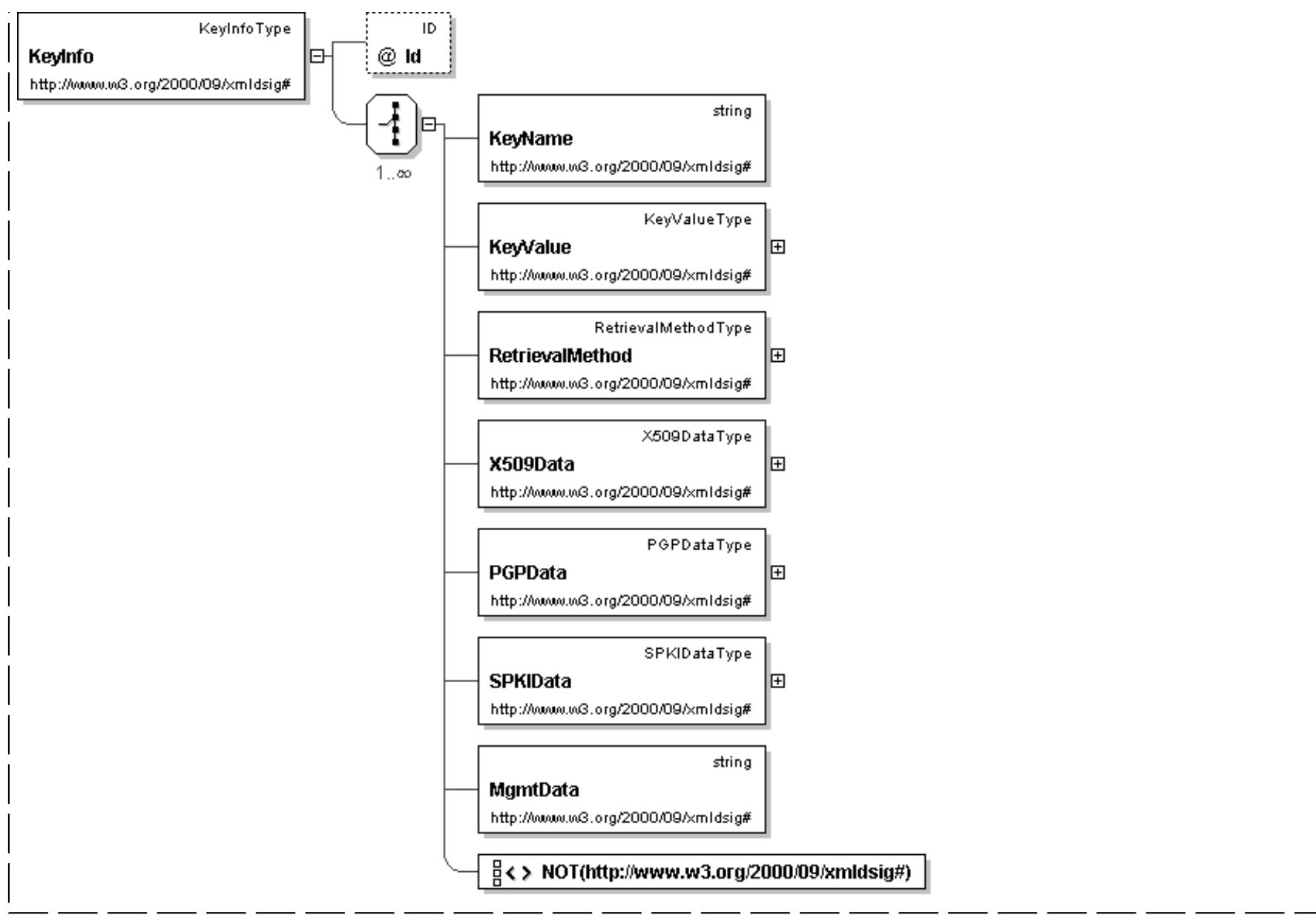
[top](#)

## Element: KeyInfo

<b>Name</b>	KeyInfo
<b>Type</b>	<a href="#">ds:KeyInfoType</a>
<b>Nillable</b>	no
<b>Abstract</b>	no

### Logical Diagram





### XML Instance Representation

```

<ds:KeyInfo
  Id="ID [0..1]">
  <!-- Mixed content -->
  Start Choice [1..*]
    <ds:KeyName> ... </ds:KeyName> [1]
    <ds:KeyValue> ... </ds:KeyValue> [1]
  
```

```

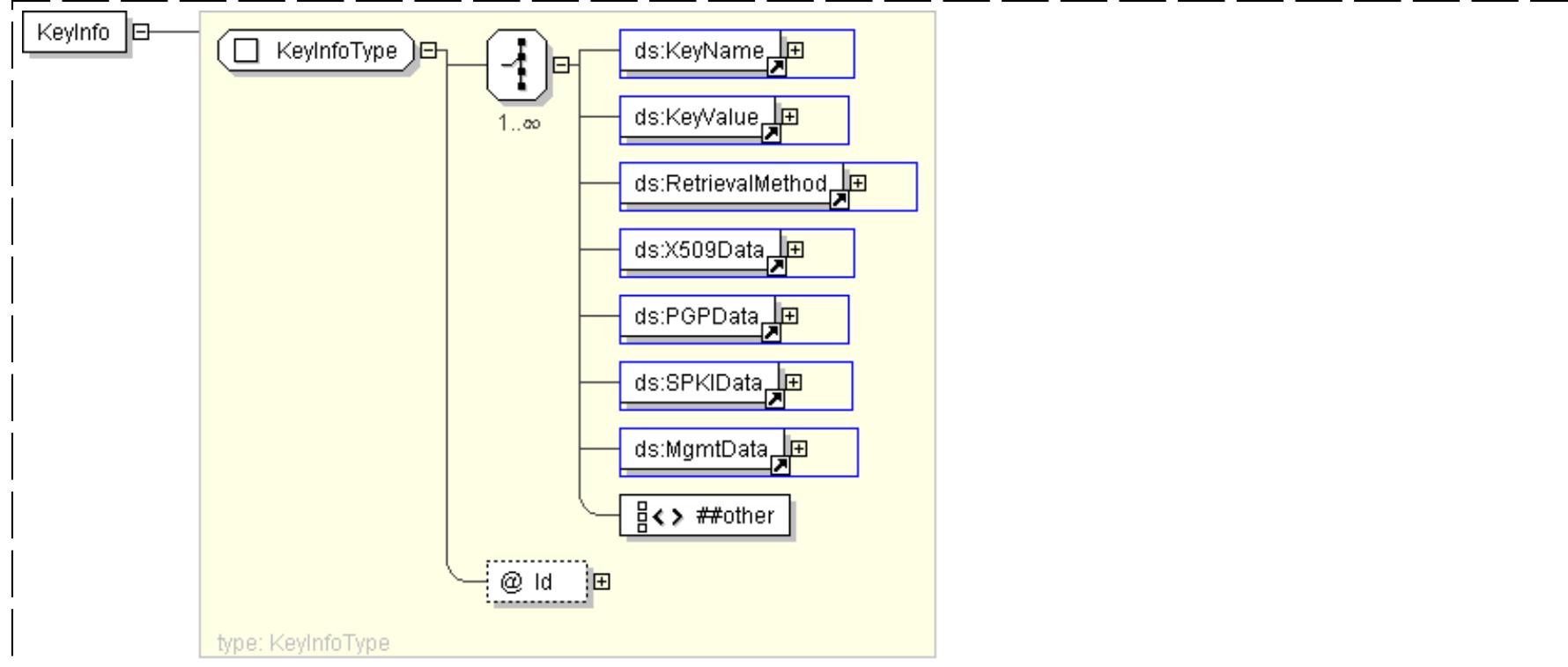
<ds:RetrievalMethod> ... </ds:RetrievalMethod> [1]
<ds:X509Data> ... </ds:X509Data> [1]
<ds:PGPData> ... </ds:PGPData> [1]
<ds:SPKIData> ... </ds:SPKIData> [1]
<ds:MgmtData> ... </ds:MgmtData> [1]

Allow any elements from a namespace other than this schema's namespace (lax validation). [1]

```

End Choice

&lt;/ds:KeyInfo&gt;

**Diagram****Schema Component Representation**

```
<element name="KeyInfo" type="ds:KeyInfoType" />
```

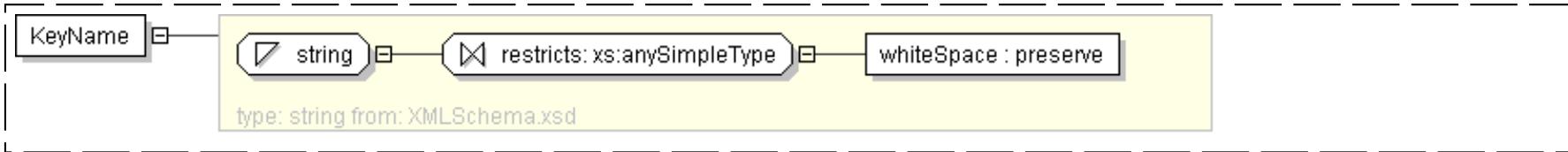
[top](#)**Element: KeyName****Name**

KeyName

Type	string
<u>Nillable</u>	no
<u>Abstract</u>	no

**Logical Diagram****XML Instance Representation**

```
<ds:KeyName> string </ds:KeyName>
```

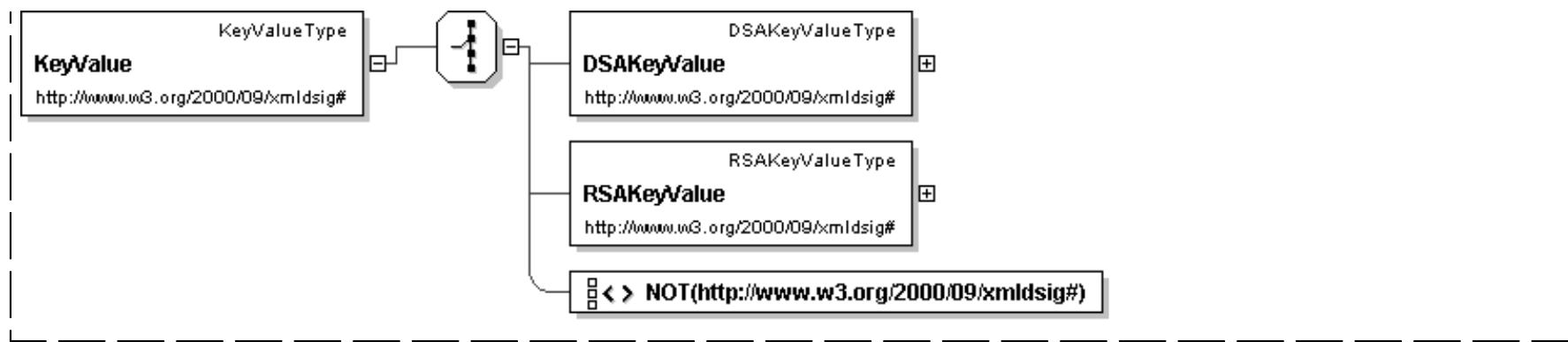
**Diagram****Schema Component Representation**

```
<element name="KeyName" type="string" />
```

[top](#)**Element: KeyValue**

Name	KeyValue
Type	<a href="#">ds:KeyValue</a>
<u>Nillable</u>	no
<u>Abstract</u>	no

**Logical Diagram**

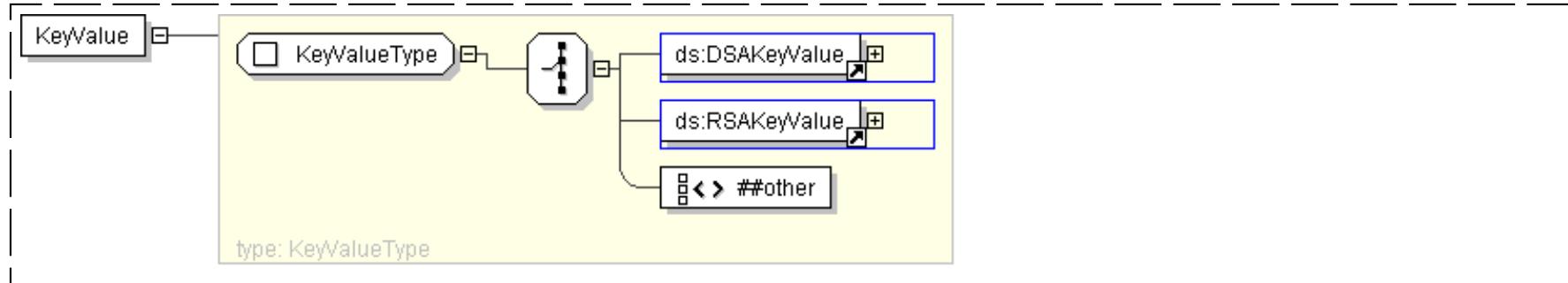


### XML Instance Representation

```

<ds:KeyValue>
  <!-- Mixed content -->
  Start Choice [1]
    <ds:DSAKeyValue> ... </ds:DSAKeyValue> [1]
    <ds:RSAKeyValue> ... </ds:RSAKeyValue> [1]
    Allow any elements from a namespace other than this schema's namespace (lax validation). [1]
  End Choice
</ds:KeyValue>
  
```

### Diagram



### Schema Component Representation

```

<element name="KeyValue" type=" ds:KeyValue_type" />
  
```

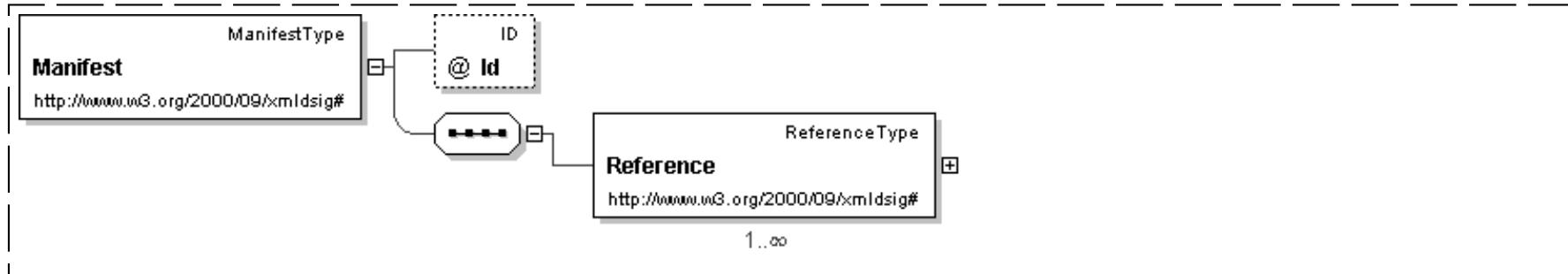
[top](#)

## Element: Manifest

Name	Manifest
------	----------

Type	<a href="#">ds:ManifestType</a>
Nillable	no
Abstract	no

### Logical Diagram

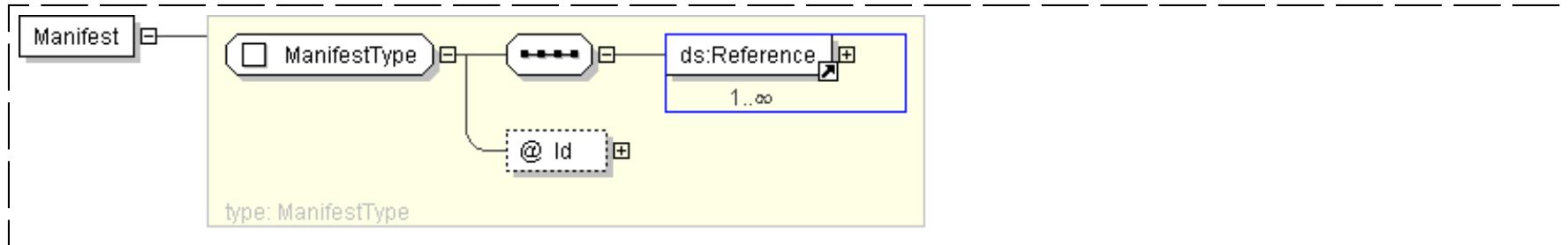


### XML Instance Representation

```

<ds:Manifest
  Id="ID [0..1]">
  <ds:Reference> ... </ds:Reference> [1..*]
</ds:Manifest>
  
```

### Diagram



### Schema Component Representation

```

<element name="Manifest" type="ds:ManifestType" />
  
```

top

## Element: MgmtData

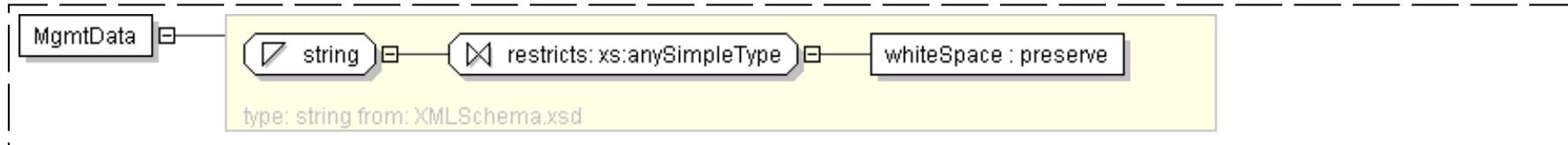
Name	MgmtData
Type	string

<b>Nillable</b>	no
<b>Abstract</b>	no

**Logical Diagram****XML Instance Representation**

```

<ds:MgmtData> string </ds:MgmtData>
  
```

**Diagram****Schema Component Representation**

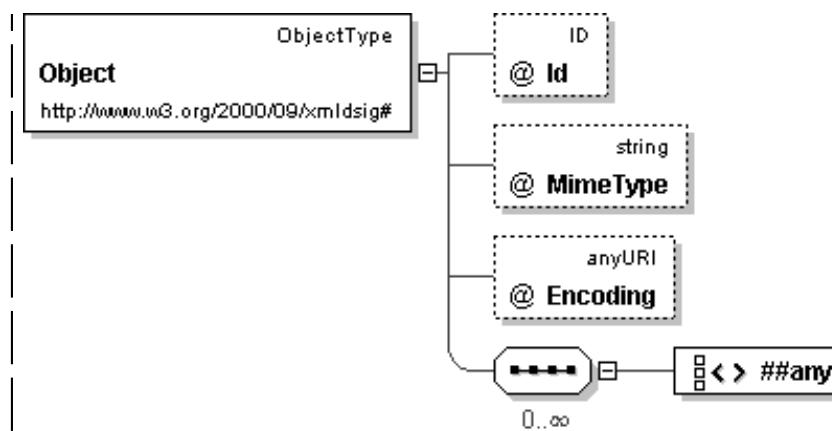
```

<element name="MgmtData" type=" string "/>
  
```

[top](#)**Element: Object**

<b>Name</b>	Object
<b>Type</b>	<a href="#">ds:ObjectType</a>
<b>Nillable</b>	no
<b>Abstract</b>	no

**Logical Diagram**

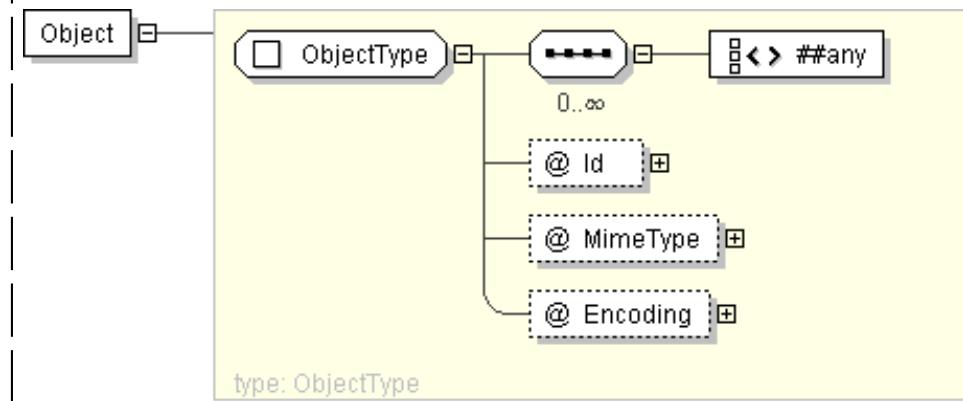


### XML Instance Representation

```

<ds:Object
  Id="ID [0..1]"
  MimeType="string [0..1]"
  Encoding="anyURI [0..1]">
  <!-- Mixed content -->
  Start Sequence [0..*]
  Allow any elements from any namespace (lax validation). [1]
  End Sequence
</ds:Object>
  
```

### Diagram



### Schema Component Representation

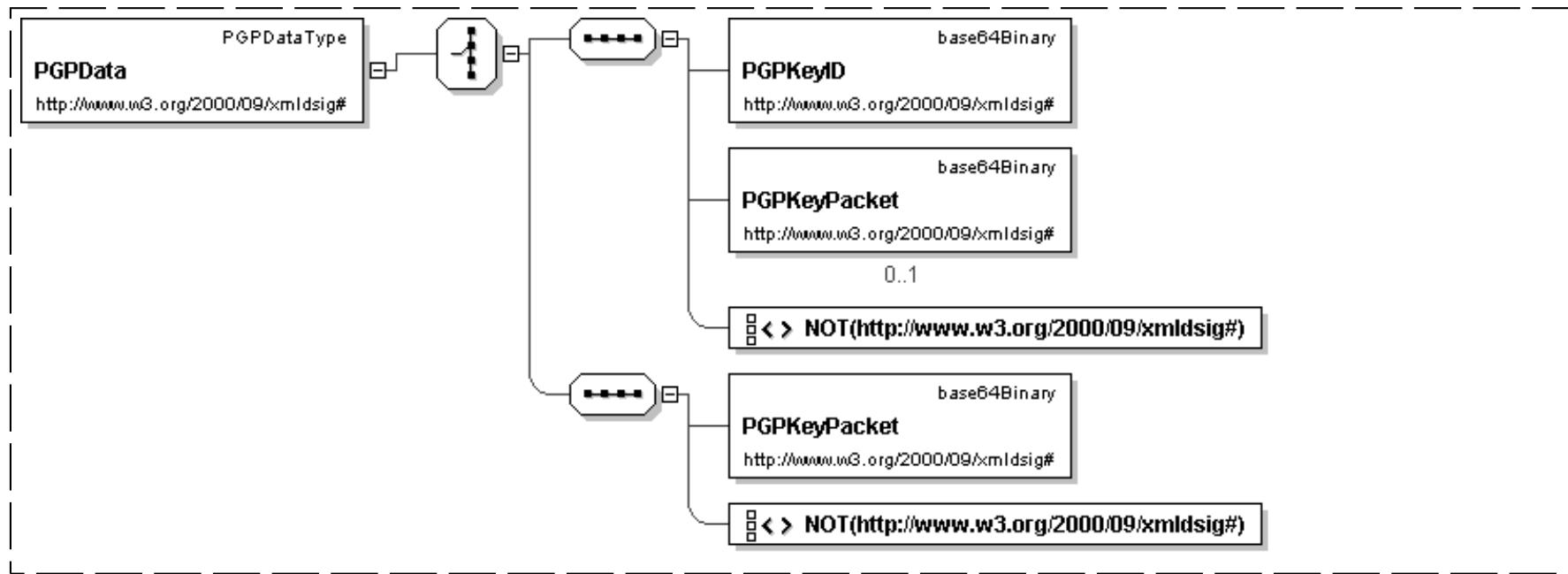
```

<element name="Object" type=" ds:ObjectType" />
  
```

## Element: PGPData

<b>Name</b>	PGPData
<b>Type</b>	<a href="#">ds:PGPDataType</a>
<b>Nillable</b>	no
<b>Abstract</b>	no

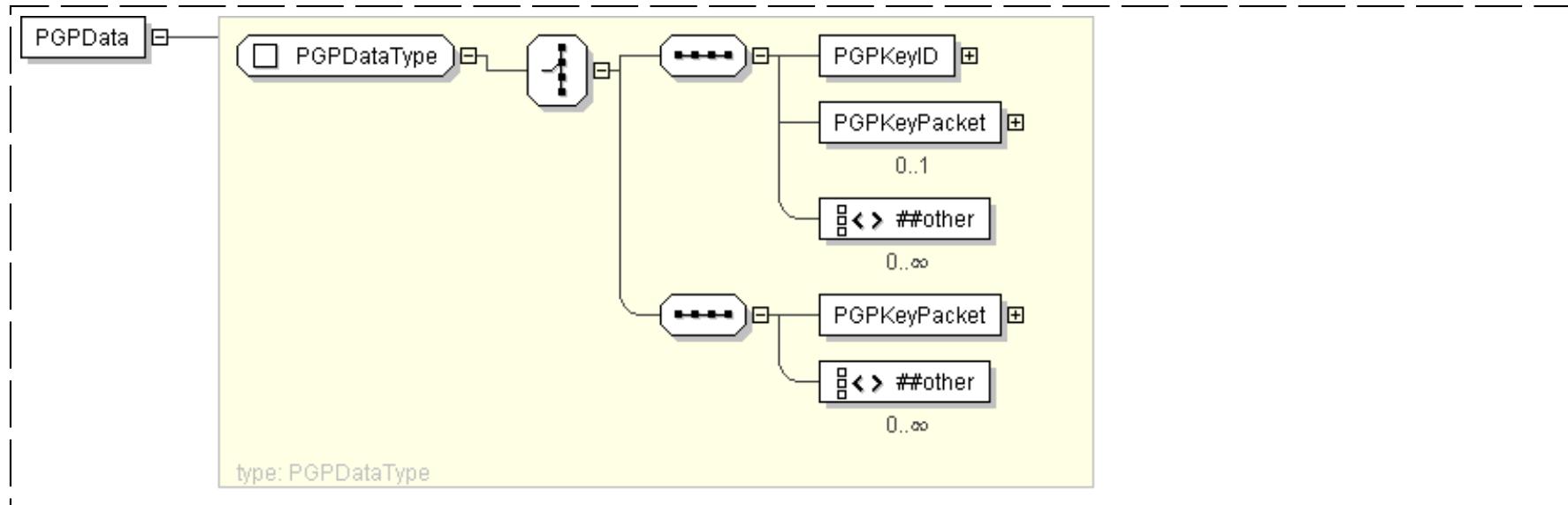
### Logical Diagram



### XML Instance Representation

```

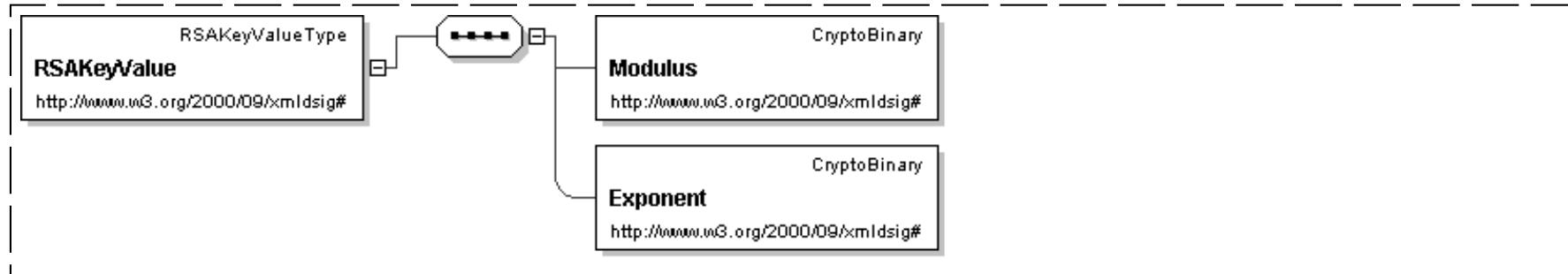
<ds:PGPData>
  Start Choice [1]
    <ds:PGPKeyID> base64Binary </ds:PGPKeyID> [1]
    <ds:PGPKeyPacket> base64Binary </ds:PGPKeyPacket> [0..1]
      Allow any elements from a namespace other than this schema's namespace (lax validation). [0..*]
    <ds:PGPKeyPacket> base64Binary </ds:PGPKeyPacket> [1]
      Allow any elements from a namespace other than this schema's namespace (lax validation). [0..*]
  End Choice
</ds:PGPData>
  
```

**Diagram****Schema Component Representation**

```
<element name="PGPData" type="ds:PGPDataType" />
```

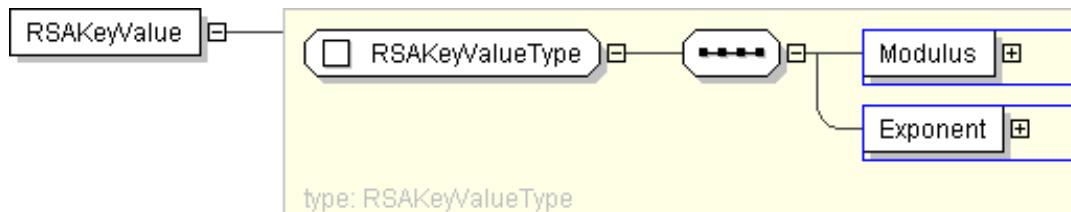
[top](#)**Element: RSAKeyValue**

<b>Name</b>	RSAKeyValue
<b>Type</b>	<a href="#">ds:RSAKeyValue</a>
<b>Nullable</b>	no
<b>Abstract</b>	no

**Logical Diagram**

**XML Instance Representation**

```
<ds:RSAKeyValue>
  <ds:Modulus> ds: CryptoBinary </ds:Modulus> [1]
  <ds:Exponent> ds: CryptoBinary </ds:Exponent> [1]
</ds:RSAKeyValue>
```

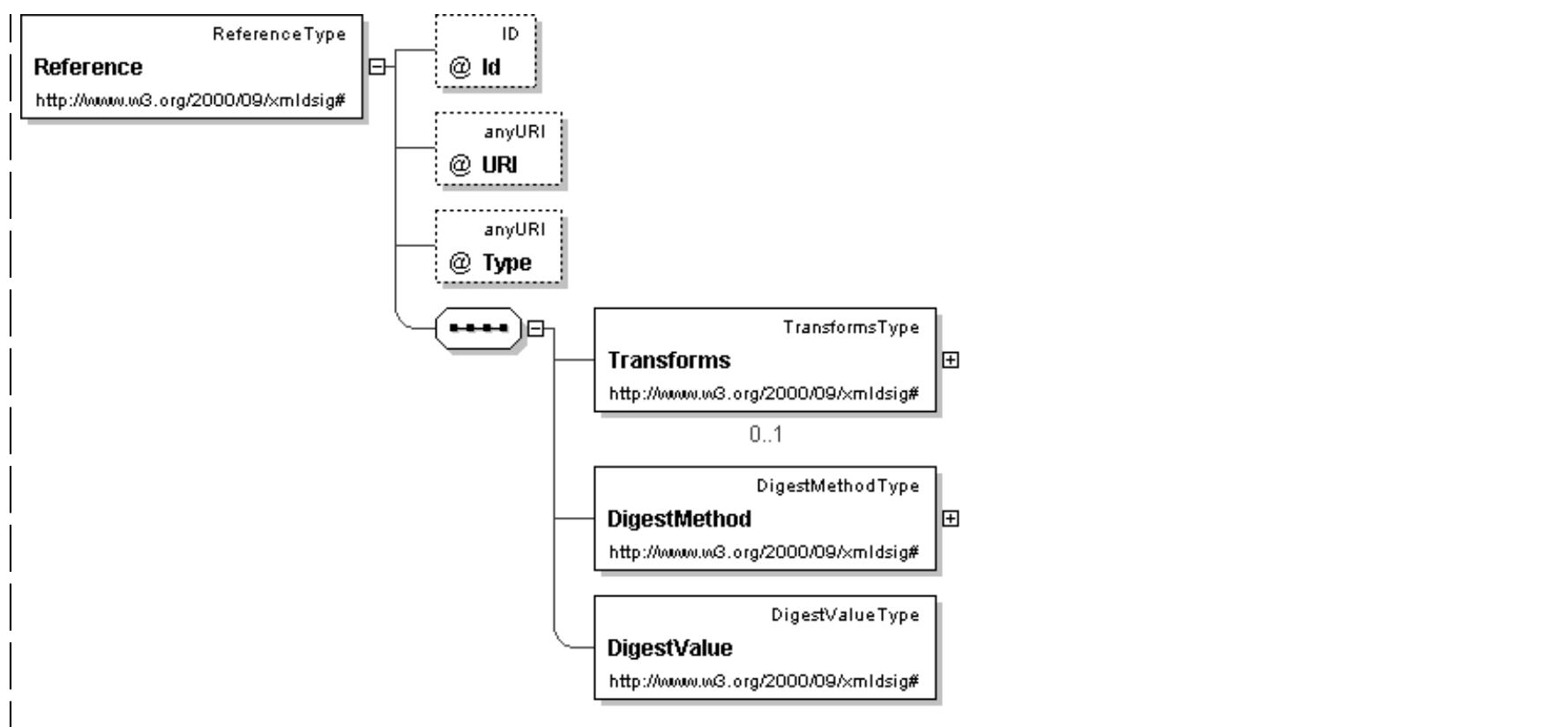
**Diagram****Schema Component Representation**

```
<element name="RSAKeyValue" type=" ds:RSAKeyValueType" />
```

[top](#)**Element: Reference**

<b>Name</b>	Reference
<b>Type</b>	<a href="#">ds:ReferenceType</a>
<b>Nillable</b>	no
<b>Abstract</b>	no

**Logical Diagram**

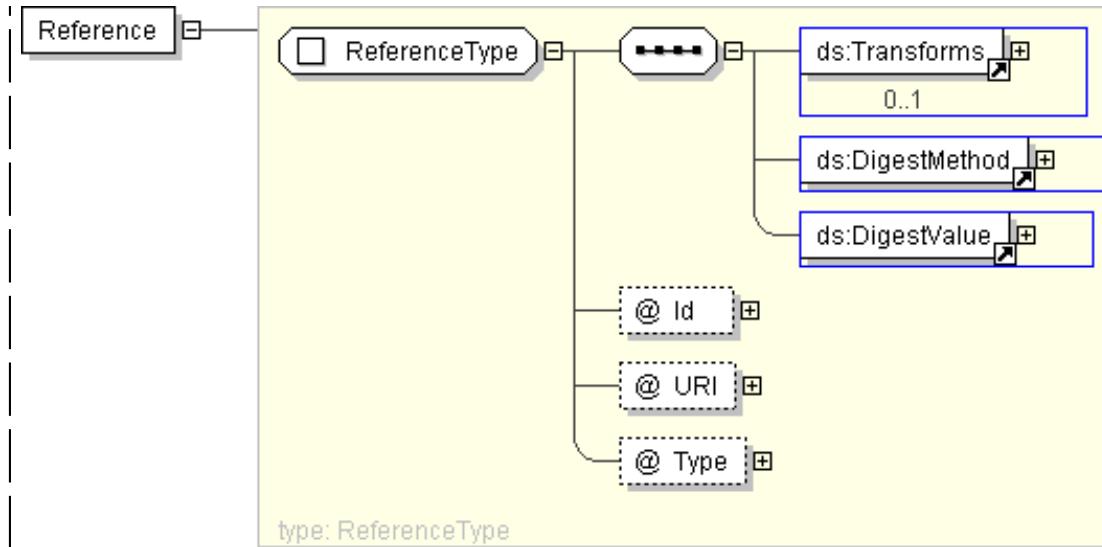


### XML Instance Representation

```

<ds:Reference
  Id="ID [0..1]"
  URI="anyURI [0..1]"
  Type="anyURI [0..1]">
  <ds:Transforms> ... </ds:Transforms> [0..1]
  <ds:DigestMethod> ... </ds:DigestMethod> [1]
  <ds:DigestValue> ... </ds:DigestValue> [1]
</ds:Reference>
  
```

### Diagram



### Schema Component Representation

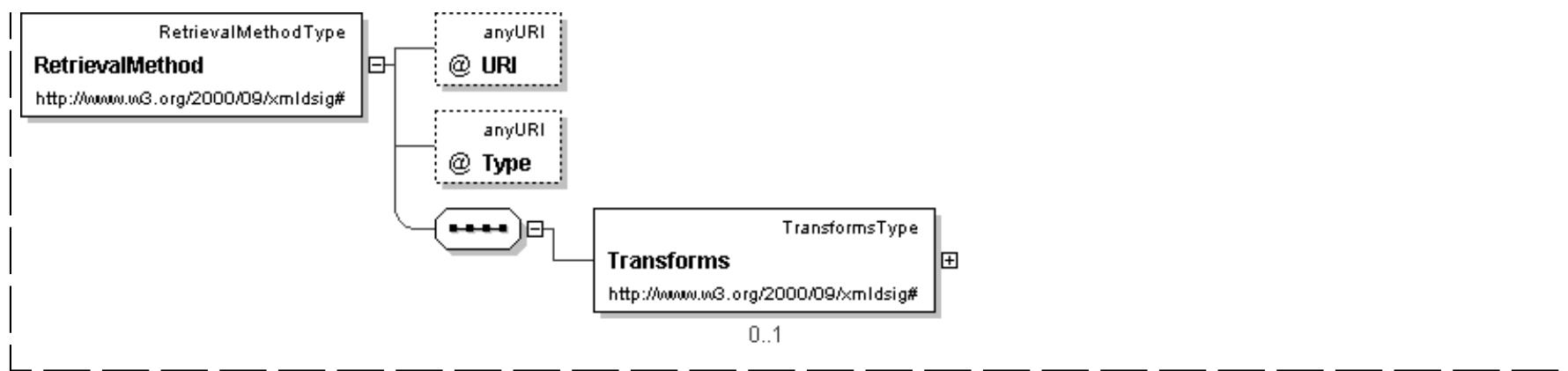
```
<element name="Reference" type="ds:ReferenceType" />
```

[top](#)

### Element: **RetrievalMethod**

<b>Name</b>	RetrievalMethod
<b>Type</b>	<a href="#">ds:RetrievalMethodType</a>
<b>Nillable</b>	no
<b>Abstract</b>	no

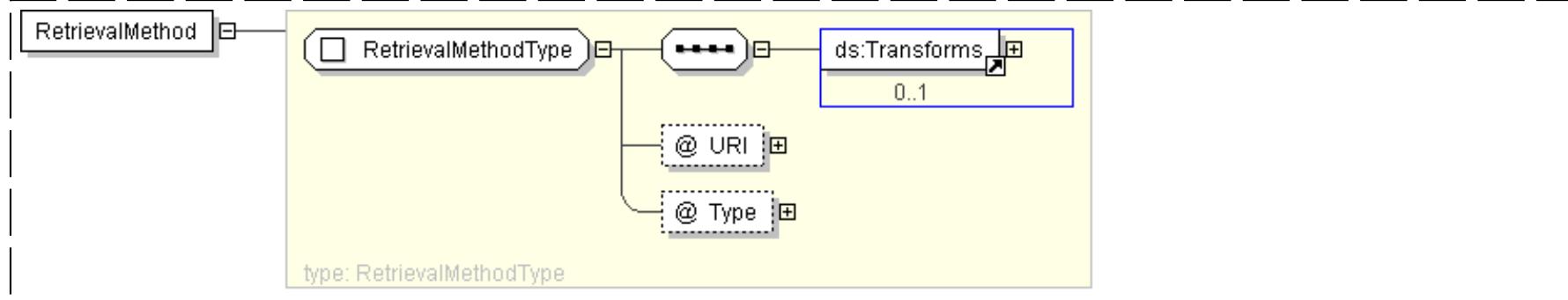
### Logical Diagram



### XML Instance Representation

```
<ds:RetrievalMethod
  URI="anyURI [0..1]"
  Type="anyURI [0..1]">
  <ds:Transforms> ... </ds:Transforms> [0..1]
</ds:RetrievalMethod>
```

### Diagram



### Schema Component Representation

```
<element name="RetrievalMethod" type=" ds:RetrievalMethodType " />
```

[top](#)

## Element: SPKIData

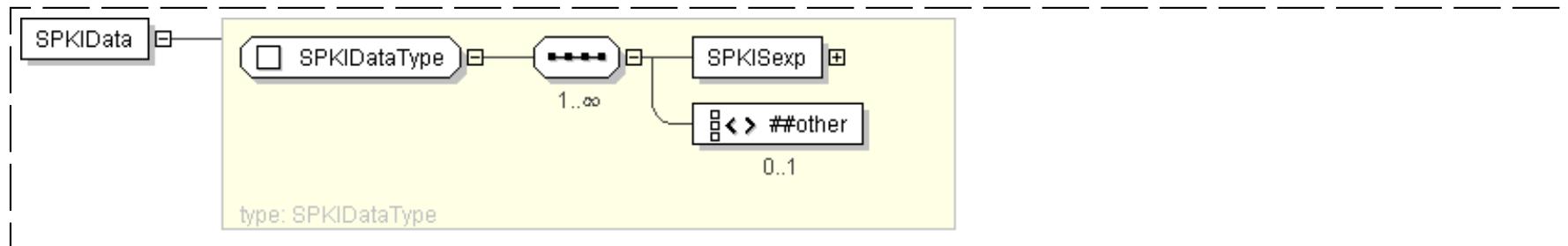
Name	SPKIData
------	----------

Type	<a href="#">ds:SPKIDataType</a>
Nillable	no
Abstract	no

**Logical Diagram****XML Instance Representation**

```

<ds:SPKIData>
  Start Sequence [1..*]
    <ds:SPKISexp> base64Binary </ds:SPKISexp> [1]
    Allow any elements from a namespace other than this schema's namespace (lax validation). [0..1]
  End Sequence
</ds:SPKIData>
  
```

**Diagram****Schema Component Representation**

```

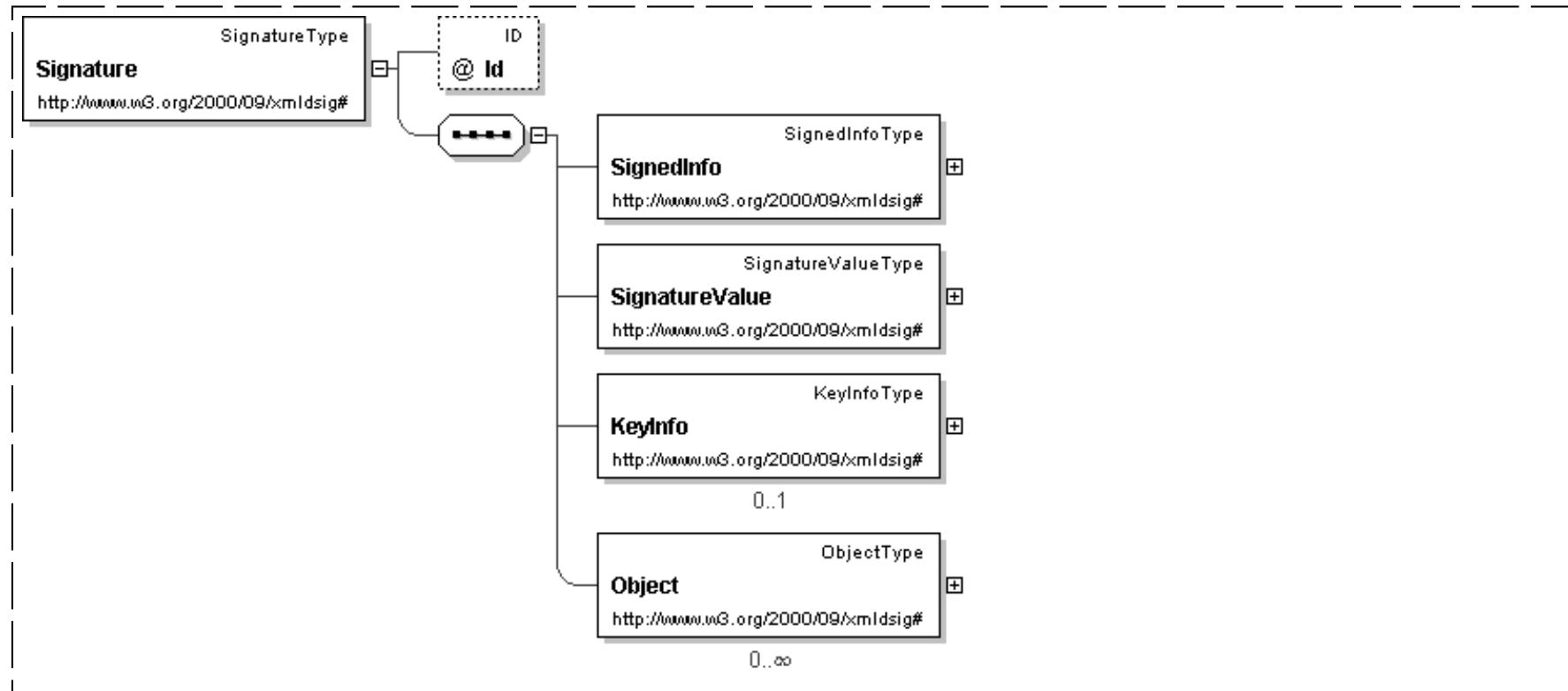
<element name="SPKIData" type="ds:SPKIDataType" />
  
```

[top](#)**Element: Signature**

Name	Signature
------	-----------

Type	<a href="#">ds:SignatureType</a>
Nillable	no
Abstract	no

### Logical Diagram

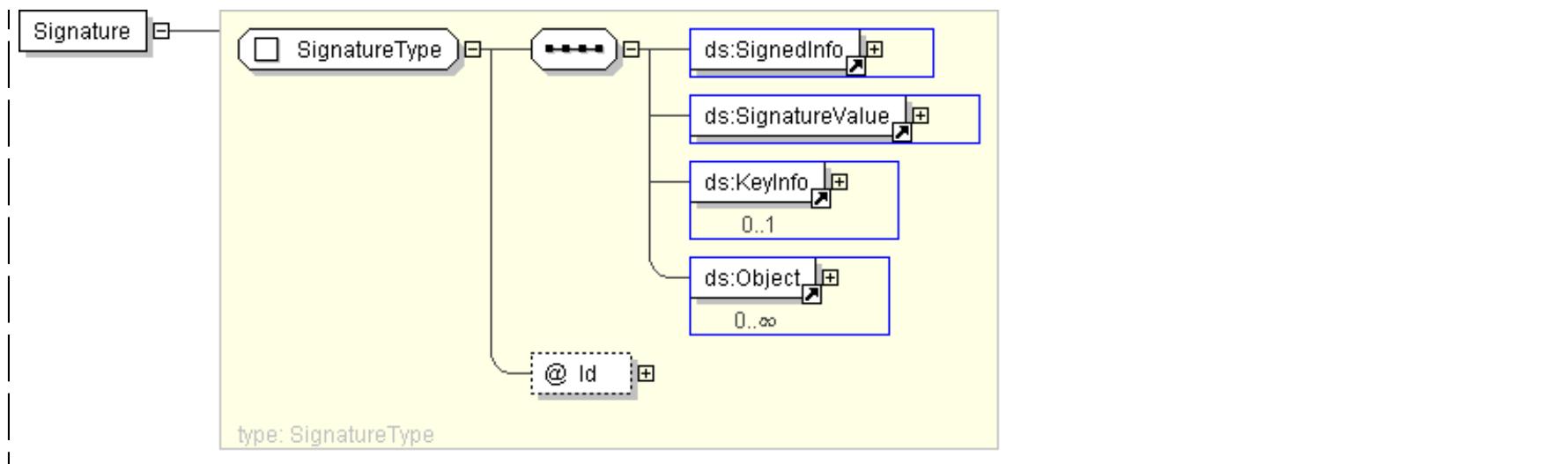


### XML Instance Representation

```

<ds:Signature
  Id="#ID [0..1]">
  <ds:SignedInfo> ... </ds:SignedInfo> [1]
  <ds:SignatureValue> ... </ds:SignatureValue> [1]
  <ds:KeyInfo> ... </ds:KeyInfo> [0..1]
  <ds:Object> ... </ds:Object> [0..*]
</ds:Signature>
  
```

### Diagram



### Schema Component Representation

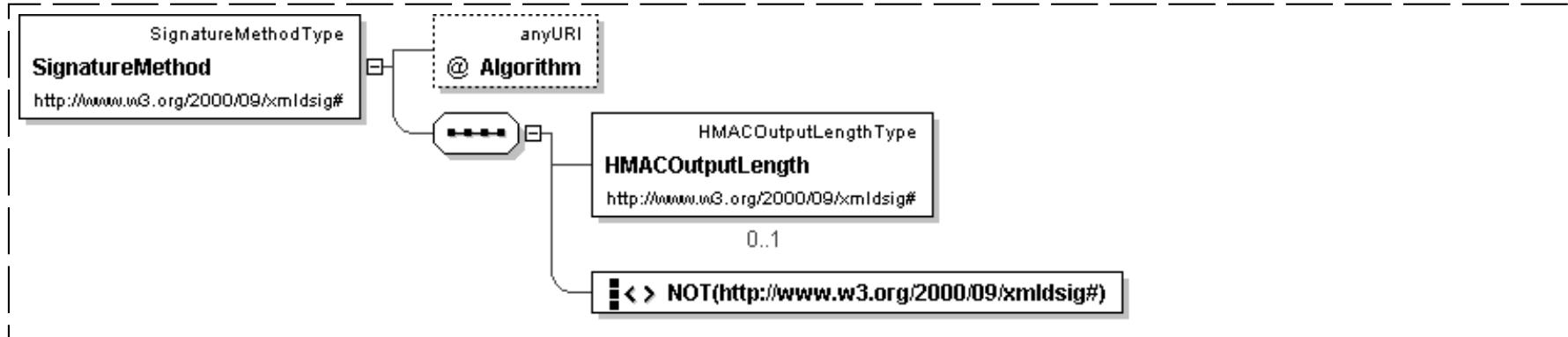
```
<element name="Signature" type="ds:SignatureType" />
```

[top](#)

## Element: SignatureMethod

<b>Name</b>	SignatureMethod
<b>Type</b>	<a href="#">ds:SignatureMethodType</a>
<b>Nillable</b>	no
<b>Abstract</b>	no

### Logical Diagram

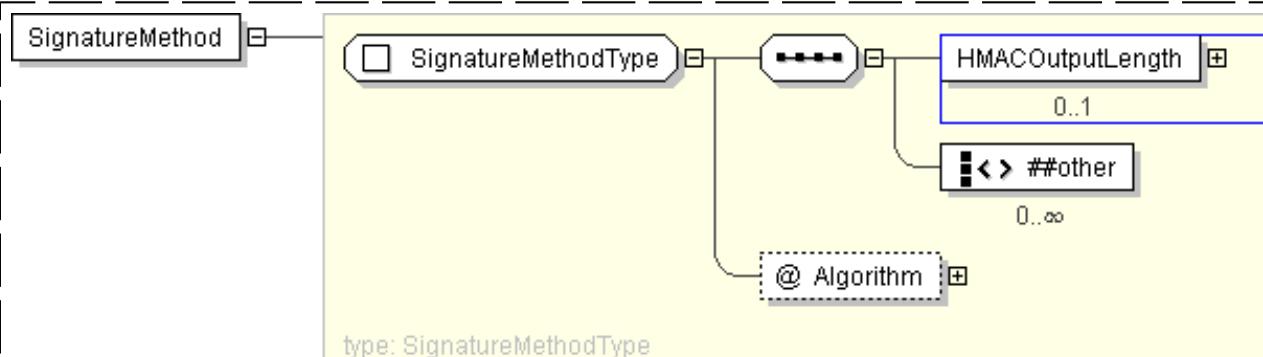


**XML Instance Representation**

```

<ds:SignatureMethod
  Algorithm="anyURI [1]">
  <!-- Mixed content -->
  <ds:HMACOutputLength> ds:HMACOutputLengthType </ds:HMACOutputLength> [0..1]
  Allow any elements from a namespace other than this schema's namespace (strict validation).
  [0..*]
</ds:SignatureMethod>

```

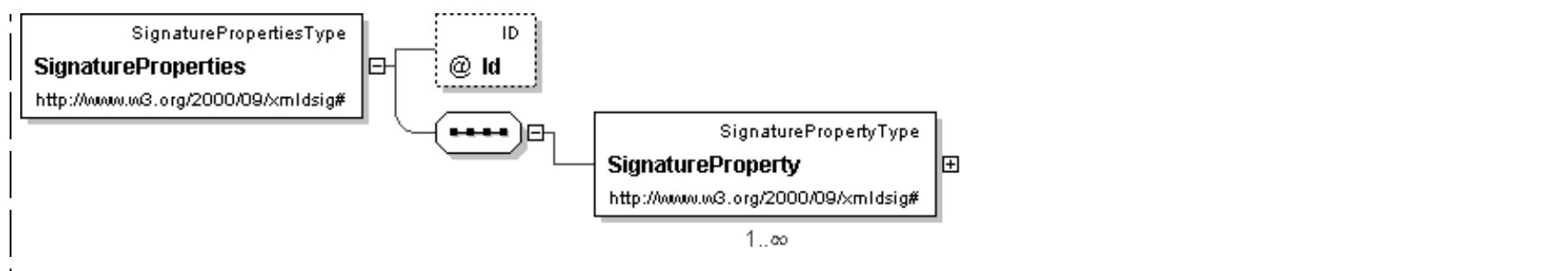
**Diagram****Schema Component Representation**

```
<element name="SignatureMethod" type="ds:SignatureMethodType" />
```

[top](#)**Element: SignatureProperties**

<b>Name</b>	SignatureProperties
<b>Type</b>	<a href="#">ds:SignaturePropertiesType</a>
<b>Nillable</b>	no
<b>Abstract</b>	no

**Logical Diagram**

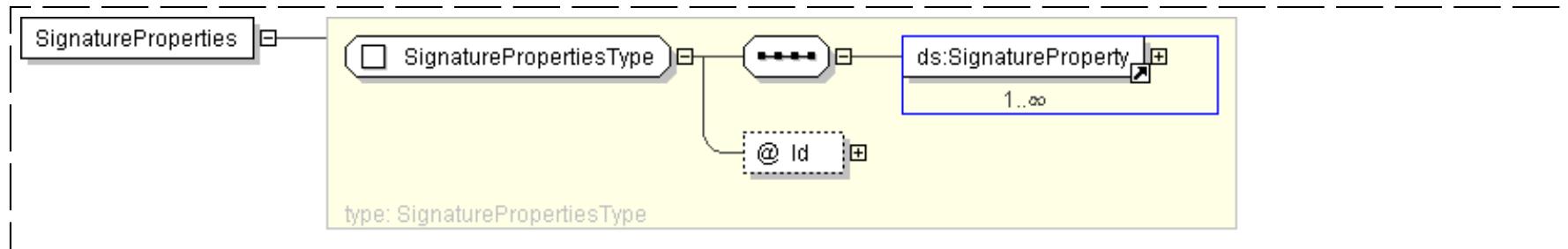


### XML Instance Representation

```

<ds:SignatureProperties
  Id="ID [0..1]">
  <ds:SignatureProperty> ... </ds:SignatureProperty> [1..*]
</ds:SignatureProperties>
  
```

### Diagram



### Schema Component Representation

```

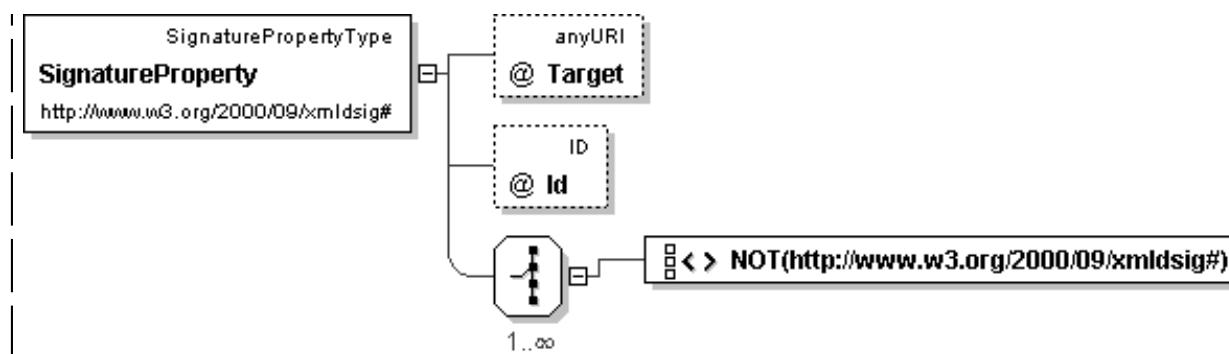
<element name="SignatureProperties" type=" ds:SignaturePropertiesType " />
  
```

[top](#)

## Element: SignatureProperty

Name	SignatureProperty
Type	<a href="#">ds:SignaturePropertyType</a>
Nillable	no
Abstract	no

### Logical Diagram

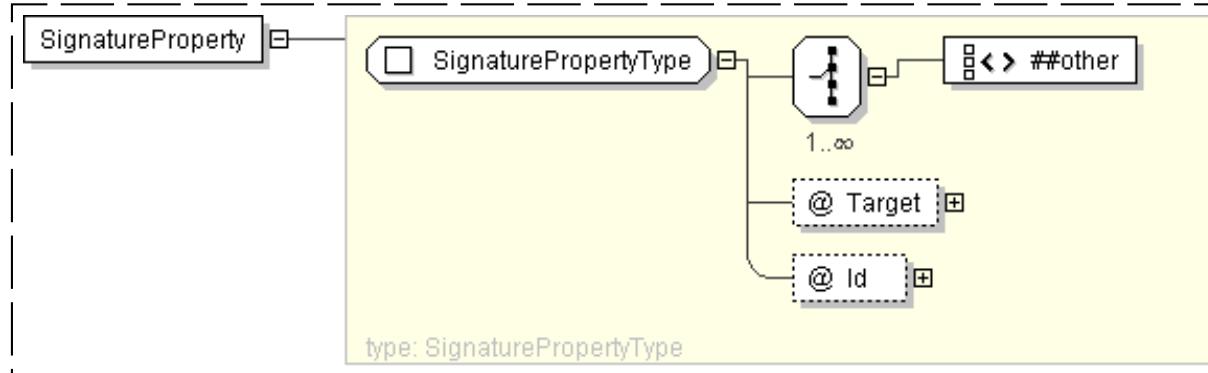


### XML Instance Representation

```

<ds:SignatureProperty
  Target="anyURI [1]"
  Id="ID [0..1]">
  <!-- Mixed content -->
  Start Choice [1..*]
    Allow any elements from a namespace other than this schema's namespace (lax validation). [1]
  End Choice
</ds:SignatureProperty>
  
```

### Diagram



### Schema Component Representation

```

<element name="SignatureProperty" type=" ds:Signature.PropertyType" />
  
```

[top](#)

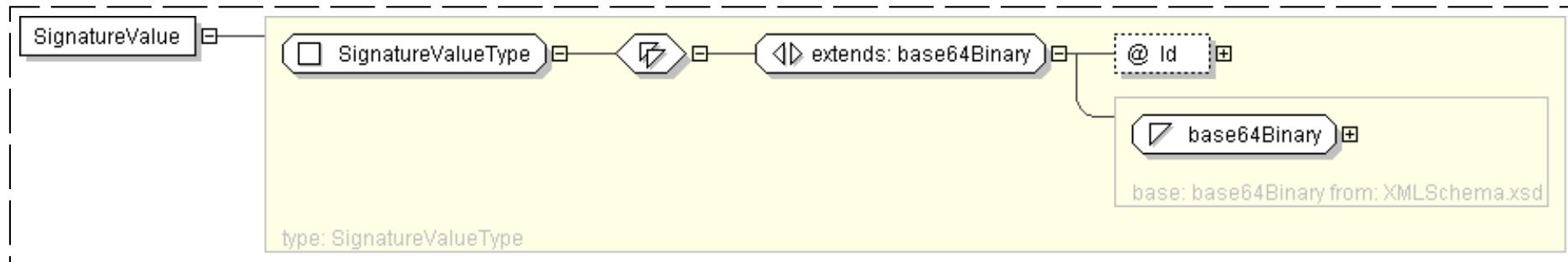
## Element: SignatureValue

<b>Name</b>	SignatureValue
<b>Type</b>	<a href="#">ds:SignatureValueType</a>
<b>Nillable</b>	no
<b>Abstract</b>	no

**Logical Diagram****XML Instance Representation**

```

<ds:SignatureValue
  Id="ID [0..1]">
  base64Binary
</ds:SignatureValue>
  
```

**Diagram****Schema Component Representation**

```

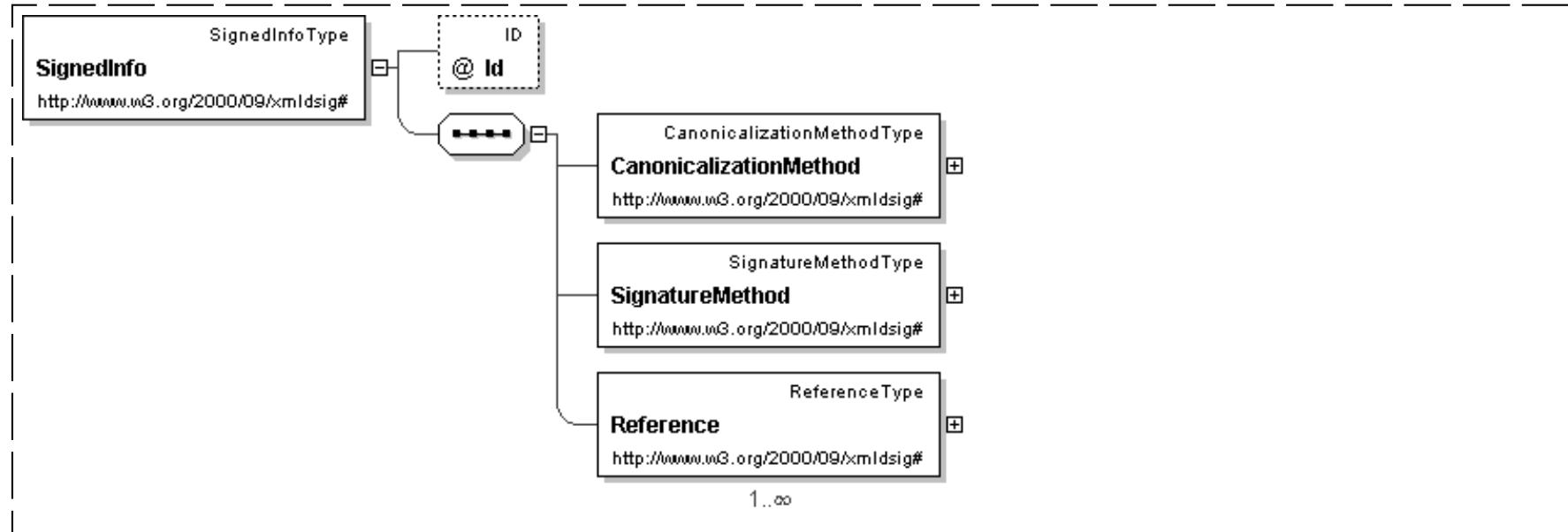
<element name="SignatureValue" type="ds:SignatureValueType" />
  
```

[top](#)**Element: SignedInfo**

<b>Name</b>	SignedInfo
<b>Type</b>	<a href="#">ds:SignedInfoType</a>
<b>Nillable</b>	no

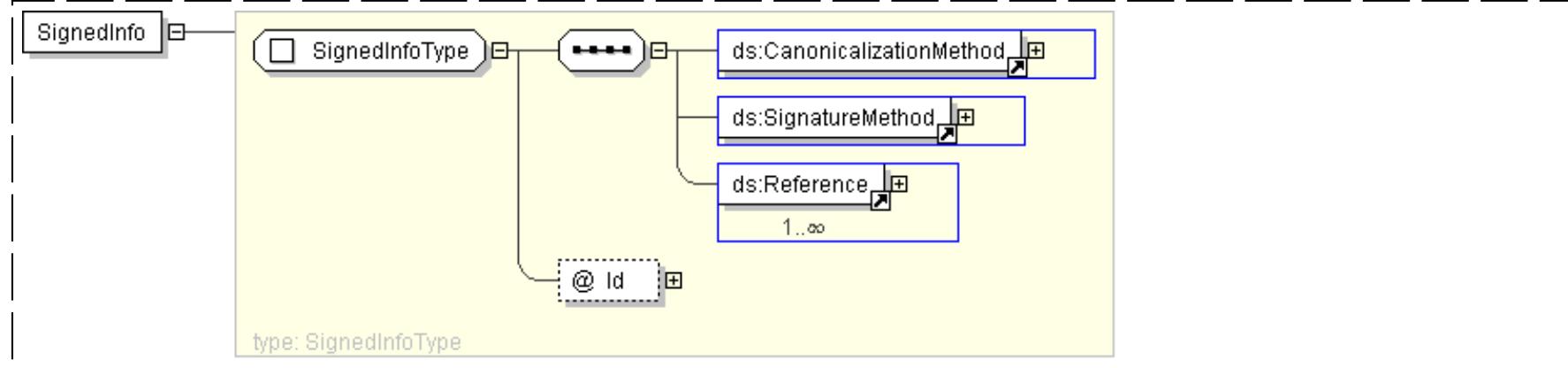
**Abstract**

no

**Logical Diagram****XML Instance Representation**

```

<ds:SignedInfo
  Id="ID [0..1]">
  <ds:CanonicalizationMethod> ... </ds:CanonicalizationMethod> [1]
  <ds:SignatureMethod> ... </ds:SignatureMethod> [1]
  <ds:Reference> ... </ds:Reference> [1..*]
</ds:SignedInfo>
  
```

**Diagram**

## Schema Component Representation

```
<element name="SignedInfo" type="ds:SignedInfoType" />
```

[top](#)

## Element: Transform

<b>Name</b>	Transform
<b>Type</b>	<a href="#">ds:TransformType</a>
<b>Nillable</b>	no
<b>Abstract</b>	no

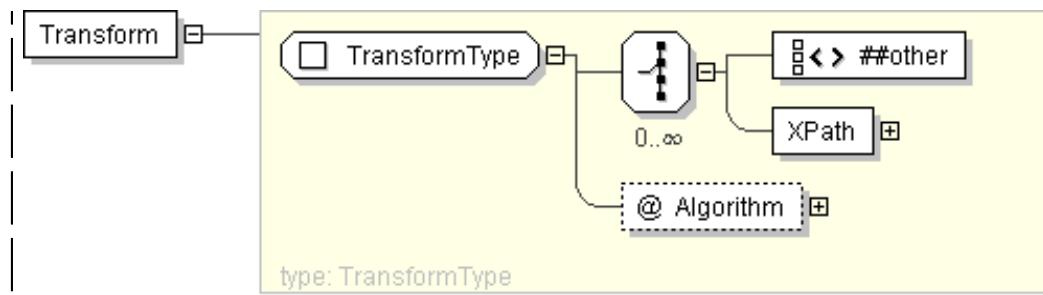
### Logical Diagram



### XML Instance Representation

```
<ds:Transform
  Algorithm="anyURI [1]">
  <!-- Mixed content -->
  Start Choice [0..*]
    Allow any elements from a namespace other than this schema's namespace (lax validation). [1]
    <ds:XPath> string </ds:XPath> [1]
  End Choice
</ds:Transform>
```

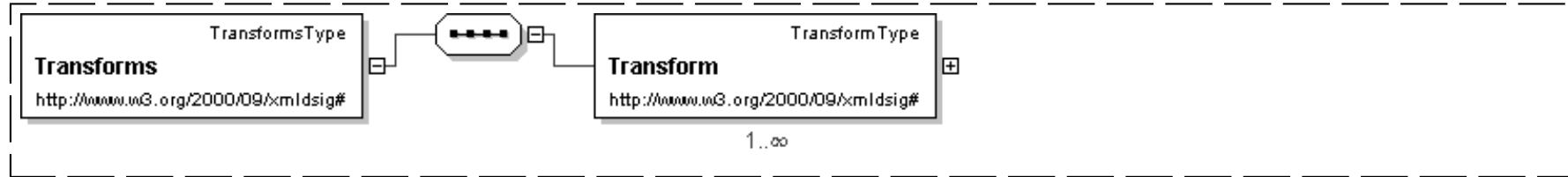
### Diagram

**Schema Component Representation**

```
<element name="Transform" type=" ds:TransformType " />
```

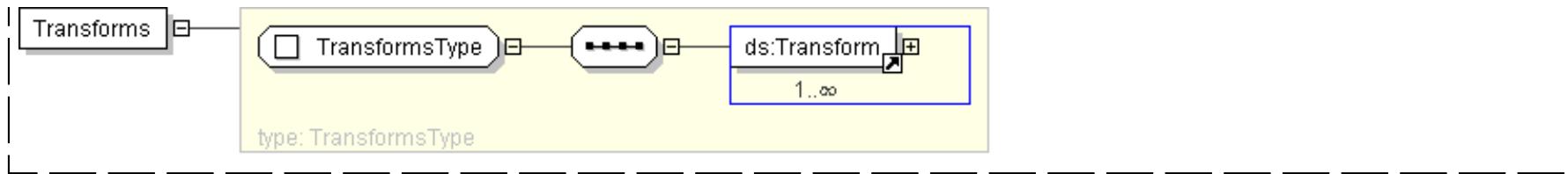
[top](#)**Element: Transforms**

<b>Name</b>	Transforms
<b>Type</b>	<a href="#">ds:TransformsType</a>
<b>Nillable</b>	no
<b>Abstract</b>	no

**Logical Diagram****XML Instance Representation**

```
<ds:Transforms>
  <ds:Transform> ... </ds:Transform> [1..*]
</ds:Transforms>
```

**Diagram**



### Schema Component Representation

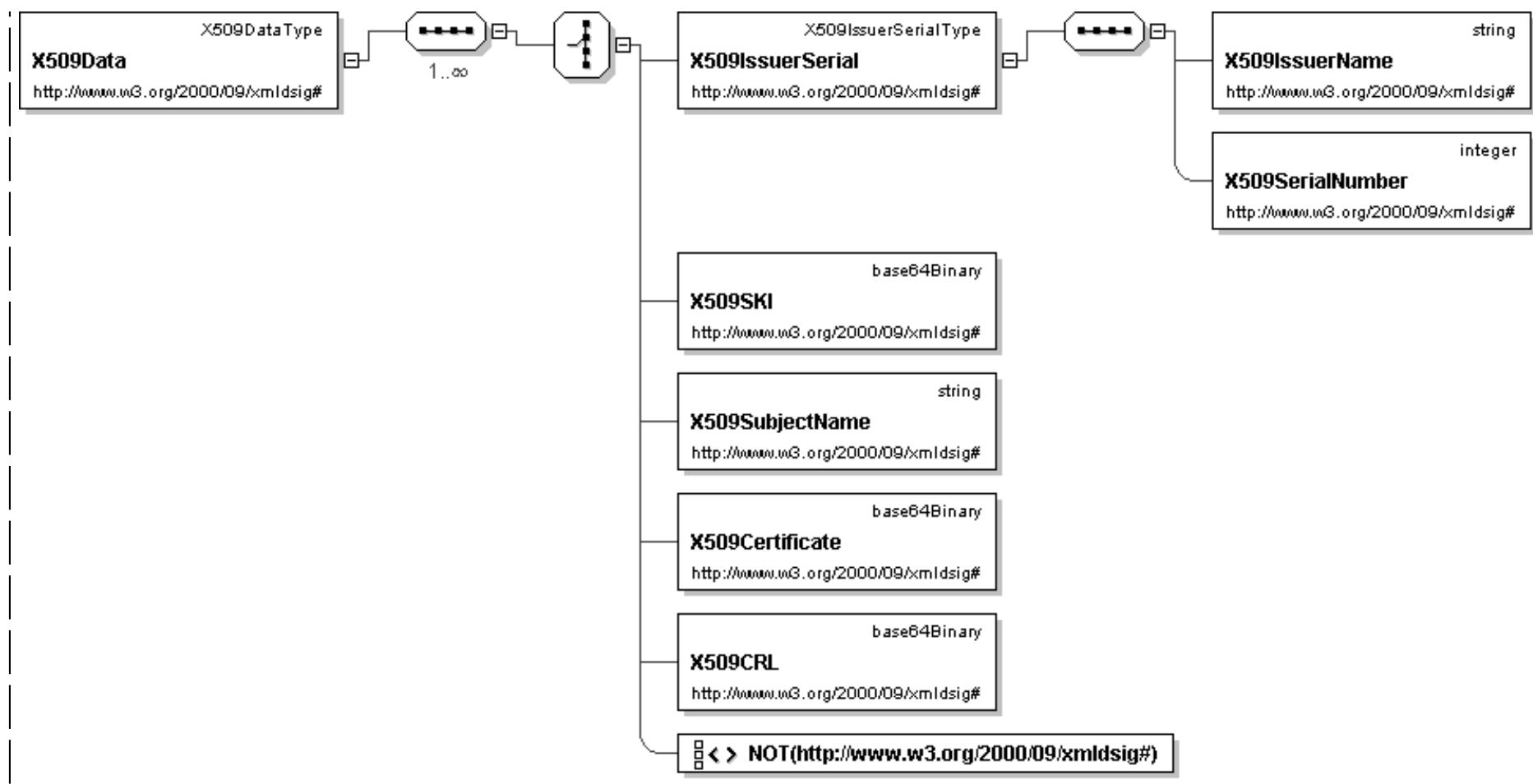
```
<element name="Transforms" type="ds:TransformsType" />
```

[top](#)

## Element: X509Data

Name	X509Data
Type	<a href="#">ds:X509DataType</a>
Nillable	no
Abstract	no

### Logical Diagram

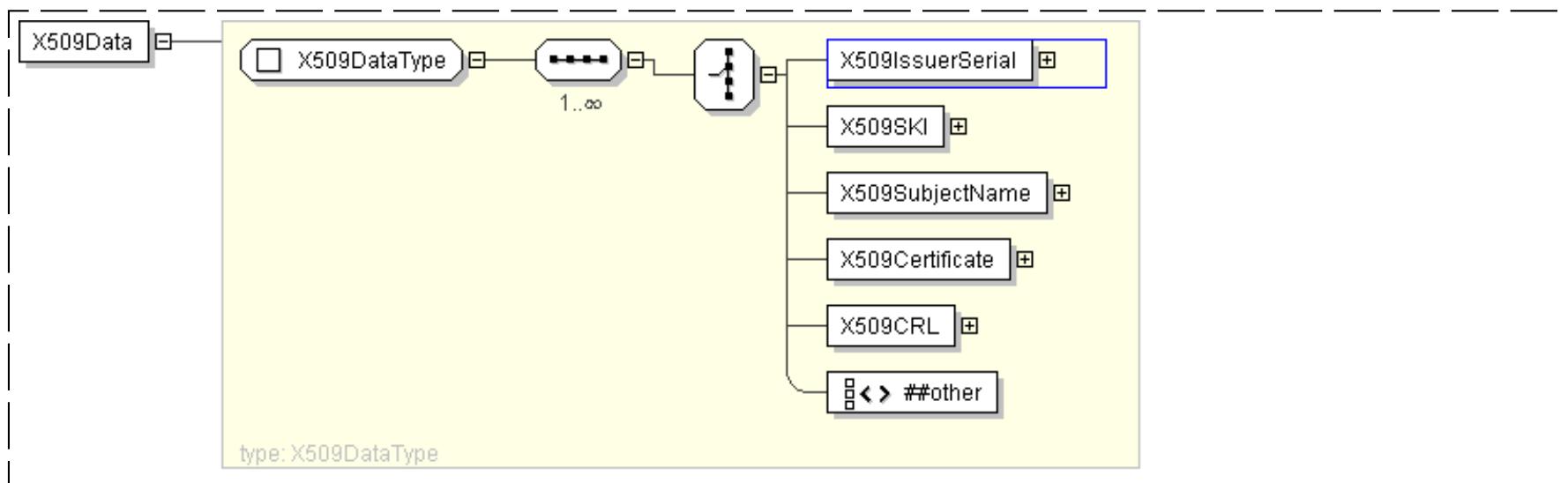


### XML Instance Representation

```

<ds:X509Data>
Start Sequence [1..*]
Start Choice [1]
    <ds:X509IssuerSerial> ds:X509IssuerSerialType </ds:X509IssuerSerial> [1]
    <ds:X509SKI> base64Binary </ds:X509SKI> [1]
    <ds:X509SubjectName> string </ds:X509SubjectName> [1]
    <ds:X509Certificate> base64Binary </ds:X509Certificate> [1]
    <ds:X509CRL> base64Binary </ds:X509CRL> [1]
    Allow any elements from a namespace other than this schema's namespace (lax validation). [1]
End Choice
End Sequence
</ds:X509Data>
  
```

### Diagram



### Schema Component Representation

```
<element name="X509Data" type="ds:X509DataType" />
```

[top](#)

## Global Definitions

### Complex Type: CanonicalizationMethodType

<b>Super-types:</b>	None
<b>Sub-types:</b>	None

<b>Name</b>	CanonicalizationMethodType
<b>Abstract</b>	no

### XML Instance Representation

```
<...>
<Algorithm="anyURI [1]">
<!-- Mixed content -->
  Allow any elements from any namespace (strict validation). [0..*]
</...>
```

### Diagram



## Schema Component Representation

```

<complexType name="CanonicalizationMethodType" mixed="true">
  <sequence>
    <any namespace="##any" minOccurs="0" maxOccurs="unbounded" />
    <!-- (0,unbounded) elements from (1,1) namespace -->
  </sequence>
  <attribute name="Algorithm" type="anyURI" use="required"/>
</complexType>
  
```

[top](#)

## Complex Type: DSAKeyValueType

<b>Super-types:</b>	None
<b>Sub-types:</b>	None

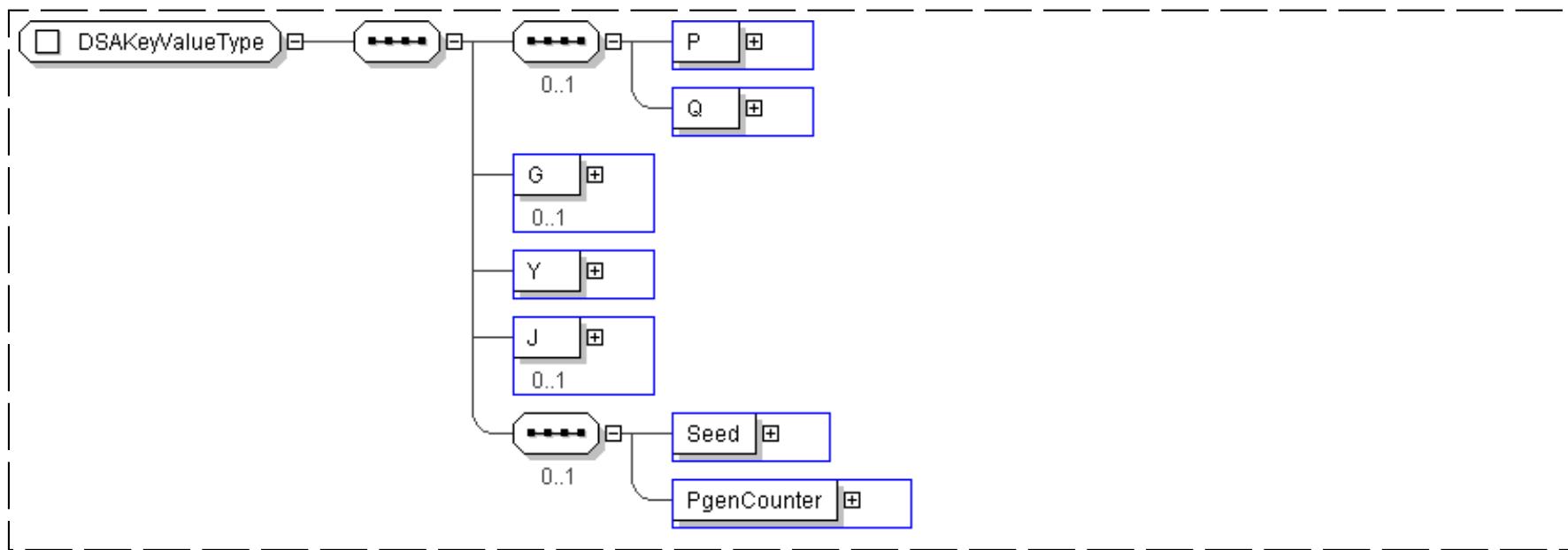
<b>Name</b>	DSAKeyValueType
<b>Abstract</b>	no

### XML Instance Representation

```

<...>
| Start Sequence [0..1]
|   <ds:P> ds: CryptoBinary </ds:P> [1]
|   <ds:Q> ds: CryptoBinary </ds:Q> [1]
|
| End Sequence
|   <ds:G> ds: CryptoBinary </ds:G> [0..1]
|   <ds:Y> ds: CryptoBinary </ds:Y> [1]
|   <ds:J> ds: CryptoBinary </ds:J> [0..1]
|
| Start Sequence [0..1]
|   <ds:Seed> ds: CryptoBinary </ds:Seed> [1]
|   <ds:PgenCounter> ds: CryptoBinary </ds:PgenCounter> [1]
|
| End Sequence
</...>
  
```

### Diagram



### Schema Component Representation

```

<complexType name="DSAKeyValueType">
  <sequence>
    <sequence minOccurs="0">
      <element name="P" type="ds: CryptoBinary" />
      <element name="Q" type="ds: CryptoBinary" />
    </sequence>
    <element name="G" type="ds: CryptoBinary" minOccurs="0" />
    <element name="Y" type="ds: CryptoBinary" />
    <element name="J" type="ds: CryptoBinary" minOccurs="0" />
    <sequence minOccurs="0">
      <element name="Seed" type="ds: CryptoBinary" />
      <element name="PgenCounter" type="ds: CryptoBinary" />
    </sequence>
  </sequence>
</complexType>
  
```

[top](#)

### Complex Type: DigestMethodType

<i>Super-types:</i>	None
<i>Sub-types:</i>	None

<b>Name</b>	DigestMethodType
<b>Abstract</b>	no

**XML Instance Representation**

```
<...>
Algorithm="anyURI [1]">
<!-- Mixed content -->
Allow any elements from a namespace other than this schema's namespace (lax validation). [0..*]
</...>
```

**Diagram****Schema Component Representation**

```
<complexType name="DigestMethodType" mixed="true">
  <sequence>
    <any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
  </sequence>
  <attribute name="Algorithm" type="anyURI" use="required"/>
</complexType>
```

[top](#)**Complex Type: KeyInfoType**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None

<b>Name</b>	KeyInfoType
<b>Abstract</b>	no

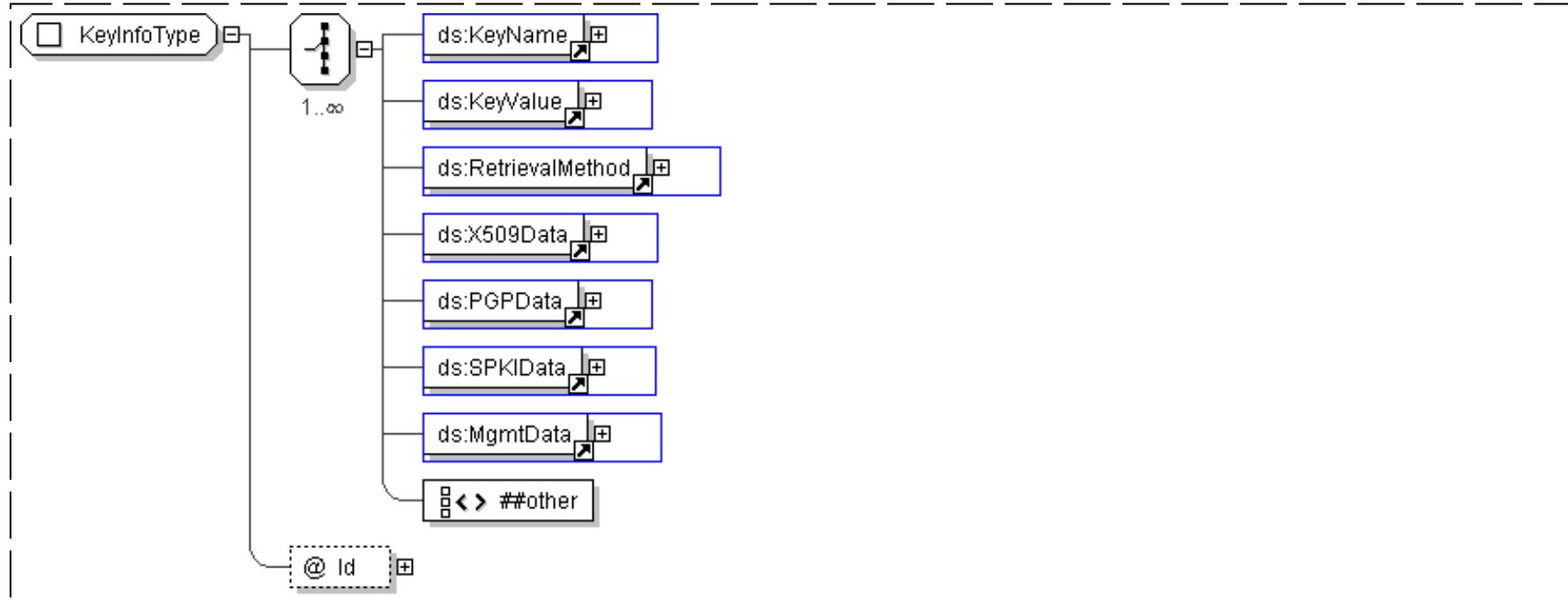
**XML Instance Representation**

```
<...>
Id="ID [0..1]">
<!-- Mixed content -->
```

```

| Start Choice [1..*]
|   <ds:KeyName> ... </ds:KeyName> [1]
|   <ds:KeyValue> ... </ds:KeyValue> [1]
|   <ds:RetrievalMethod> ... </ds:RetrievalMethod> [1]
|   <ds:X509Data> ... </ds:X509Data> [1]
|   <ds:PGPData> ... </ds:PGPData> [1]
|   <ds:SPKIData> ... </ds:SPKIData> [1]
|   <ds:MgmtData> ... </ds:MgmtData> [1]
|   Allow any elements from a namespace other than this schema's namespace (lax validation). [1]
| End Choice
</...>

```

**Diagram****Schema Component Representation**

```

<complexType name="KeyInfoType" mixed="true">
  <choice maxOccurs="unbounded">
    <element ref=" ds:KeyName " />
    <element ref=" ds:KeyValue " />
    <element ref=" ds:RetrievalMethod " />
    <element ref=" ds:X509Data " />
    <element ref=" ds:PGPData " />
    <element ref=" ds:SPKIData " />

```

```

<element ref=" ds:MgmtData " />
<any namespace="#other" processContents="lax" />
<!-- (1,1) elements from (0,unbounded) namespaces --&gt;
&lt;/choice&gt;
&lt;attribute name="Id" type=" ID " use="optional" /&gt;
&lt;/complexType&gt;
</pre>

```

[top](#)

## Complex Type: **KeyValue**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None

<b>Name</b>	KeyValue
<b>Abstract</b>	no

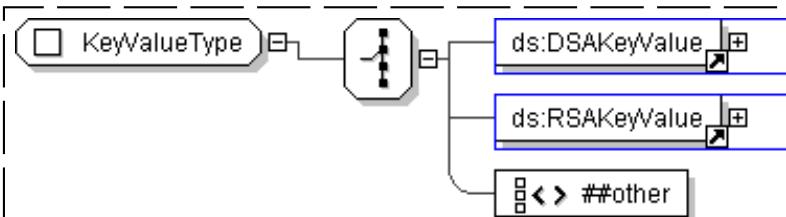
### XML Instance Representation

```

<...>
<!-- Mixed content -->
Start Choice [1]
  <ds:DSAKeyValue> ... </ds:DSAKeyValue> [1]
  <ds:RSAKeyValue> ... </ds:RSAKeyValue> [1]
  Allow any elements from a namespace other than this schema's namespace (lax validation). [1]
End Choice
</...>

```

### Diagram



### Schema Component Representation

```

<complexType name="KeyValue" mixed="true">
<choice>
  <element ref=" ds:DSAKeyValue " />
  <element ref=" ds:RSAKeyValue " />

```

```

<any namespace="##other" processContents="lax"/>
</choice>
</complexType>

```

[top](#)

## Complex Type: **ManifestType**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None

<b>Name</b>	ManifestType
<b>Abstract</b>	no

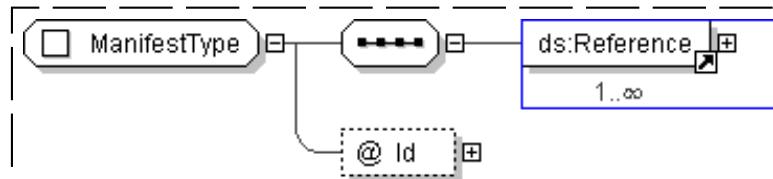
### XML Instance Representation

```

<...
  Id="ID [0..1]">
    <ds:Reference> ... </ds:Reference> [1..*]
  </...>

```

### Diagram



### Schema Component Representation

```

<complexType name="ManifestType">
  <sequence>
    <element ref=" ds:Reference " maxOccurs="unbounded" />
  </sequence>
  <attribute name="Id" type=" ID " use="optional" />
</complexType>

```

[top](#)

## Complex Type: **ObjectType**

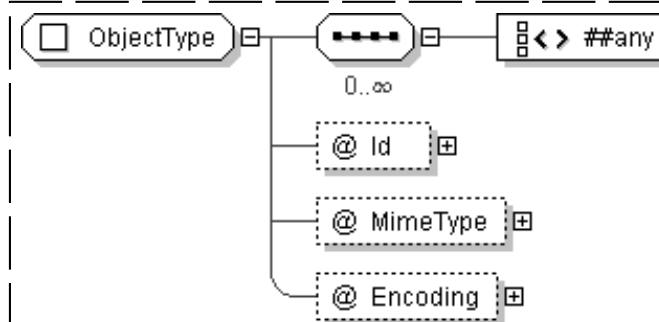
<b>Super-types:</b>	None
<b>Sub-types:</b>	None

<b>Name</b>	ObjectType
<b>Abstract</b>	no

### XML Instance Representation

```
<...
| Id="ID [0..1]"
|MimeType="string [0..1]"
|Encoding="anyURI [0..1]">
<!-- Mixed content -->
Start Sequence [0..*]
    Allow any elements from any namespace (lax validation). [1]
End Sequence
</...>
```

### Diagram



### Schema Component Representation

```

<complexType name="ObjectType" mixed="true">
    <sequence minOccurs="0" maxOccurs="unbounded">
        <any namespace="##any" processContents="lax"/>
    </sequence>
    <attribute name="Id" type="ID" use="optional"/>
    <attribute name="MimeType" type="string" use="optional"/>
    <attribute name="Encoding" type="anyURI" use="optional"/>
    <!-- add a grep facet -->
</complexType>
```

## Complex Type: PGPDataType

**Super-types:** None  
**Sub-types:** None

<b>Name</b>	PGPDataType
<b>Abstract</b>	no

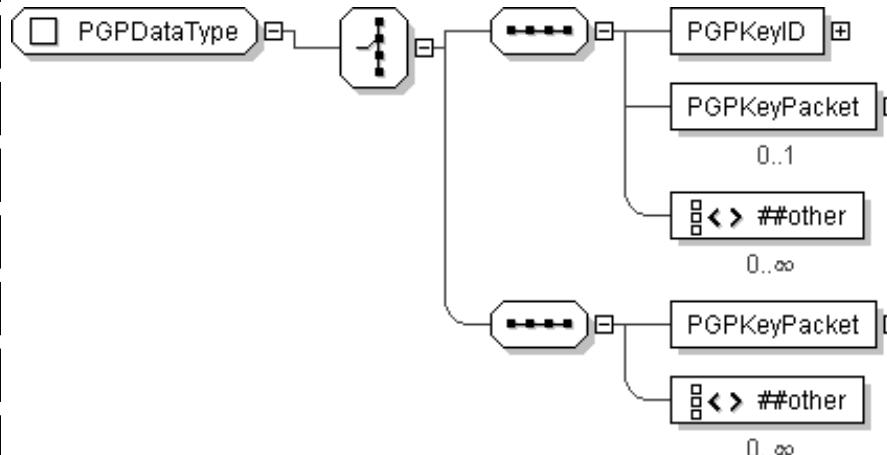
### XML Instance Representation

```

<...>
Start Choice [1]
  <ds:PGPKeyID> base64Binary </ds:PGPKeyID> [1]
  <ds:PGPKeyPacket> base64Binary </ds:PGPKeyPacket> [0..1]
    Allow any elements from a namespace other than this schema's namespace (lax validation). [0..*]
    <ds:PGPKeyPacket> base64Binary </ds:PGPKeyPacket> [1]
      Allow any elements from a namespace other than this schema's namespace (lax validation). [0..*]
End Choice
</...>

```

### Diagram



### Schema Component Representation

```

<complexType name="PGPDataType">
  <choice>
    <sequence>
      <element name="PGPKeyID" type="base64Binary" />
      <element name="PGPKeyPacket" type="base64Binary" minOccurs="0" />
      <any namespace="#other" processContents="lax" minOccurs="0" maxOccurs="unbounded" />
    </sequence>
  </choice>
</complexType>

```

```

</sequence>
<sequence>
  <element name="PGPKeyPacket" type="base64Binary" />
  <any namespace="#other" processContents="lax" minOccurs="0" maxOccurs="unbounded" />
</sequence>
</choice>
</complexType>

```

[top](#)

## Complex Type: RSAKeyValue-Type

<b>Super-types:</b>	None
<b>Sub-types:</b>	None

<b>Name</b>	RSAKeyValue-Type
<b>Abstract</b>	no

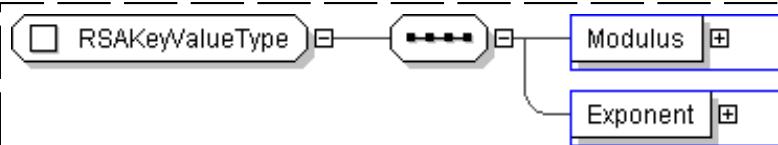
### XML Instance Representation

```

<...>
  <ds:Modulus> ds: CryptoBinary </ds:Modulus> [1]
  <ds:Exponent> ds: CryptoBinary </ds:Exponent> [1]
</...>

```

### Diagram



### Schema Component Representation

```

<complexType name="RSAKeyValue-Type">
  <sequence>
    <element name="Modulus" type="ds: CryptoBinary" />
    <element name="Exponent" type="ds: CryptoBinary" />
  </sequence>
</complexType>

```

[top](#)

## Complex Type: ReferenceType

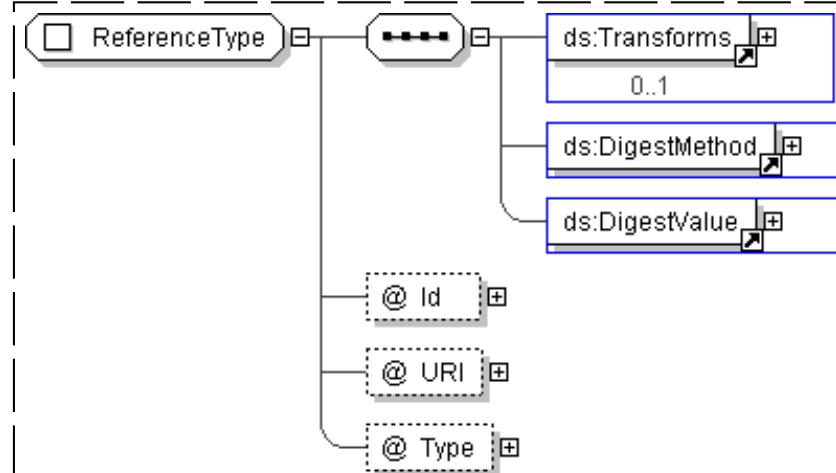
<b>Super-types:</b>	None
<b>Sub-types:</b>	None

<b>Name</b>	ReferenceType
<b>Abstract</b>	no

### XML Instance Representation

```
<...>
  Id="ID [0..1]"
  URI="anyURI [0..1]"
  Type="anyURI [0..1]>
    <ds:Transforms> ... </ds:Transforms> [0..1]
    <ds:DigestMethod> ... </ds:DigestMethod> [1]
    <ds:DigestValue> ... </ds:DigestValue> [1]
</...>
```

### Diagram



### Schema Component Representation

```

<complexType name="ReferenceType">
  <sequence>
    <element ref=" ds:Transforms " minOccurs="0"/>
    <element ref=" ds:DigestMethod "/>
    <element ref=" ds:DigestValue "/>
  </sequence>

```

```

<attribute name="Id" type="ID" use="optional"/>
<attribute name="URI" type="anyURI" use="optional"/>
<attribute name="Type" type="anyURI" use="optional"/>
</complexType>

```

[top](#)

## Complex Type: RetrievalMethodType

**Super-types:** None

**Sub-types:** None

<b>Name</b>	RetrievalMethodType
<b>Abstract</b>	no

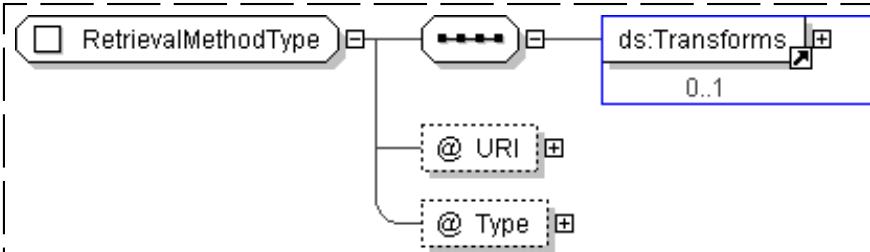
### XML Instance Representation

```

<...>
  URI="anyURI [0..1]"
  Type="anyURI [0..1]>
    <ds:Transforms> ... </ds:Transforms> [0..1]
</...>

```

### Diagram



### Schema Component Representation

```

<complexType name="RetrievalMethodType">
  <sequence>
    <element ref="ds:Transforms" minOccurs="0"/>
  </sequence>
  <attribute name="URI" type="anyURI" />
  <attribute name="Type" type="anyURI" use="optional"/>
</complexType>

```

## Complex Type: SPKIDDataType

**Super-types:** None  
**Sub-types:** None

<b>Name</b>	SPKIDDataType
<b>Abstract</b>	no

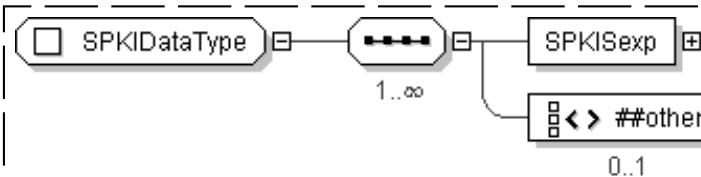
### XML Instance Representation

```

<...>
Start Sequence [1..*]
  <ds:SPKISexp> base64Binary </ds:SPKISexp> [1]
    Allow any elements from a namespace other than this schema's namespace (lax validation). [0..1]
End Sequence
</...>

```

### Diagram



### Schema Component Representation

```

<complexType name="SPKIDDataType">
  <sequence maxOccurs="unbounded">
    <element name="SPKISexp" type="base64Binary" />
    <any namespace="#other" processContents="lax" minOccurs="0" />
  </sequence>
</complexType>

```

## Complex Type: SignatureMethodType

**Super-types:** None

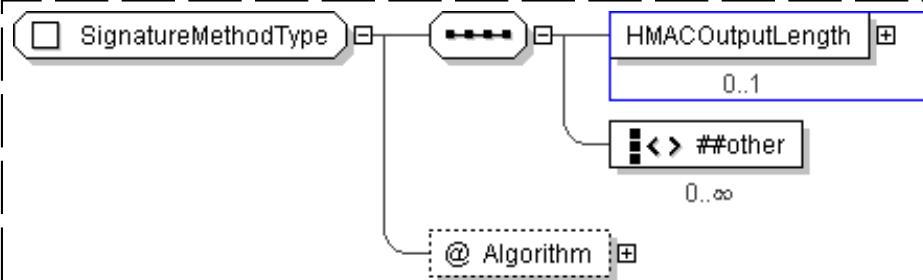
**Sub-types:** None

<b>Name</b>	SignatureMethodType
<b>Abstract</b>	no

### XML Instance Representation

```
<...>
  Algorithm="anyURI [1]">
  <!-- Mixed content -->
  <ds:MACOutputLength> ds:MACOutputLengthType </ds:MACOutputLength> [0..1]
    Allow any elements from a namespace other than this schema's namespace (strict validation).
    [0..*]
</...>
```

### Diagram



### Schema Component Representation

```

<complexType name="SignatureMethodType" mixed="true">
  <sequence>
    <element name="HMACOutputLength" type="ds:MACOutputLengthType" minOccurs="0"/>
    <any namespace="##other" minOccurs="0" maxOccurs="unbounded"/>
    <!-- (0,unbounded) elements from (1,1) external namespace -->
  </sequence>
  <attribute name="Algorithm" type="anyURI" use="required"/>
</complexType>
  
```

[top](#)

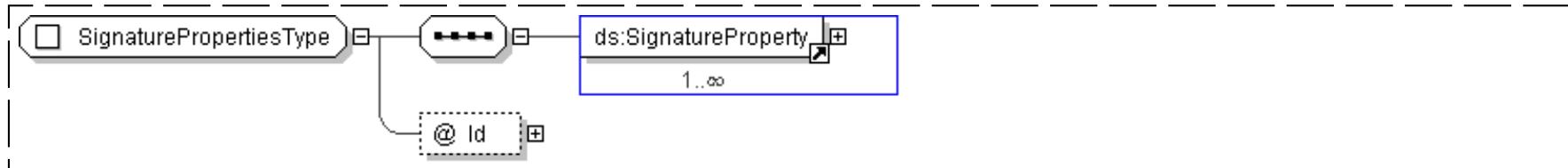
## Complex Type: `SignaturePropertiesType`

**Super-types:** None  
**Sub-types:** None

<b>Name</b>	SignaturePropertiesType
<b>Abstract</b>	no

**XML Instance Representation**

```
<...>
  <Id="ID [0..1]">
    <ds:SignatureProperty> ... </ds:SignatureProperty> [1..*]
  </...>
```

**Diagram****Schema Component Representation**

```

<complexType name="SignaturePropertiesType">
  <sequence>
    <element ref=" ds:SignatureProperty " maxOccurs="unbounded" />
  </sequence>
  <attribute name="Id" type=" ID " use="optional" />
</complexType>

```

[top](#)**Complex Type: SignaturePropertyType**

<i>Super-types:</i>	None
<i>Sub-types:</i>	None

<b>Name</b>	SignaturePropertyType
<b>Abstract</b>	no

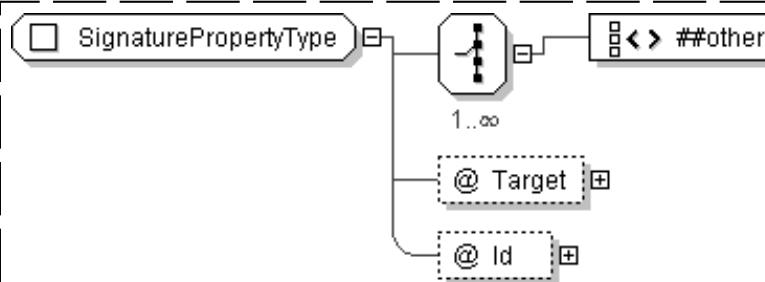
**XML Instance Representation**

```
<...>
  <Target="anyURI [1]">
  <Id="ID [0..1]">
  | <!-- Mixed content -->
```

```

| Start Choice [1..*]
|   Allow any elements from a namespace other than this schema's namespace (lax validation). [1]
| End Choice
</...>

```

**Diagram****Schema Component Representation**

```

<complexType name="Signature.PropertyType" mixed="true">
  <choice maxOccurs="unbounded">
    <any namespace="##other" processContents="lax"/>
    <!-- (1,1) elements from (1,unbounded) namespaces -->
  </choice>
  <attribute name="Target" type="anyURI" use="required"/>
  <attribute name="Id" type="ID" use="optional"/>
</complexType>

```

[top](#)**Complex Type: SignatureType**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None

<b>Name</b>	SignatureType
<b>Abstract</b>	no

**XML Instance Representation**

```

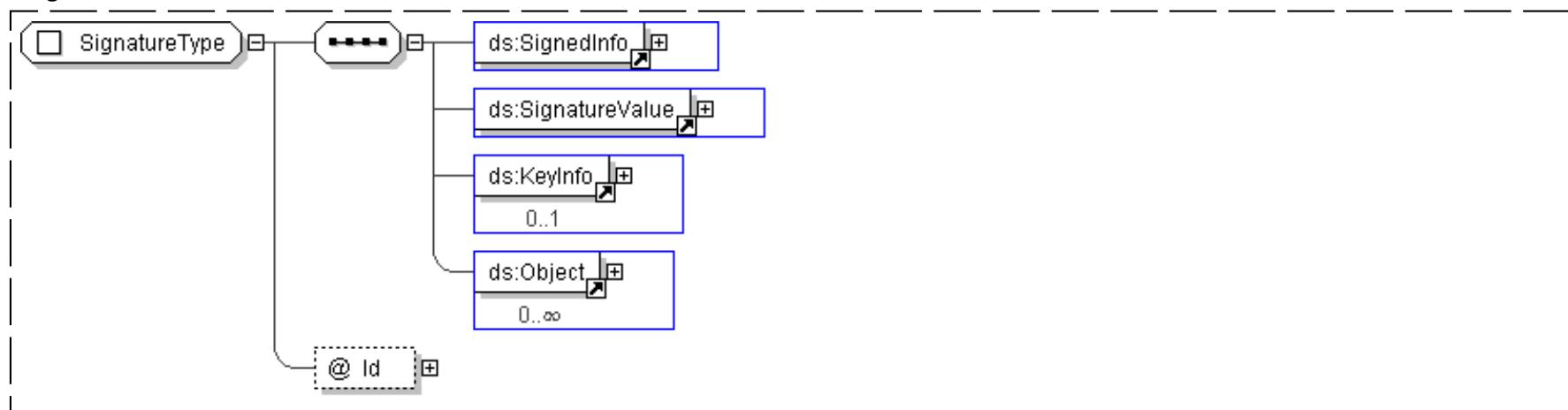
<...
  Id="ID [0..1]">
  <ds:SignedInfo> ... </ds:SignedInfo> [1]
  <ds:SignatureValue> ... </ds:SignatureValue> [1]

```

```

|   <ds:KeyInfo> ... </ds:KeyInfo> [0..1]
|   <ds:Object> ... </ds:Object> [0..*]
| </...>

```

**Diagram****Schema Component Representation**

```

<complexType name="SignatureType">
  <sequence>
    <element ref=" ds:SignedInfo " />
    <element ref=" ds:SignatureValue " />
    <element ref=" ds:KeyInfo " minOccurs="0" />
    <element ref=" ds:Object " minOccurs="0" maxOccurs="unbounded" />
  </sequence>
  <attribute name="Id" type=" ID " use="optional" />
</complexType>

```

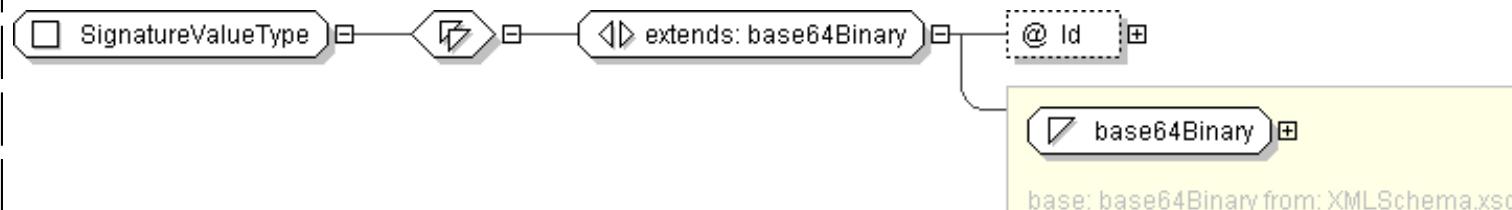
[top](#)**Complex Type: `SignatureValueType`**

<b>Super-types:</b>	base64Binary < <b>SignatureValueType</b> (by extension)
<b>Sub-types:</b>	None

Name	SignatureValueType
Abstract	no

**XML Instance Representation**

```
<...
  Id="ID [0..1]">
  base64Binary
</...>
```

**Diagram****Schema Component Representation**

```
<complexType name="SignatureValueType">
  <simpleContent>
    <extension base="base64Binary">
      <attribute name="Id" type="ID" use="optional"/>
    </extension>
  </simpleContent>
</complexType>
```

[top](#)**Complex Type: SignedInfoType**

<b>Super-types:</b>	None
---------------------	------

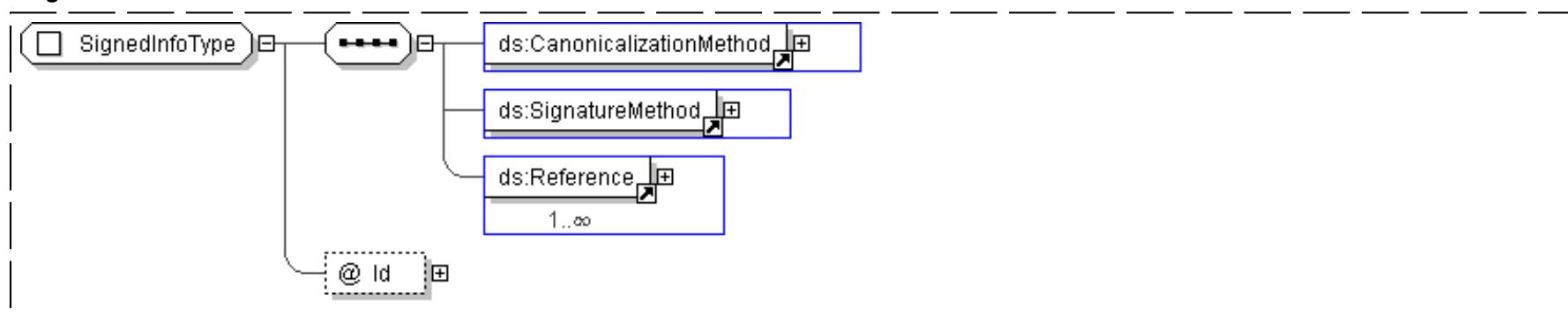
<b>Sub-types:</b>	None
-------------------	------

<b>Name</b>	SignedInfoType
-------------	----------------

<b>Abstract</b>	no
-----------------	----

**XML Instance Representation**

```
<...
  Id="ID [0..1]">
  <ds:CanonicalizationMethod> ... </ds:CanonicalizationMethod> [1]
  <ds:SignatureMethod> ... </ds:SignatureMethod> [1]
  <ds:Reference> ... </ds:Reference> [1..*]
</...>
```

**Diagram****Schema Component Representation**

```

<complexType name="SignedInfoType">
  <sequence>
    <element ref=" ds:CanonicalizationMethod " />
    <element ref=" ds:SignatureMethod " />
    <element ref=" ds:Reference " maxOccurs="unbounded" />
  </sequence>
  <attribute name="Id" type=" ID " use="optional" />
</complexType>
  
```

[top](#)**Complex Type: TransformType**

<b>Super-types:</b>	None
<b>Sub-types:</b>	None

<b>Name</b>	TransformType
<b>Abstract</b>	no

**XML Instance Representation**

```

<...
Algorithm="anyURI [1]"
<!-- Mixed content -->
Start Choice [0...*]
  Allow any elements from a namespace other than this schema's namespace (lax validation). [1]
  <ds:XPath> string </ds:XPath> [1]
End Choice
</...>
  
```

L

**Diagram****Schema Component Representation**

```

<complexType name="TransformType" mixed="true">
  <choice minOccurs="0" maxOccurs="unbounded">
    <any namespace="##other" processContents="lax"/>
    <!-- (1,1) elements from (0,unbounded) namespaces -->
    <element name="XPath" type="string" />
  </choice>
  <attribute name="Algorithm" type="anyURI" use="required"/>
</complexType>
  
```

[top](#)**Complex Type: TransformsType**

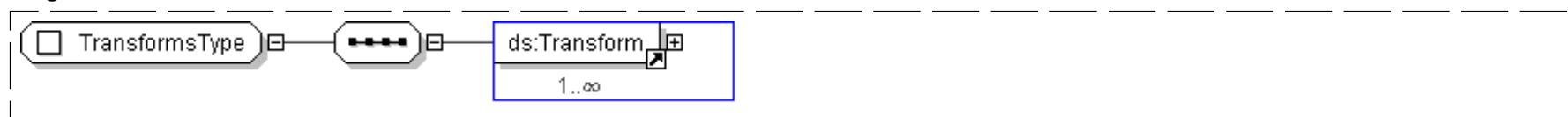
<b>Super-types:</b>	None
<b>Sub-types:</b>	None

<b>Name</b>	TransformsType
<b>Abstract</b>	no

**XML Instance Representation**

```

<...>
  <ds:Transform> ... </ds:Transform> [1..*]
</...>
  
```

**Diagram**

## Schema Component Representation

```
<complexType name="TransformsType">
  <sequence>
    <element ref=" ds:Transform " maxOccurs="unbounded" />
  </sequence>
</complexType>
```

[top](#)

## Complex Type: X509DataType

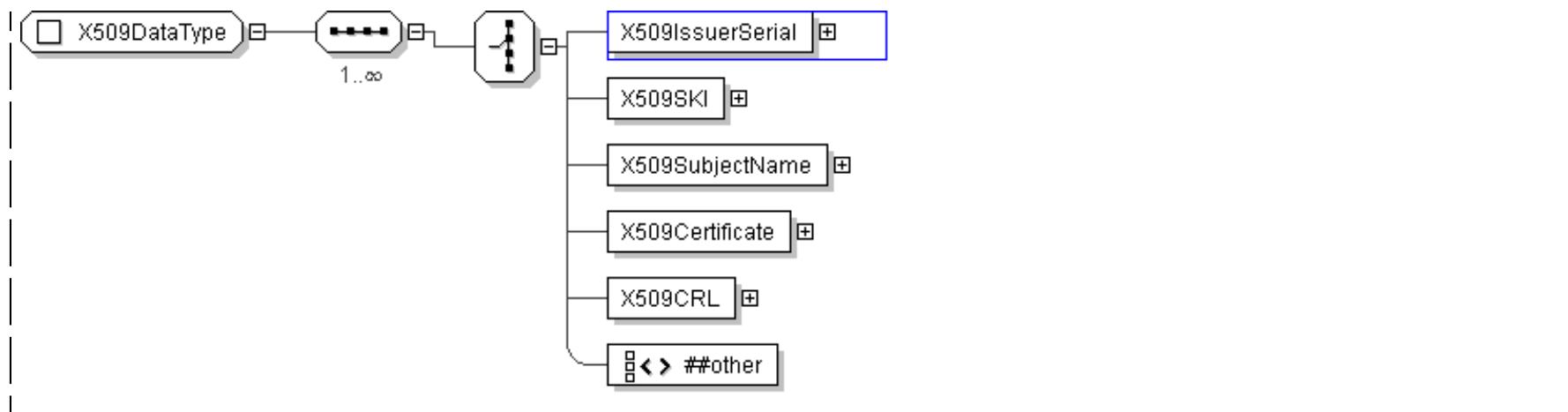
<b>Super-types:</b>	None
<b>Sub-types:</b>	None

<b>Name</b>	X509DataType
<b>Abstract</b>	no

### XML Instance Representation

```
<...>
Start Sequence [1..*]
Start Choice [1]
  <ds:X509IssuerSerial> ds:X509IssuerSerialType </ds:X509IssuerSerial> [1]
  <ds:X509SKI> base64Binary </ds:X509SKI> [1]
  <ds:X509SubjectName> string </ds:X509SubjectName> [1]
  <ds:X509Certificate> base64Binary </ds:X509Certificate> [1]
  <ds:X509CRL> base64Binary </ds:X509CRL> [1]
    Allow any elements from a namespace other than this schema's namespace (lax validation). [1]
End Choice
End Sequence
</...>
```

### Diagram



## Schema Component Representation

```

<complexType name="X509DataType">
  <sequence maxOccurs="unbounded">
    <choice>
      <element name="X509IssuerSerial" type="ds:X509IssuerSerialType" />
      <element name="X509SKI" type="base64Binary" />
      <element name="X509SubjectName" type="string" />
      <element name="X509Certificate" type="base64Binary" />
      <element name="X509CRL" type="base64Binary" />
      <any namespace="#other" processContents="lax" />
    </choice>
  </sequence>
</complexType>
  
```

[top](#)

## Complex Type: X509IssuerSerialType

<i>Super-types:</i>	None
<i>Sub-types:</i>	None

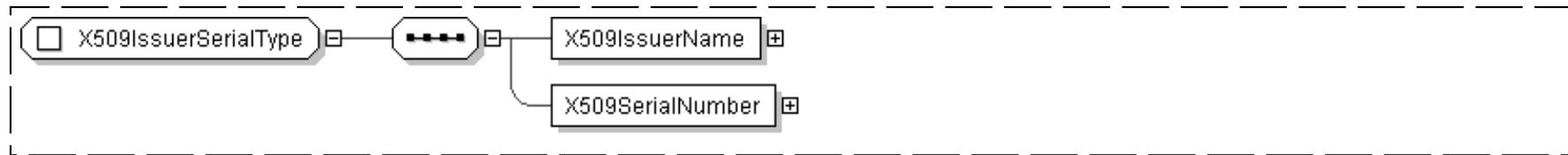
<b>Name</b>	X509IssuerSerialType
<b>Abstract</b>	no

### XML Instance Representation

```

<...>
  <ds:X509IssuerName> string </ds:X509IssuerName> [1]
  
```

```
<ds:X509SerialNumber> integer </ds:X509SerialNumber> [1]
</...>
```

**Diagram****Schema Component Representation**

```
<complexType name="X509IssuerSerialType">
  <sequence>
    <element name="X509IssuerName" type="string" />
    <element name="X509SerialNumber" type="integer" />
  </sequence>
</complexType>
```

[top](#)**Simple Type: CryptoBinary**

<b>Super-types:</b>	base64Binary < <b>CryptoBinary</b> (by restriction)
<b>Sub-types:</b>	None

<b>Name</b>	CryptoBinary
<b>Content</b>	<ul style="list-style-type: none"> <li>• Base XSD Type: base64Binary</li> </ul>

**Diagram****Schema Component Representation**

```
<simpleType name="CryptoBinary">
  <restriction base="base64Binary" />
</simpleType>
```

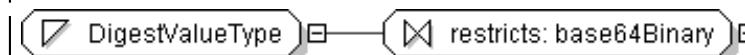
[top](#)

## Simple Type: DigestValueType

<b>Super-types:</b>	base64Binary < <b>DigestValueType</b> (by restriction)
<b>Sub-types:</b>	None

<b>Name</b>	DigestValueType
<b>Content</b>	<ul style="list-style-type: none"> <li>• Base XSD Type: base64Binary</li> </ul>

### Diagram



### Schema Component Representation

```

<simpleType name="DigestValueType">
  <restriction base=" base64Binary " />
</simpleType>
  
```

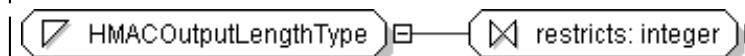
[top](#)

## Simple Type: HMACOutputLengthType

<b>Super-types:</b>	integer < <b>HMACOutputLengthType</b> (by restriction)
<b>Sub-types:</b>	None

<b>Name</b>	HMACOutputLengthType
<b>Content</b>	<ul style="list-style-type: none"> <li>• Base XSD Type: integer</li> </ul>

### Diagram



### Schema Component Representation

```

<simpleType name="HMACOutputLengthType">
  <restriction base=" integer " />
</simpleType>
  
```

[top](#)

## Legend

### Complex Type:

Schema Component Type

### AusAddress

Schema Component Name

**Super-types:** [Address](#) < AusAddress (by extension)

**Sub-types:**

- [QLDAddress](#) (by restriction)

If this schema component is a type definition, its type hierarchy is shown in a gray-bordered box.

Name	AusAddress
Abstract	no

The table above displays the properties of this schema component.

### XML Instance Representation

```
<... country="Australia" >
<unitNo> string </unitNo> [0..1]
<houseNo> string </houseNo> [1]
<street> string </street> [1]
Start Choice [1]
<city> string </city> [1]
<town> string </town> [1]
End Choice
<state> AusStates </state> [1]
<postcode> string <><pattern = [1-9][0-9]{3}></postcode> [1]
</...>
```

The XML Instance Representation table above shows the schema component's content as an XML instance.

- The minimum and maximum occurrence of elements and attributes are provided in square brackets, e.g. [0..1].
- Model group information are shown in gray, e.g. Start Choice ... End Choice.
- For type derivations, the elements and attributes that have been added to or changed from the base type's content are shown in **bold**.
- If an element/attribute has a fixed value, the fixed value is shown in green, e.g. country="Australia".
- Otherwise, the type of the element/attribute is displayed.
  - If the element/attribute's type is in the schema, a link is provided to it.
  - For local simple type definitions, the constraints are displayed in angle brackets, e.g. <><pattern = [1-9][0-9]{3}></>.

### Schema Component Representation

```

<complexType name="AusAddress">
<complexContent>
<extension base=" Address ">
<sequence>
<element name="state" type=" AusStates " />
<element name="postcode">
<simpleType>
<restriction base=" string ">
<pattern value="[1-9][0-9]{3}" />
</restriction>
</simpleType>
</element>
</sequence>
<attribute name="country" type=" string " fixed="Australia" />
</extension>
</complexContent>
</complexType>

```

The Schema Component Representation table above displays the underlying XML representation of the schema component. (Annotations are not shown.)

[top](#)

## Glossary

**Abstract** (Applies to complex type definitions and element declarations). An abstract element or complex type cannot be used to validate an element instance. If there is a reference to an abstract element, only element declarations that can substitute the abstract element can be used to validate the instance. For references to abstract type definitions, only derived types can be used.

**All Model Group** Child elements can be provided *in any order* in instances. See: <http://www.w3.org/TR/xmlschema-1/#element-all>.

**Choice Model Group** *Only one* from the list of child elements and model groups can be provided in instances. See: <http://www.w3.org/TR/xmlschema-1/#element-choice>.

**Collapse Whitespace Policy** Replace tab, line feed, and carriage return characters with space character (Unicode character 32). Then, collapse contiguous sequences of space characters into single space character, and remove leading and trailing space characters.

**Disallowed Substitutions** (Applies to element declarations). If *substitution* is specified, then *substitution group* members cannot be used in place of the given element declaration to validate element instances. If *derivation methods*, e.g. extension, restriction, are specified, then the given element declaration will not validate element instances that have types derived from the element declaration's type using the specified derivation methods. Normally, element instances can override their declaration's type by specifying an *xsi:type* attribute.

**Key Constraint** Like [Uniqueness Constraint](#), but additionally requires that the specified value(s) must be provided. See: [http://www.w3.org/TR/xmlschema-1/#clidentity-constraint\\_Definitions](http://www.w3.org/TR/xmlschema-1/#clidentity-constraint_Definitions).

**Key Reference Constraint** Ensures that the specified value(s) must match value(s) from a [Key Constraint](http://www.w3.org/TR/xmlschema-1/#clidentity-constraint_Definitions) or [Uniqueness Constraint](http://www.w3.org/TR/xmlschema-1/#clidentity-constraint_Definitions). See: [http://www.w3.org/TR/xmlschema-1/#clidentity-constraint\\_Definitions](http://www.w3.org/TR/xmlschema-1/#clidentity-constraint_Definitions).

**Model Group** Groups together element content, specifying the order in which the element content can occur and the number of times the group of element content may be repeated. See: [http://www.w3.org/TR/xmlschema-1/#Model\\_Groups](http://www.w3.org/TR/xmlschema-1/#Model_Groups).

**Nillable** (Applies to element declarations). If an element declaration is nillable, instances can use the `xsi:nil` attribute. The `xsi:nil` attribute is the boolean attribute, `nil`, from the <http://www.w3.org/2001/XMLSchema-instance> namespace. If an element instance has an `xsi:nil` attribute set to true, it can be left empty, even though its element declaration may have required content.

**Notation** A notation is used to identify the format of a piece of data. Values of elements and attributes that are of type, NOTATION, must come from the names of declared notations. See: [http://www.w3.org/TR/xmlschema-1/#cNotation\\_Declarations](http://www.w3.org/TR/xmlschema-1/#cNotation_Declarations).

**Preserve Whitespace Policy** Preserve whitespaces exactly as they appear in instances.

**Prohibited Derivations** (Applies to type definitions). Derivation methods that cannot be used to create sub-types from a given type definition.

**Prohibited Substitutions** (Applies to complex type definitions). Prevents sub-types that have been derived using the specified derivation methods from validating element instances in place of the given type definition.

**Replace Whitespace Policy** Replace tab, line feed, and carriage return characters with space character (Unicode character 32).

**Sequence Model Group** Child elements and model groups must be provided *in the specified order* in instances. See: <http://www.w3.org/TR/xmlschema-1/#element-sequence>.

**Substitution Group** Elements that are *members* of a substitution group can be used wherever the *head* element of the substitution group is referenced.

**Substitution Group Exclusions** (Applies to element declarations). Prohibits element declarations from nominating themselves as being able to substitute a given element declaration, if they have types that are derived from the original element's type using the specified derivation methods.

**Target Namespace** The target namespace identifies the namespace that components in this schema belongs to. If no target namespace is provided, then the schema components do not belong to any namespace.

**Uniqueness Constraint** Ensures uniqueness of an element/attribute value, or a combination of values, within a specified scope. See: [http://www.w3.org/TR/xmlschema-1/#clidentity-constraint\\_Definitions](http://www.w3.org/TR/xmlschema-1/#clidentity-constraint_Definitions).

[top](#)