XML Schema Documentation

Table of Contents

- Schema Document Properties
- Global Definitions
  - Complex Type: **AllegedCashflow**
  - Complex Type: **AssertedCashflow**
  - Complex Type: **AssertedPosition**
  - Complex Type: **CalculationDetails**
  - Complex Type: **CancelTradeCashflows**
  - Complex Type: **CashflowCalculationElements**
  - Complex Type: **CashflowCalculationPeriod**
  - Complex Type: **CashflowFixing**
  - Complex Type: **CashflowFixingReference**
  - Complex Type: **CashflowId**
  - Complex Type: **CashflowNotional**
  - Complex Type: **CashflowObservation**
  - Complex Type: **CashflowObservationReference**
  - Complex Type: **DefinePosition**
  - Complex Type: **GrossCashflow**
  - Complex Type: **InitialPortfolioDefinition**
  - Complex Type: **MatchId**
  - Complex Type: **PaymentId**
  - Complex Type: **PaymentMatching**
  - Complex Type: **PortfolioDefinition**
  - Complex Type: **PositionMatchResult**
  - Complex Type: **PositionMatchStatus**
  - Complex Type: **PositionProposedMatch**
  - Complex Type: **PositionReference**
  - Complex Type: **PositionsAcknowledged**
  - Complex Type: **PositionsAsserted**
  - Complex Type: **PositionsMatchResults**
  - Complex Type: **RequestPortfolio**
  - Complex Type: **StepReference**
  - Complex Type: **TradeCashflowsAsserted**
  - Complex Type: **TradeCashflowsId**
  - Complex Type: **TradeCashflowsMatchResult**
  - Complex Type: **TradeCashflowsProposedMatch**
  - Complex Type: **TradeCashflowsStatus**
  - Complex Type: **TradeDetails**
  - Complex Type: **TradeIdentifyingItems**
  - Complex Type: **TradeUnderlyer**
  - Complex Type: **TradeUnderlyerReference**
  - Complex Type: **UnderlyerReferenceUnits**
  - Complex Type: **UnprocessedPosition**
  - Model Group: **DefinitionAndCashflows.model**
  - Model Group: **IdAndTradeCashflows.model**
  - Model Group: **PositionWithoutId.model**
  - Model Group: **TradeCashflows.model**
  - Model Group: **TradeCashflowsDefinition.model**
- Legend
- Glossary

top

Schema Document Properties

Target Namespace	**http://www.fpml.org/2007/FpML-4-4**

Version	$Revision: 2709 $
Element and Attribute Namespaces	• Global element and attribute declarations belong to this schema's target namespace. • By default, local element declarations belong to this schema's target namespace. • By default, local attribute declarations have no namespace.
Schema Composition	• This schema includes components from the following schema document(s): ○ fpml-reporting-4-4.xsd

Declared Namespaces

Prefix	Namespace
Default namespace	**http://www.fpml.org/2007/FpML-4-4**
ecore	http://www.eclipse.org/emf/2002/Ecore
xml	http://www.w3.org/XML/1998/namespace
xsd	http://www.w3.org/2001/XMLSchema
fpml	**http://www.fpml.org/2007/FpML-4-4**

Schema Component Representation

```
<xsd:schema nsPrefix="fpml" package="org.fpml" documentRoot="FpML" targetNamespace="http://
www.fpml.org/2007/FpML-4-4" version="$Revision: 2709 $"
attributeFormDefault="unqualified" elementFormDefault="qualified">
   <xsd:include schemaLocation="fpml-reporting-4-4.xsd"/>
   ...
</xsd:schema>
```

top

Global Definitions

Complex Type: AllegedCashflow

Super-types:	None
Sub-types:	None

Name	AllegedCashflow
Used by (from the same schema document)	Complex Type **TradeCashflowsMatchResult**
Abstract	no

XML Instance Representation

```
<...>
   <asOfDate> xsd:dateTime </asOfDate> [0..1]
   'The date and time at which the set of cashflows was defined.'

   <tradeCashflowsId> TradeCashflowsId </tradeCashflowsId> [1]
   'Unique identifier assigned by the party asserting the set of cashflows to be reconciled.'

Start Group: TradeCashflows.model [0..1]
      <tradeIdentifyingItems> TradeIdentifyingItems </tradeIdentifyingItems> [1]
      'Structure that holds reference to the trade through the tradeId and optionally some
      trade-specific elements for identifying the trade in the case of trades that have not
      been negotiated through electronic platforms and for which the counterparty\'s trade ID has
      not been captured.'

      <adjustedPaymentDate> xsd:date </adjustedPaymentDate> [1]
      'The adjusted date in which the payments are being paid/received.'

      <payment> PaymentMatching </payment> [1..*]
      'Specifies the payment that is exposed to the matching process. Usually there will be a
      single payment but for cross-currency swaps a different payment per currency shall be provided.'
```

```
End Group: TradeCashflows.model
</...>
```

Diagram

Schema Component Representation

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

top

Complex Type: AssertedCashflow

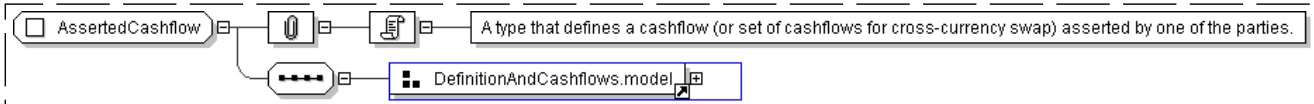
Super-types:	None
Sub-types:	None

Name	AssertedCashflow
Used by (from the same schema document)	Complex Type **TradeCashflowsMatchResult**
Abstract	no
Documentation	A type that defines a cashflow (or set of cashflows for cross-currency swap) asserted by one of the parties.

XML Instance Representation

```
<...>
   <asOfDate> xsd:dateTime </asOfDate> [0..1]
   'The date and time at which the set of cashflows was defined.'

   <tradeCashflowsId> TradeCashflowsId </tradeCashflowsId> [1]
   'Unique identifier assigned by the party asserting the set of cashflows to be reconciled.'

Start Group: TradeCashflows.model [0..1]
      <tradeIdentifyingItems> TradeIdentifyingItems </tradeIdentifyingItems> [1]
      'Structure that holds reference to the trade through the tradeId and optionally some
      trade-specific elements for identifying the trade in the case of trades that have not
      been negotiated through electronic platforms and for which the counterparty\'s trade ID has
      not been captured.'

      <adjustedPaymentDate> xsd:date </adjustedPaymentDate> [1]
      'The adjusted date in which the payments are being paid/received.'

      <payment> PaymentMatching </payment> [1..*]
      'Specifies the payment that is exposed to the matching process. Usually there will be a
      single payment but for cross-currency swaps a different payment per currency shall be provided.'

End Group: TradeCashflows.model
</...>
```

Diagram



Schema Component Representation

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

top

Complex Type: AssertedPosition

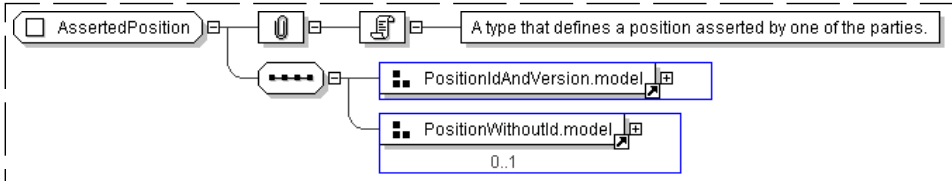
Super-types:	None
Sub-types:	None

Name	AssertedPosition
Used by (from the same schema document)	Complex Type **PositionMatchResult** , Complex Type **PositionMatchResult**
Abstract	no
Documentation	A type that defines a position asserted by one of the parties.

XML Instance Representation

```
<...>
   <positionId> PositionId </positionId> [1]
   'A version-independent identifier for the position, possibly based on trade identifier.'

   <version> xsd:positiveInteger </version> [0..1]
   'A version identifier. Version identifiers must be ascending, i.e. higher numbers imply
   newer versions. There is no requirement that version identifiers for a position be
   sequential or small, so for example timestamp-based version identifiers could be used.'

Start Group: PositionWithoutId.model [0..1]
      <reportingRoles> ReportingRoles </reportingRoles> [0..1]
      'Information about the roles of the parties with respect to reporting the positions.'

      <constituent> PositionConstituent </constituent> [1]
      'The components that create this position.'

      <scheduledDate> ScheduledDate </scheduledDate> [0..*]
      'Position level schedule date, such as final payment dates, in a simple and flexible format.'

      <valuation> AssetValuation </valuation> [0..*]
      'Valuation reported for the position, such as NPV or accrued interest. The asset/
      object references in the valuations should refer to the deal or components of the deal in
      the position, e.g. legs, streams, or underlyers.'

End Group: PositionWithoutId.model
</...>
```

Diagram



Schema Component Representation

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

```
      <xsd:group ref=" PositionWithoutId.model " minOccurs="0"/>
   </xsd:sequence>
</xsd:complexType>
```

top

Complex Type: CalculationDetails

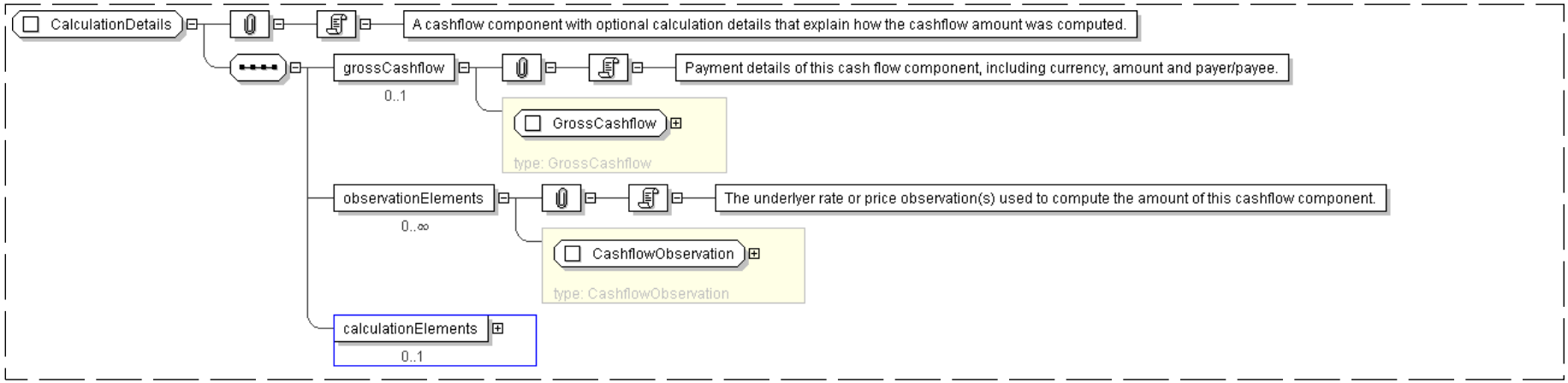
Super-types:	None
Sub-types:	None

Name	CalculationDetails
Used by (from the same schema document)	Complex Type **PaymentMatching**
Abstract	no
Documentation	A cashflow component with optional calculation details that explain how the cashflow amount was computed.

XML Instance Representation

```
<...>
   <grossCashflow> GrossCashflow </grossCashflow> [0..1]
   'Payment details of this cash flow component, including currency, amount and payer/payee.'

   <observationElements> CashflowObservation </observationElements> [0..*]
   'The underlyer rate or price observation(s) used to compute the amount of this
   cashflow component.'

   <calculationElements> CashflowCalculationElements </calculationElements> [0..1]
</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="CalculationDetails">
   <xsd:sequence>
      <xsd:element name="grossCashflow" type=" GrossCashflow " minOccurs="0"/>
      <xsd:element name="observationElements" type=" CashflowObservation "
      minOccurs="0" maxOccurs="unbounded"/>
      <xsd:element name="calculationElements" type=" CashflowCalculationElements " minOccurs="0"/>
   </xsd:sequence>
</xsd:complexType>
```

top

Complex Type: CancelTradeCashflows

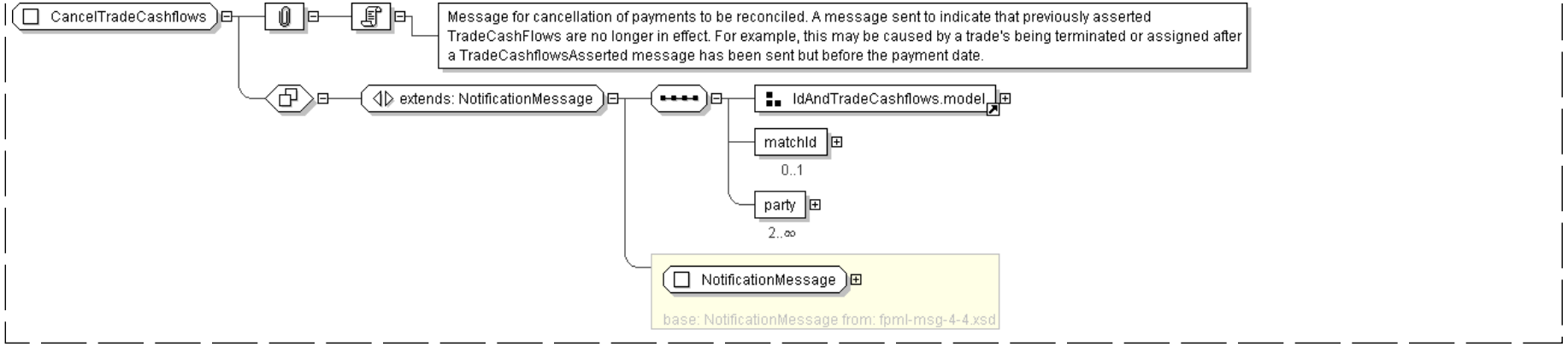
Super-types:	NotificationMessage < **CancelTradeCashflows** (by extension)
Sub-types:	None

Name	CancelTradeCashflows
Abstract	no
Documentation	Message for cancellation of payments to be reconciled. A message sent to indicate that previously asserted TradeCashFlows are no longer in effect. For example, this may be caused by a trade's being terminated or assigned after a TradeCashflowsAsserted message has been sent but before the payment date.

XML Instance Representation

```
<...
version=" xsd:token (value comes from list: {'4-0'|'4-1'|'4-2'|'4-3'|'4-4'}) [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="2 [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..*]
   <tradeCashflowsId> TradeCashflowsId </tradeCashflowsId> [1]
    'Unique identifier assigned by either party to a set of cashflows.'

Start Group: TradeCashflows.model [0..1]
      <tradeIdentifyingItems> TradeIdentifyingItems </tradeIdentifyingItems> [1]
       'Structure that holds reference to the trade through the tradeId and optionally some
       trade-specific elements for identifying the trade in the case of trades that have not
       been negotiated through electronic platforms and for which the counterparty\'s trade ID has
       not been captured.'

      <adjustedPaymentDate> xsd:date </adjustedPaymentDate> [1]
       'The adjusted date in which the payments are being paid/received.'

      <payment> PaymentMatching </payment> [1..*]
       'Specifies the payment that is exposed to the matching process. Usually there will be a
       single payment but for cross-currency swaps a different payment per currency shall be provided.'

End Group: TradeCashflows.model
   <matchId> MatchId </matchId> [0..1]
    'A unique identifier assigned by matching service to each set of matched cashflows.'

   <party> Party </party> [2..*]
    'One party element for each of the principal parties and any other party that is referenced.'

</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="CancelTradeCashflows">
   <xsd:complexContent>
      <xsd:extension base=" NotificationMessage ">
         <xsd:sequence>
            <xsd:group ref=" IdAndTradeCashflows.model "/>
            <xsd:element name="matchId" type=" MatchId " minOccurs="0"/>
            <xsd:element name="party" type=" Party " minOccurs="2" maxOccurs="unbounded"/>
         </xsd:sequence>
      </xsd:extension>
   </xsd:complexContent>
</xsd:complexType>
```

top

Complex Type: CashflowCalculationElements

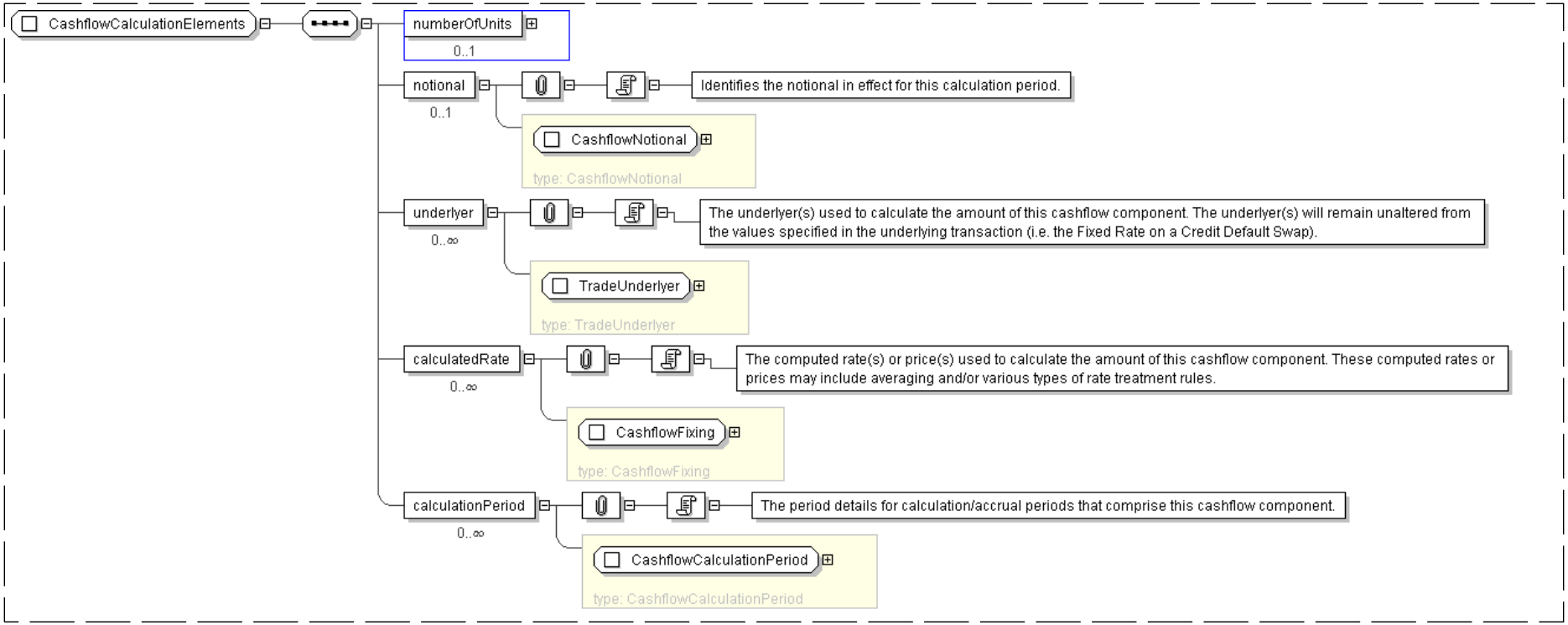
Super-types:	None
Sub-types:	None

Name	CashflowCalculationElements
Used by (from the same schema document)	Complex Type **CalculationDetails**
Abstract	no

XML Instance Representation

```
<...>
   <numberOfUnits> UnderlyerReferenceUnits </numberOfUnits> [0..1]
   <notional> CashflowNotional </notional> [0..1]
   'Identifies the notional in effect for this calculation period.'

   <underlyer> TradeUnderlyer </underlyer> [0..*]
   'The underlyer(s) used to calculate the amount of this cashflow component. The underlyer
   (s) will remain unaltered from the values specified in the underlying transaction (i.e.
   the Fixed Rate on a Credit Default Swap).'

   <calculatedRate> CashflowFixing </calculatedRate> [0..*]
   'The computed rate(s) or price(s) used to calculate the amount of this cashflow
   component. These computed rates or prices may include averaging and/or various types of
   rate treatment rules.'

   <calculationPeriod> CashflowCalculationPeriod </calculationPeriod> [0..*]
   'The period details for calculation/accrual periods that comprise this cashflow component.'

</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="CashflowCalculationElements">
   <xsd:sequence>
      <xsd:element name="numberOfUnits" type=" UnderlyerReferenceUnits " minOccurs="0"/>
      <xsd:element name="notional" type=" CashflowNotional " minOccurs="0"/>
      <xsd:element name="underlyer" type=" TradeUnderlyer " minOccurs="0" maxOccurs="unbounded"/>
      <xsd:element name="calculatedRate" type=" CashflowFixing " minOccurs="0" maxOccurs="unbounded"/>
      <xsd:element name="calculationPeriod" type=" CashflowCalculationPeriod "
      minOccurs="0" maxOccurs="unbounded"/>
   </xsd:sequence>
</xsd:complexType>
```

top

Complex Type: CashflowCalculationPeriod

Super-types:	None
Sub-types:	None

Name	CashflowCalculationPeriod
Used by (from the same schema document)	Complex Type **CashflowCalculationElements**
Abstract	no
Documentation	The period calculation details for a calculation/accrual period. This will include information about the dates and duration of the accrual period, the rate fixing (s), the notional in effect, and the amount of the accrual.

XML Instance Representation

```
<...>
```

```
<calculatedRateReference> CashflowFixingReference </calculatedRateReference> [0..*]
```

'Reference to the fixing details defined somewhere in the document.'

```
<adjustedStartDate> xsd:date </adjustedStartDate> [0..1]
```

'Date that defines the beginning of the calculation period.'

```
<adjustedEndDate> xsd:date </adjustedEndDate> [0..1]
```

'Date that defines the end of the calculation period.'

```
<numberOfDays> xsd:positiveInteger </numberOfDays> [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.'

```
<fixedRateStepReference> StepReference </fixedRateStepReference> [0..1]
```

'Reference to the fixed rate schedule\'s step in order to identify the calculation period fixed rate.'

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

'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.'

```
<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.'

```
<compoundingMethod> CompoundingMethodEnum </compoundingMethod> [0..1]
```

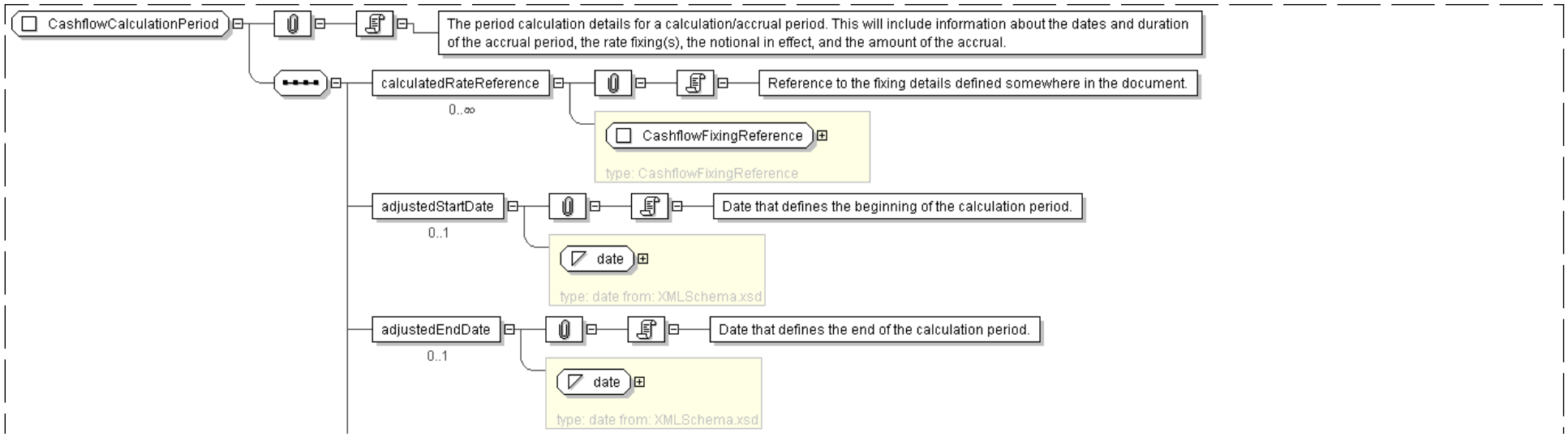
'If more that one calculation period contributes to a single payment amount this element specifies whether compounding is applicable, and if so, what compounding method is to be used. This element must only be included when more that one calculation period contributes to a single payment amount.'

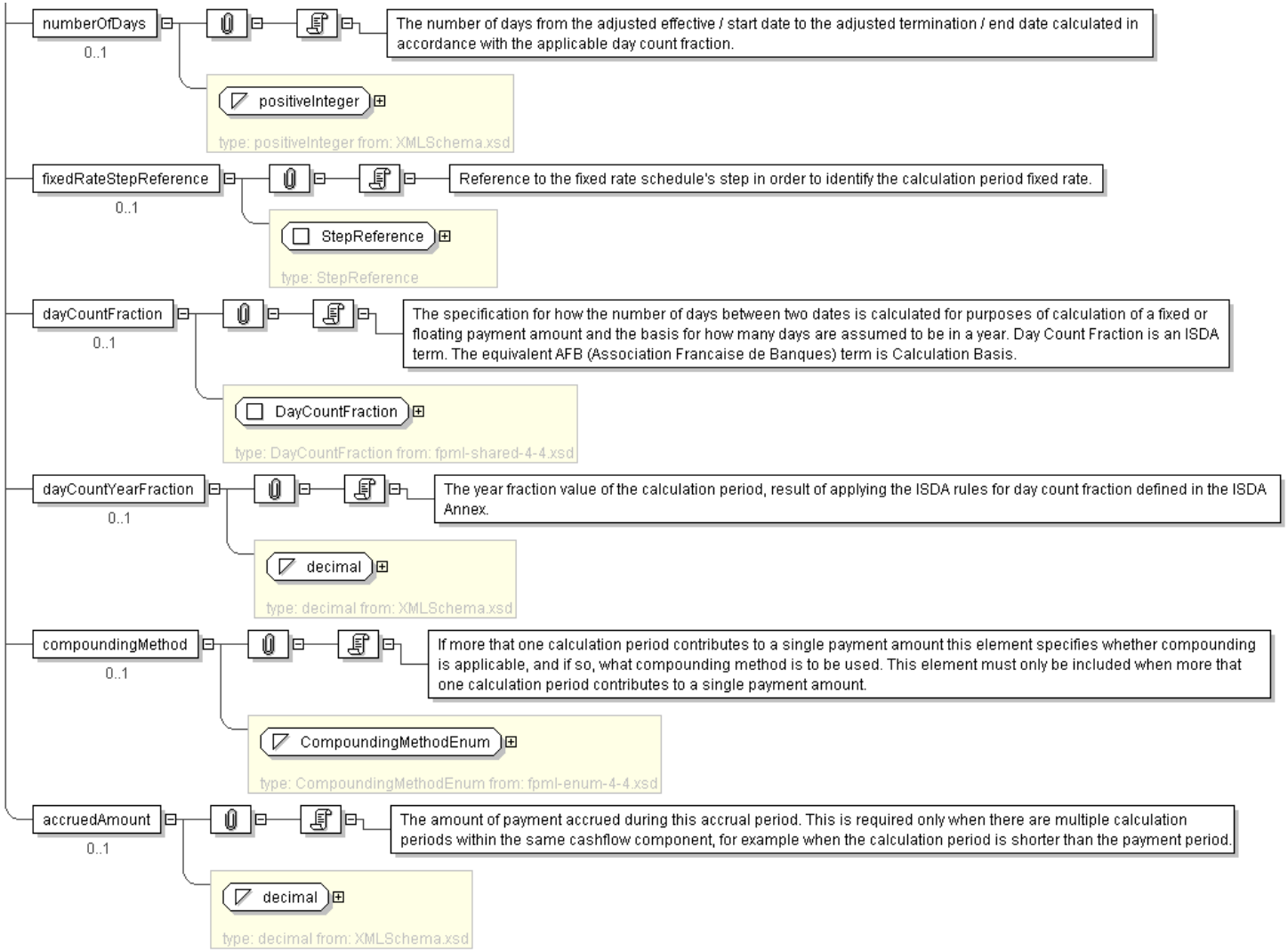
```
<accruedAmount> xsd:decimal </accruedAmount> [0..1]
```

'The amount of payment accrued during this accrual period. This is required only when there are multiple calculation periods within the same cashflow component, for example when the calculation period is shorter than the payment period.'

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

Diagram





Schema Component Representation

```
<xsd:complexType name="CashflowCalculationPeriod">
   <xsd:sequence>
      <xsd:element name="calculatedRateReference" type=" CashflowFixingReference "
      minOccurs="0" maxOccurs="unbounded"/>
      <xsd:element name="adjustedStartDate" type=" xsd:date " minOccurs="0"/>
      <xsd:element name="adjustedEndDate" type=" xsd:date " minOccurs="0"/>
      <xsd:element name="numberOfDays" type=" xsd:positiveInteger " minOccurs="0"/>
      <xsd:element name="fixedRateStepReference" type=" StepReference " minOccurs="0"/>
      <xsd:element name="dayCountFraction" type=" DayCountFraction " minOccurs="0"/>
      <xsd:element name="dayCountYearFraction" type=" xsd:decimal " minOccurs="0"/>
      <xsd:element name="compoundingMethod" type=" CompoundingMethodEnum " minOccurs="0"/>
      <xsd:element name="accruedAmount" type=" xsd:decimal " minOccurs="0"/>
   </xsd:sequence>
</xsd:complexType>
```

top

Complex Type: CashflowFixing

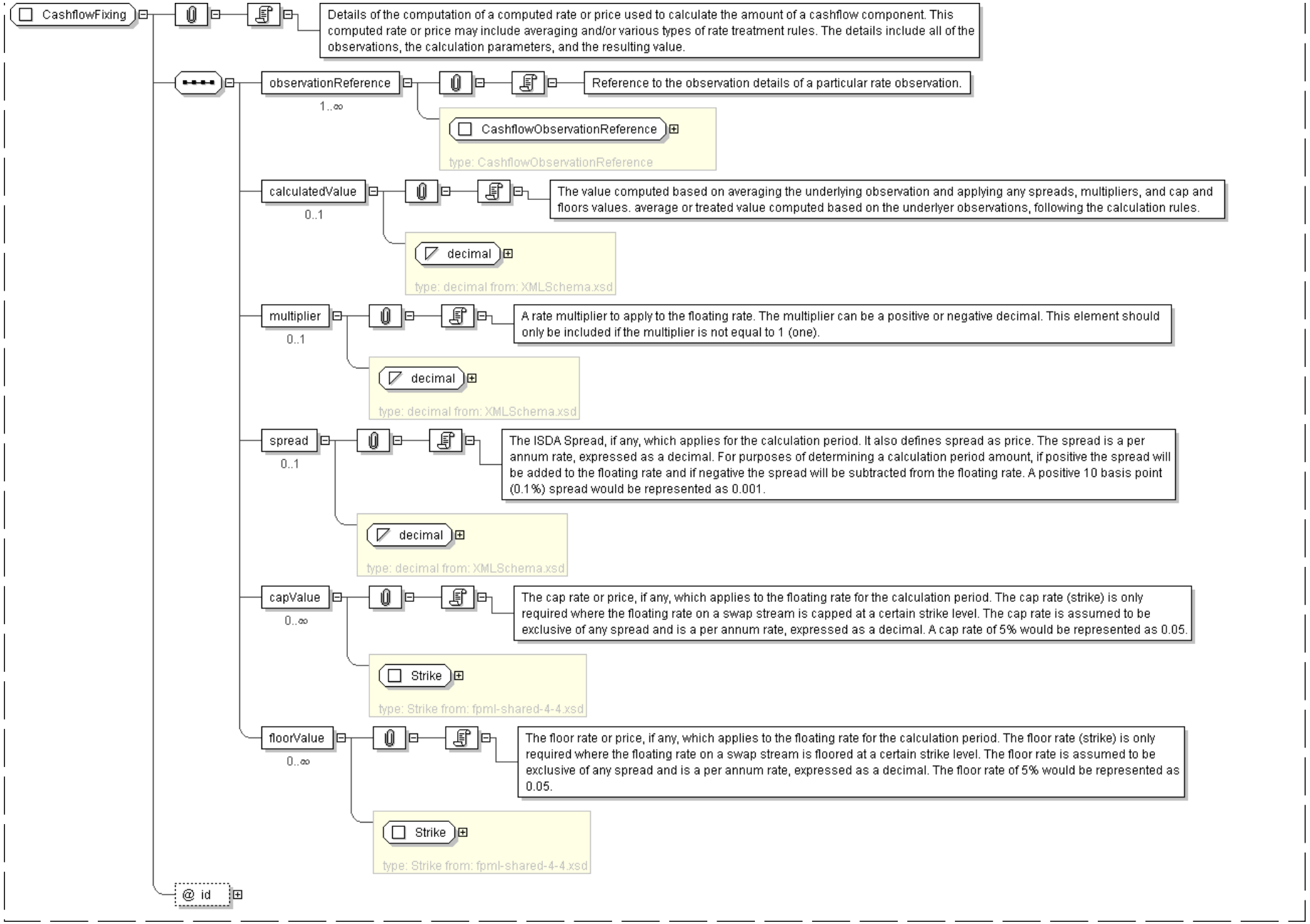
Super-types:	None
Sub-types:	None

Name	CashflowFixing
Used by (from the same schema document)	Complex Type **CashflowCalculationElements**
Abstract	no
Documentation	Details of the computation of a computed rate or price used to calculate the amount of a cashflow component. This computed rate or price may include averaging and/or various types of rate treatment rules. The details include all of the observations, the calculation parameters, and the resulting value.

XML Instance Representation

```
<...
id=" xsd:ID [0..1]">
   <observationReference> CashflowObservationReference </observationReference> [1..*]
   'Reference to the observation details of a particular rate observation.'

   <calculatedValue> xsd:decimal </calculatedValue> [0..1]
   'The value computed based on averaging the underlying observation and applying any
   spreads, multipliers, and cap and floors values. average or treated value computed based on
   the underlyer observations, following the calculation rules.'

   <multiplier> xsd:decimal </multiplier> [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. It also defines spread
   as price. 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.'

   <capValue> Strike </capValue> [0..*]
   'The cap rate or price, 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.'

   <floorValue> Strike </floorValue> [0..*]
   'The floor rate or price, 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="CashflowFixing">
   <xsd:sequence>
      <xsd:element name="observationReference" type=" CashflowObservationReference
      " maxOccurs="unbounded"/>
      <xsd:element name="calculatedValue" type=" xsd:decimal " minOccurs="0"/>
      <xsd:element name="multiplier" type=" xsd:decimal " minOccurs="0"/>
      <xsd:element name="spread" type=" xsd:decimal " minOccurs="0"/>
      <xsd:element name="capValue" type=" Strike " minOccurs="0" maxOccurs="unbounded"/>
```

```
      <xsd:element name="floorValue" type=" Strike " minOccurs="0" maxOccurs="unbounded"/>
   </xsd:sequence>
   <xsd:attribute name="id" type=" xsd:ID "/>
</xsd:complexType>
```

top

Complex Type: CashflowFixingReference

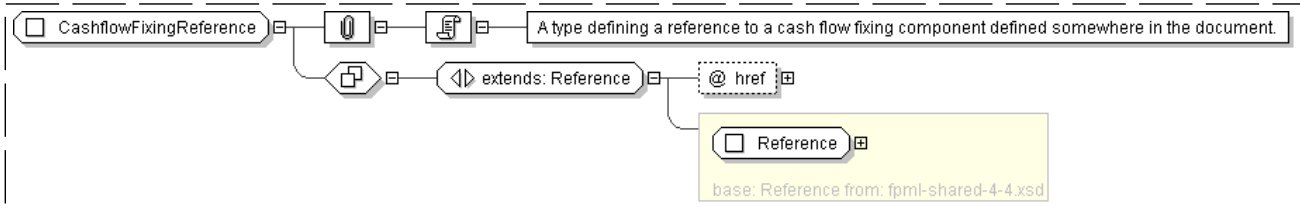
Super-types:	Reference < **CashflowFixingReference** (by extension)
Sub-types:	None

Name	CashflowFixingReference
Used by (from the same schema document)	Complex Type **CashflowCalculationPeriod**
Abstract	no
Documentation	A type defining a reference to a cash flow fixing component defined somewhere in the document.

XML Instance Representation

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

Diagram



Schema Component Representation

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

top

Complex Type: CashflowId

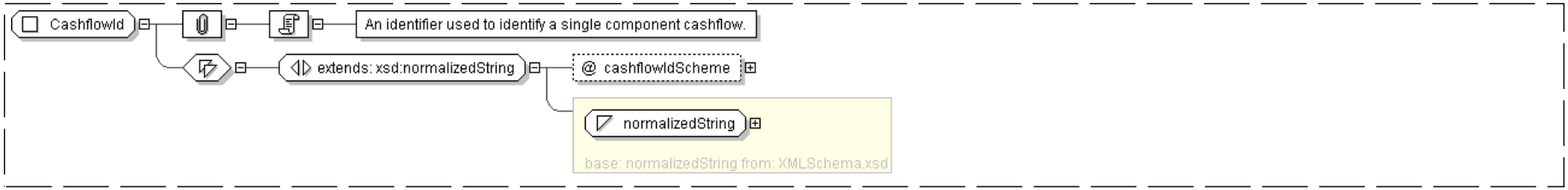
Super-types:	xsd:normalizedString < **CashflowId** (by extension)
Sub-types:	None

Name	CashflowId
Used by (from the same schema document)	Complex Type **GrossCashflow**
Abstract	no
Documentation	An identifier used to identify a single component cashflow.

XML Instance Representation

```
<...
cashflowIdScheme=" xsd:anyURI [0..1]">
xsd:normalizedString
</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="CashflowId">
   <xsd:simpleContent>
      <xsd:extension base=" xsd:normalizedString ">
         <xsd:attribute name="cashflowIdScheme" type=" xsd:anyURI "/>
      </xsd:extension>
   </xsd:simpleContent>
</xsd:complexType>
```

top

Complex Type: CashflowNotional

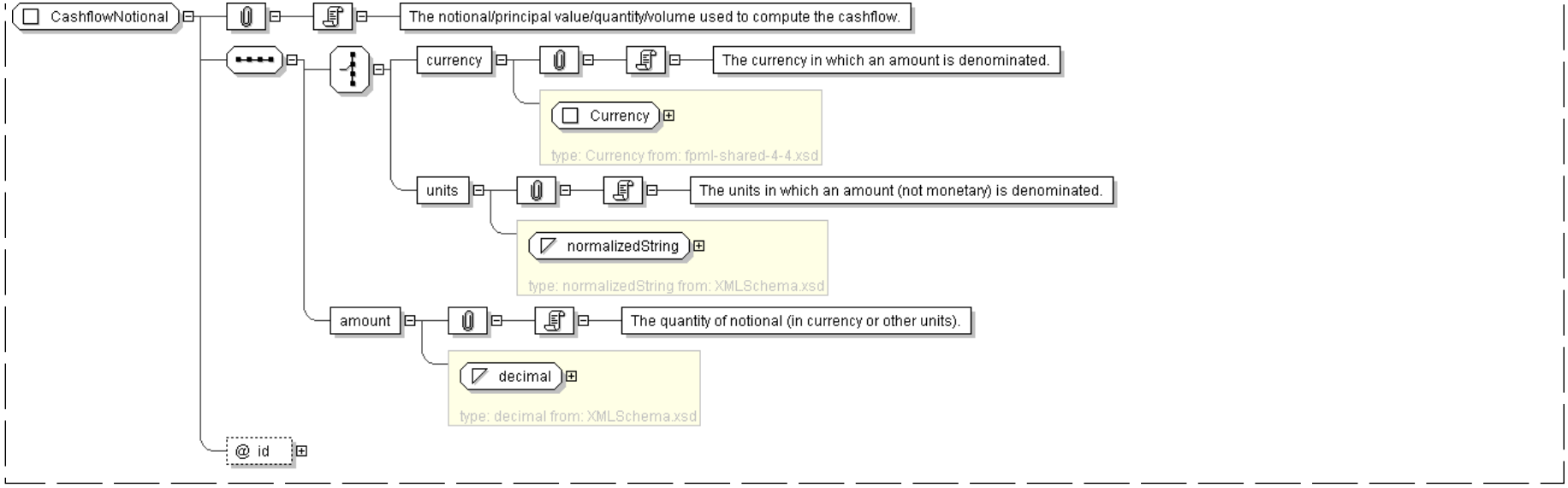
Super-types:	None
Sub-types:	None

Name	CashflowNotional
Used by (from the same schema document)	Complex Type **CashflowCalculationElements** , Complex Type **TradeDetails**
Abstract	no
Documentation	The notional/principal value/quantity/volume used to compute the cashflow.

XML Instance Representation

```
<...
id=" xsd:ID [0..1]">
Start Choice [1]
      <currency> Currency </currency> [1]
      'The currency in which an amount is denominated.'

      <units> xsd:normalizedString </units> [1]
      'The units in which an amount (not monetary) is denominated.'

End Choice
   <amount> xsd:decimal </amount> [1]
   'The quantity of notional (in currency or other units).'

</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="CashflowNotional">
   <xsd:sequence>
      <xsd:choice>
         <xsd:element name="currency" type=" Currency "/>
         <xsd:element name="units" type=" xsd:normalizedString "/>
      </xsd:choice>
      <xsd:element name="amount" type=" xsd:decimal "/>
   </xsd:sequence>
   <xsd:attribute name="id" type=" xsd:ID "/>
</xsd:complexType>
```

top

Complex Type: CashflowObservation

Super-types:	None
Sub-types:	None

Name	CashflowObservation
Used by (from the same schema document)	Complex Type **CalculationDetails**
Abstract	no
Documentation	An observation of a rate or a price of an underlyer used in the computation of a cash flow amount.

XML Instance Representation

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

'The underlyer whose rate or price is observed. Reference to an underlyer defined within the calculationElements structure.'

```
   <underlyingAsset> ... </underlyingAsset> [0..1]
```

'In cases where the underlying index is observed by observing the value of a specific security different from the index (typically a futures price), the specific security whose price was observed. For example, the underlying index might be NYMEX Crude Oil, and the underlying asset whose price is observed on a particular day might be CLU7. The index is specified via the underlyerReference, while the specific asset is specified via

the underlyingAsset.'

<observationDate> xsd:date </observationDate> [1]

'The date when the rate is observed. Corresponds to adjustedFixingDate on the Interest Rate Derivatives subschema.'

<observedValue> BasicQuotation </observedValue> [0..1]

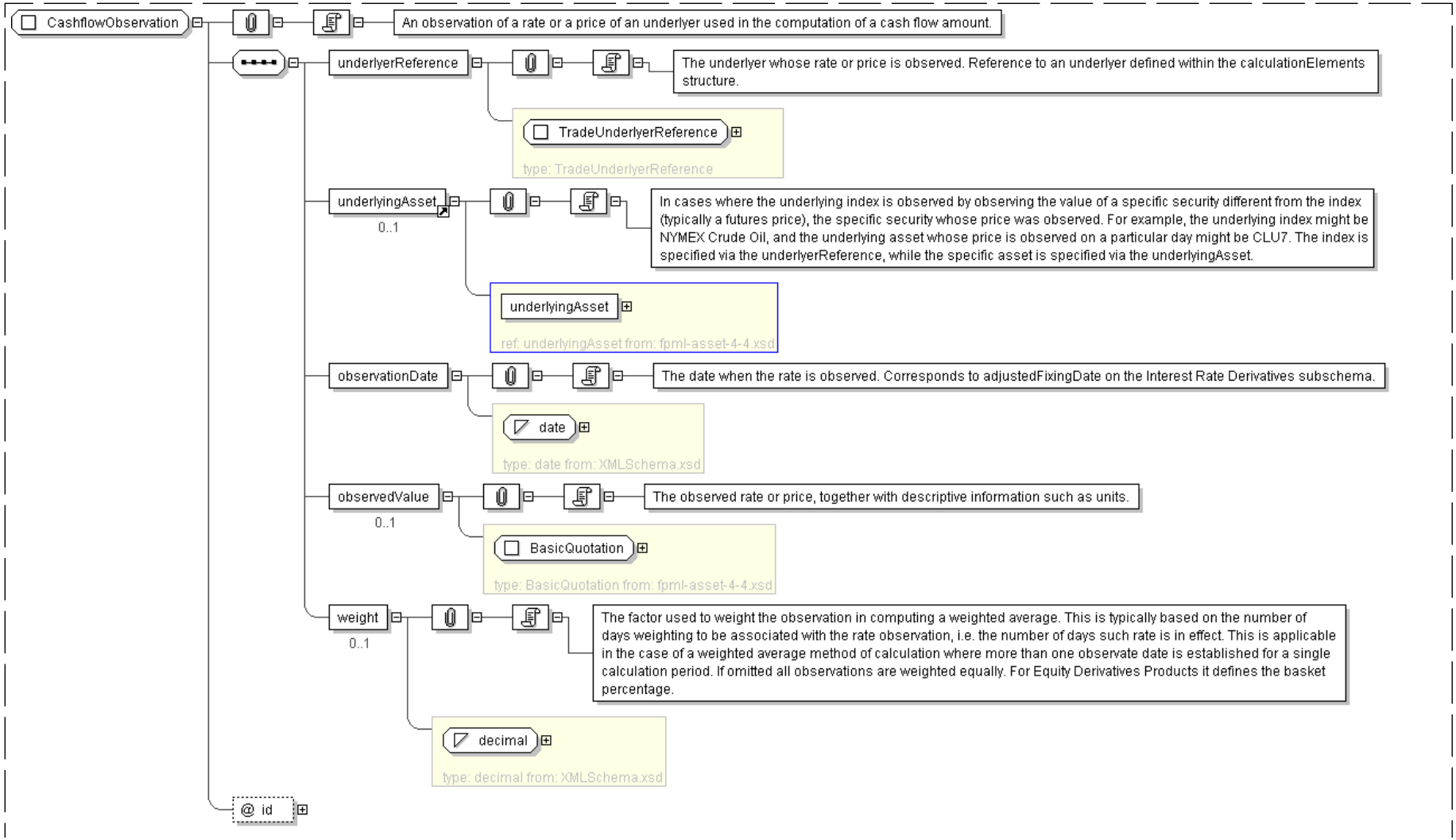
'The observed rate or price, together with descriptive information such as units.'

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

'The factor used to weight the observation in computing a weighted average. This is typically based on 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 observate date is established for a single calculation period. If omitted all observations are weighted equally. For Equity Derivatives Products it defines the basket percentage.'

</...>

Diagram



Schema Component Representation

```
<xsd:complexType name="CashflowObservation">
   <xsd:sequence>
      <xsd:element name="underlyerReference" type=" TradeUnderlyerReference "/>
      <xsd:element ref=" underlyingAsset " minOccurs="0"/>
      <xsd:element name="observationDate" type=" xsd:date "/>
      <xsd:element name="observedValue" type=" BasicQuotation " minOccurs="0"/>
      <xsd:element name="weight" type=" xsd:decimal " minOccurs="0"/>
   </xsd:sequence>
   <xsd:attribute name="id" type=" xsd:ID " use="optional"/>
</xsd:complexType>
```

top

Complex Type: CashflowObservationReference

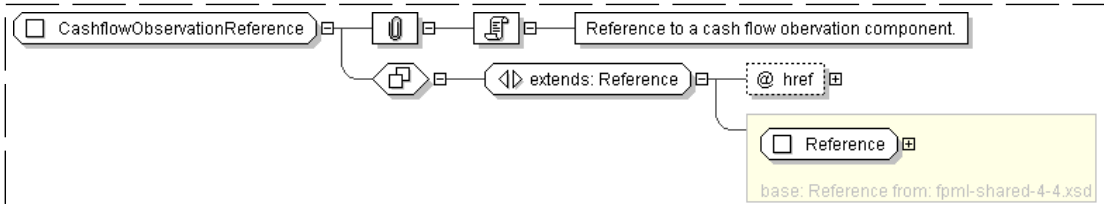
Super-types:	Reference < **CashflowObservationReference** (by extension)
Sub-types:	None

Name	CashflowObservationReference
Used by (from the same schema document)	Complex Type **CashflowFixing**
Abstract	no
Documentation	Reference to a cash flow obervation component.

XML Instance Representation

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

Diagram



Schema Component Representation

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

top

Complex Type: DefinePosition

Super-types:	Position < **DefinePosition** (by extension)
Sub-types:	None

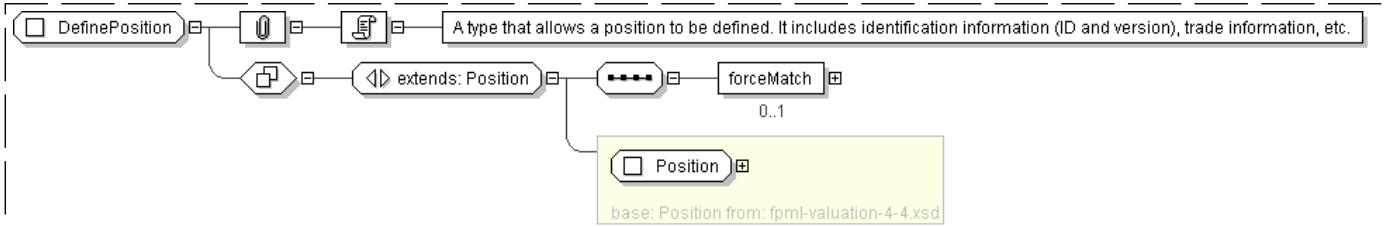
Name	DefinePosition
Used by (from the same schema document)	Complex Type **PositionsAsserted** , Complex Type **PositionsAsserted**

Abstract	no
Documentation	A type that allows a position to be defined. It includes identification information (ID and version), trade information, etc.

XML Instance Representation

```
<...
id=" xsd:ID [0..1]">
   <positionId> PositionId </positionId> [1]
   'A version-independent identifier for the position, possibly based on trade identifier.'

   <version> xsd:positiveInteger </version> [0..1]
   'A version identifier. Version identifiers must be ascending, i.e. higher numbers imply
   newer versions. There is no requirement that version identifiers for a position be
   sequential or small, so for example timestamp-based version identifiers could be used.'

   <reportingRoles> ReportingRoles </reportingRoles> [0..1]
   'Information about the roles of the parties with respect to reporting the positions.'

   <constituent> PositionConstituent </constituent> [1]
   'The components that create this position.'

   <scheduledDate> ScheduledDate </scheduledDate> [0..*]
   'Position level schedule date, such as final payment dates, in a simple and flexible format.'

   <valuation> AssetValuation </valuation> [0..*]
   'Valuation reported for the position, such as NPV or accrued interest. The asset/
   object references in the valuations should refer to the deal or components of the deal in
   the position, e.g. legs, streams, or underlyers.'

   <forceMatch> PositionReference </forceMatch> [0..1]
   'An optional reference to a position supplied by the matching party that is known to match
   this one.'

</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="DefinePosition">
   <xsd:complexContent>
      <xsd:extension base=" Position ">
         <xsd:sequence>
            <xsd:element name="forceMatch" type=" PositionReference " minOccurs="0"/>
         </xsd:sequence>
      </xsd:extension>
   </xsd:complexContent>
</xsd:complexType>
```

top

Complex Type: GrossCashflow

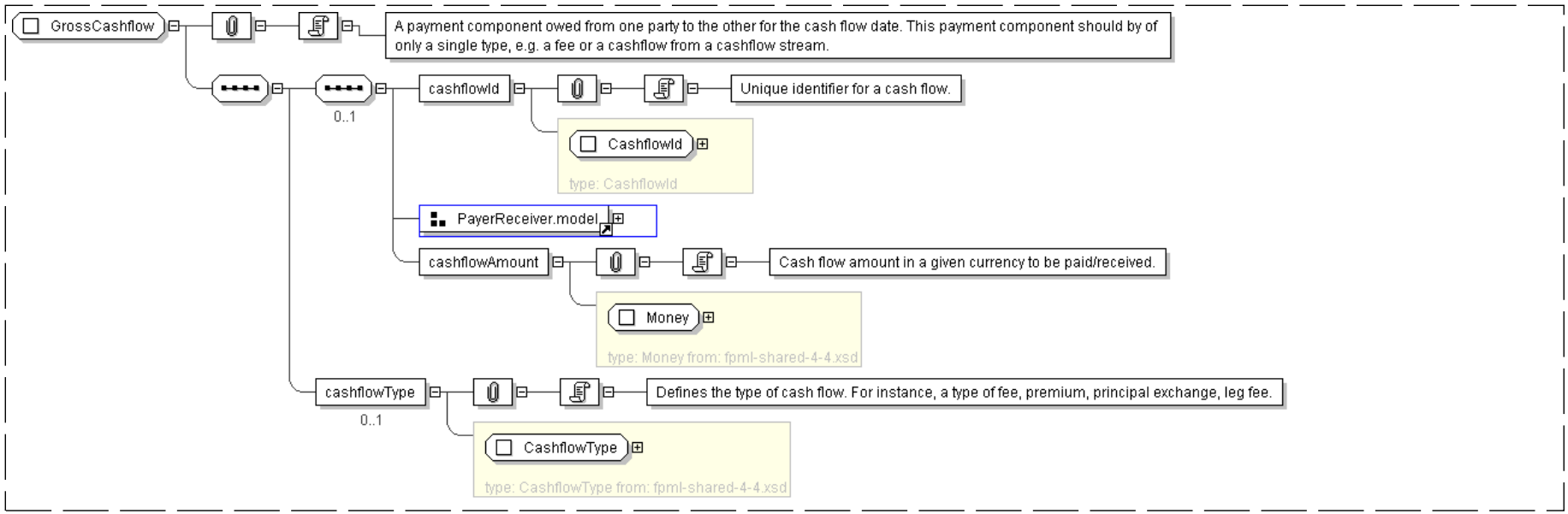
Super-types:	None
Sub-types:	None

Name	GrossCashflow
Used by (from the same schema document)	Complex Type **CalculationDetails**
Abstract	no
Documentation	A payment component owed from one party to the other for the cash flow date. This payment component should by 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> PartyOrAccountReference </payerPartyReference> [1]
      'A reference to the party responsible for making the payments defined by this structure.'

      <receiverPartyReference> PartyOrAccountReference </receiverPartyReference> [1]
      'A reference to the party 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.'

</...>
```

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: InitialPortfolioDefinition

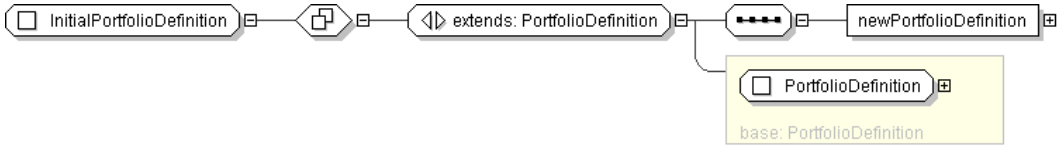
Super-types:	PortfolioDefinition < **InitialPortfolioDefinition** (by extension)
Sub-types:	None

Name	InitialPortfolioDefinition
Used by (from the same schema document)	Complex Type **PositionsAsserted**
Abstract	no

XML Instance Representation

```
<...>
   <portfolioName> xsd:normalizedString </portfolioName> [1]
   <asOfDate> xsd:date </asOfDate> [1]
   <definingParty> PartyReference </definingParty> [1]
   <matchingParty> PartyReference </matchingParty> [0..1]
   <newPortfolioDefinition> xsd:boolean </newPortfolioDefinition> [1]
   'Indicates whether it\'s a definition of a new portfolio (true) or an update to an existing
   one (false).'
</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="InitialPortfolioDefinition">
   <xsd:complexContent>
      <xsd:extension base=" PortfolioDefinition ">
         <xsd:sequence>
            <xsd:element name="newPortfolioDefinition" type=" xsd:boolean "/>
         </xsd:sequence>
      </xsd:extension>
   </xsd:complexContent>
</xsd:complexType>
```

top

Complex Type: MatchId

Super-types:	xsd:normalizedString < **MatchId** (by extension)
Sub-types:	None

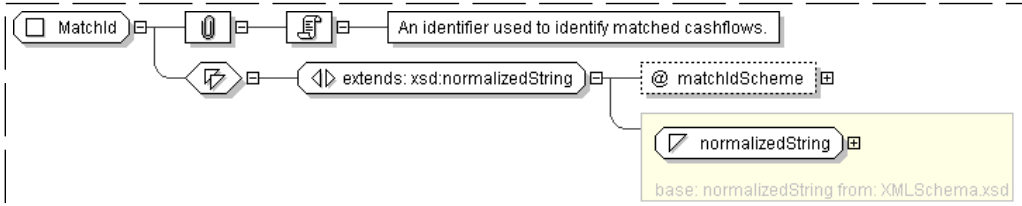
Name	MatchId

Used by (from the same schema document)	Complex Type **CancelTradeCashflows** , Complex Type **PositionProposedMatch** , Complex Type **TradeCashflowsAsserted** , Complex Type **TradeCashflowsProposedMatch**
Abstract	no
Documentation	An identifier used to identify matched cashflows.

XML Instance Representation

```
<...
matchIdScheme=" xsd:anyURI [0..1]">
xsd:normalizedString
</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="MatchId">
   <xsd:simpleContent>
      <xsd:extension base=" xsd:normalizedString ">
         <xsd:attribute name="matchIdScheme" type=" xsd:anyURI "/>
      </xsd:extension>
   </xsd:simpleContent>
</xsd:complexType>
```

top

Complex Type: PaymentId

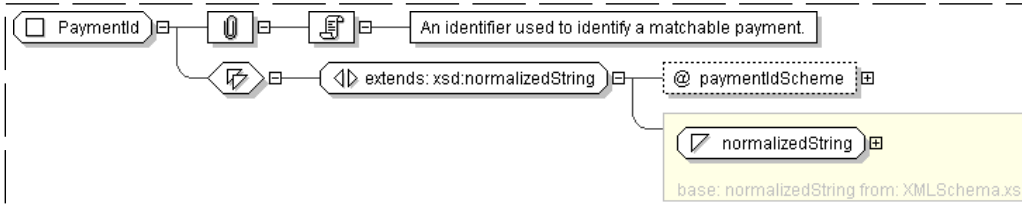
Super-types:	xsd:normalizedString < **PaymentId** (by extension)
Sub-types:	None

Name	PaymentId
Used by (from the same schema document)	Complex Type **PaymentMatching**
Abstract	no
Documentation	An identifier used to identify a matchable payment.

XML Instance Representation

```
<...
paymentIdScheme=" xsd:anyURI [0..1]">
xsd:normalizedString
</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="PaymentId">
   <xsd:simpleContent>
      <xsd:extension base=" xsd:normalizedString ">
         <xsd:attribute name="paymentIdScheme" type=" xsd:anyURI "/>
      </xsd:extension>
   </xsd:simpleContent>
</xsd:complexType>
```

top

Complex Type: PaymentMatching

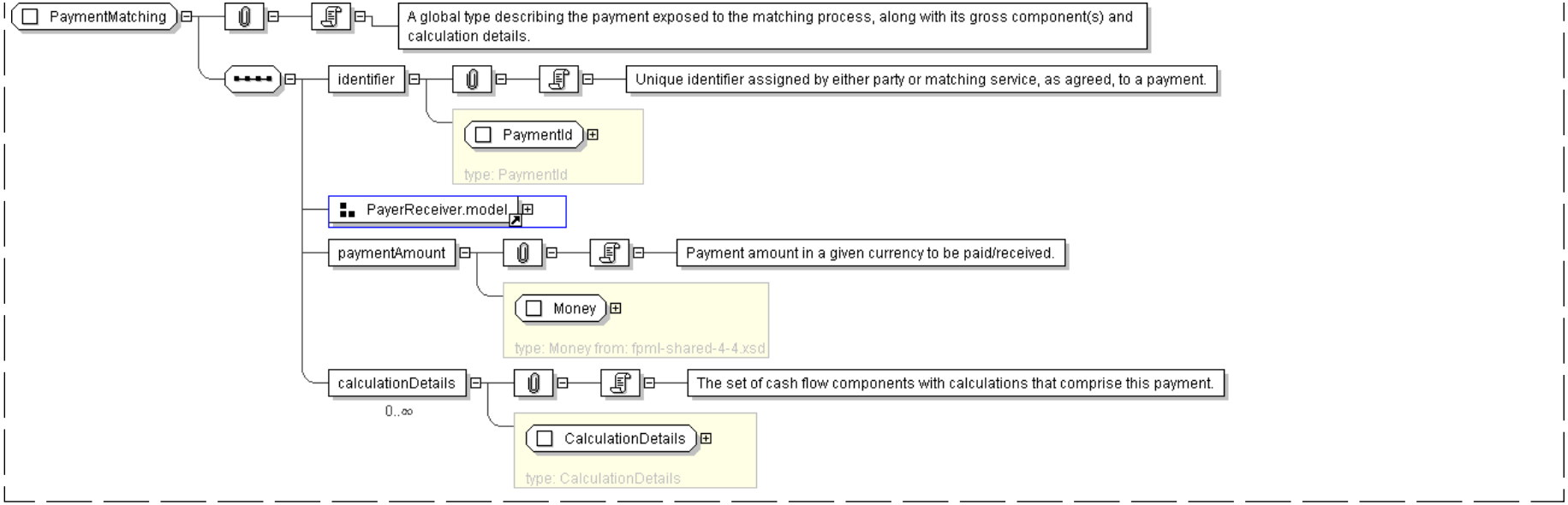
Super-types:	None
Sub-types:	None

Name	PaymentMatching
Used by (from the same schema document)	Model Group **TradeCashflows.model**
Abstract	no
Documentation	A global type describing the payment exposed to the matching process, along with its gross component(s) and calculation details.

XML Instance Representation

```
<...>
   <identifier> PaymentId </identifier> [1]
   'Unique identifier assigned by either party or matching service, as agreed, to a payment.'

   <payerPartyReference> PartyOrAccountReference </payerPartyReference> [1]
   'A reference to the party responsible for making the payments defined by this structure.'

   <receiverPartyReference> PartyOrAccountReference </receiverPartyReference> [1]
   'A reference to the party that receives the payments corresponding to this structure.'

   <paymentAmount> Money </paymentAmount> [1]
   'Payment amount in a given currency to be paid/received.'

   <calculationDetails> CalculationDetails </calculationDetails> [0..*]
   'The set of cash flow components with calculations that comprise this payment.'

</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="PaymentMatching">
   <xsd:sequence>
      <xsd:element name="identifier" type=" PaymentId "/>
      <xsd:group ref=" PayerReceiver.model "/>
      <xsd:element name="paymentAmount" type=" Money "/>
      <xsd:element name="calculationDetails" type=" CalculationDetails "
      minOccurs="0" maxOccurs="unbounded"/>
   </xsd:sequence>
</xsd:complexType>
```

top

Complex Type: PortfolioDefinition

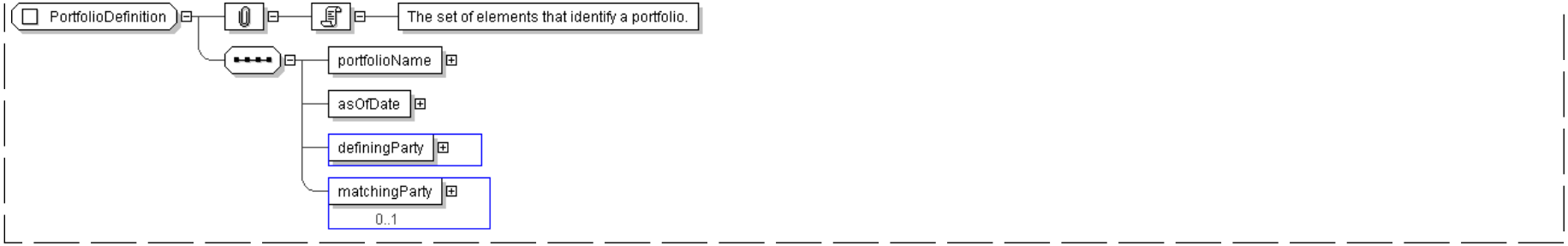
Super-types:	None
Sub-types:	• InitialPortfolioDefinition (by extension)

Name	PortfolioDefinition
Used by (from the same schema document)	Complex Type **PositionsAcknowledged** , Complex Type **PositionsMatchResults**
Abstract	no
Documentation	The set of elements that identify a portfolio.

XML Instance Representation

```
<...>
   <portfolioName> xsd:normalizedString </portfolioName> [1]
   <asOfDate> xsd:date </asOfDate> [1]
   <definingParty> PartyReference </definingParty> [1]
   <matchingParty> PartyReference </matchingParty> [0..1]
</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="PortfolioDefinition">
   <xsd:sequence>
      <xsd:element name="portfolioName" type=" xsd:normalizedString "/>
      <xsd:element name="asOfDate" type=" xsd:date "/>
      <xsd:element name="definingParty" type=" PartyReference "/>
      <xsd:element name="matchingParty" type=" PartyReference " minOccurs="0"/>
   </xsd:sequence>
</xsd:complexType>
```

top

Complex Type: PositionMatchResult

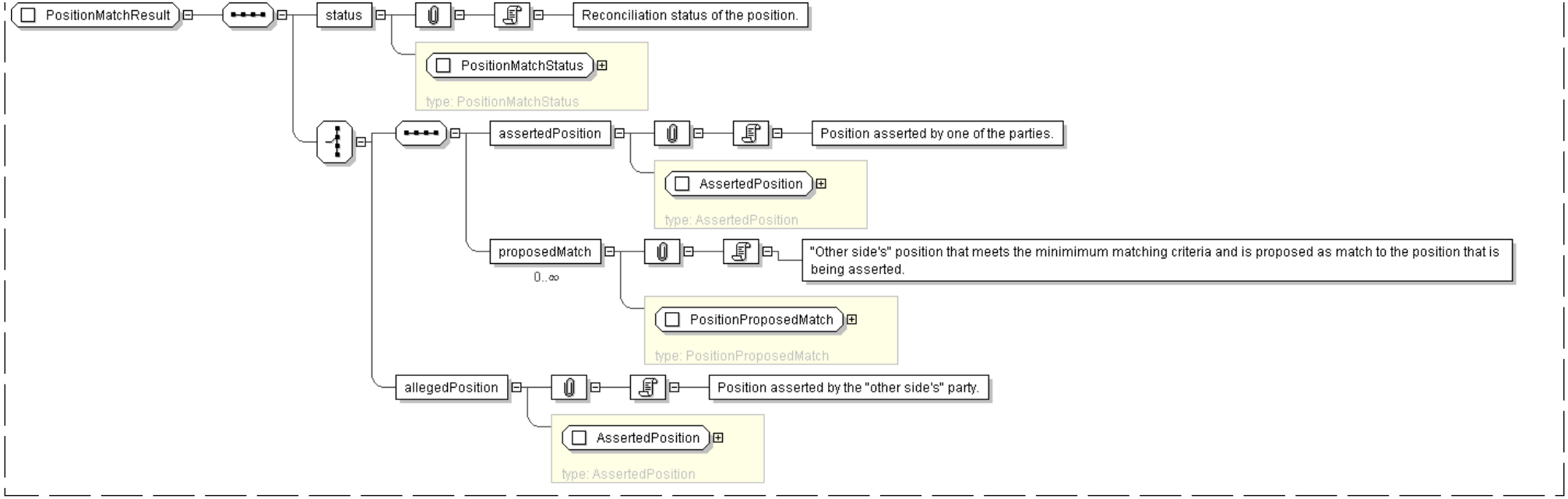
Super-types:	None
Sub-types:	None

Name	PositionMatchResult
Used by (from the same schema document)	Complex Type **PositionsMatchResults**
Abstract	no

XML Instance Representation

```
<...>
   <status> PositionMatchStatus </status> [1]
   'Reconciliation status of the position.'

Start Choice [1]
      <assertedPosition> AssertedPosition </assertedPosition> [1]
      'Position asserted by one of the parties.'

      <proposedMatch> PositionProposedMatch </proposedMatch> [0..*]
      '"Other side\'s" position that meets the minimimum matching criteria and is proposed as
      match to the position that is being asserted.'

      <allegedPosition> AssertedPosition </allegedPosition> [1]
      'Position asserted by the \"other side\'s\" party.'

End Choice
</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="PositionMatchResult">
   <xsd:sequence>
      <xsd:element name="status" type=" PositionMatchStatus "/>
      <xsd:choice>
         <xsd:sequence>
            <xsd:element name="assertedPosition" type=" AssertedPosition "/>
            <xsd:element name="proposedMatch" type=" PositionProposedMatch "
            minOccurs="0" maxOccurs="unbounded"/>
         </xsd:sequence>
         <xsd:element name="allegedPosition" type=" AssertedPosition "/>
      </xsd:choice>
   </xsd:sequence>
</xsd:complexType>
```

top

Complex Type: PositionMatchStatus

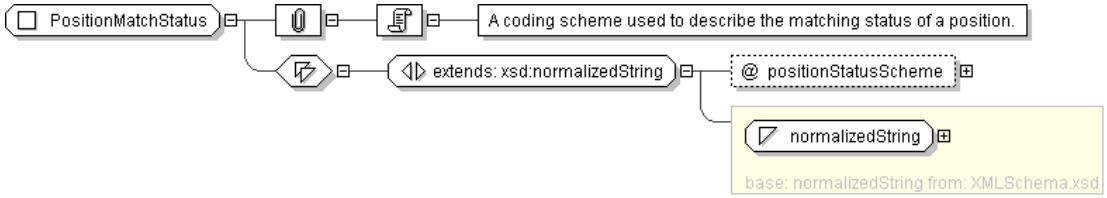
Super-types:	xsd:normalizedString < **PositionMatchStatus** (by extension)
Sub-types:	None

Name	PositionMatchStatus
Used by (from the same schema document)	Complex Type **PositionMatchResult**
Abstract	no
Documentation	A coding scheme used to describe the matching status of a position.

XML Instance Representation

```
<...
positionStatusScheme=" xsd:anyURI [0..1]">
xsd:normalizedString
</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="PositionMatchStatus">
   <xsd:simpleContent>
      <xsd:extension base=" xsd:normalizedString ">
         <xsd:attribute name="positionStatusScheme" type=" xsd:anyURI " default="http://www.fpml.
         org/coding-scheme/position-status-1-0"/>
      </xsd:extension>
   </xsd:simpleContent>
</xsd:complexType>
```

top

Complex Type: PositionProposedMatch

Super-types:	None
Sub-types:	None

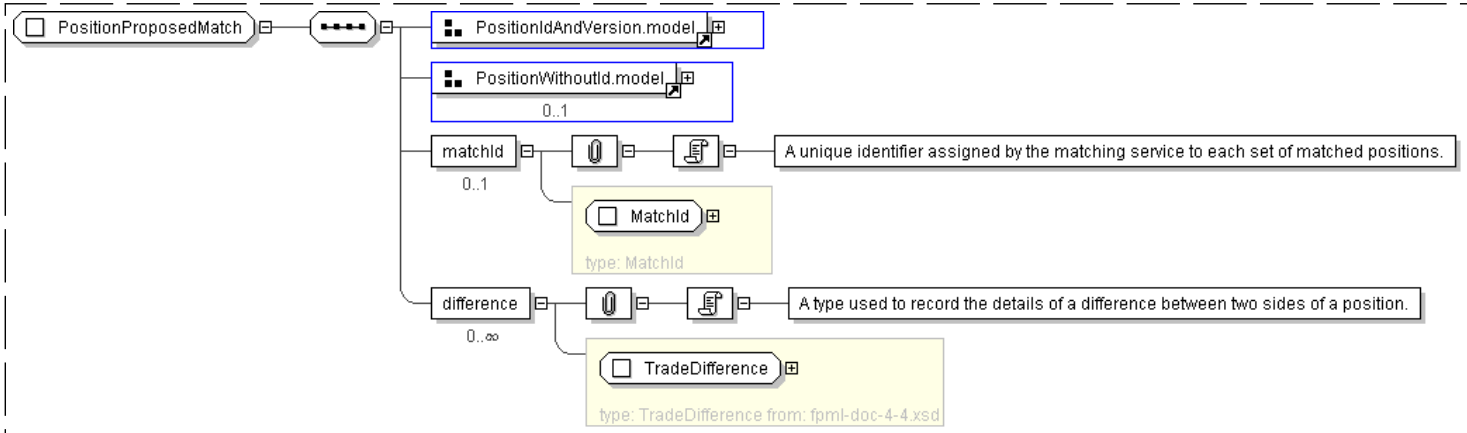
Name	PositionProposedMatch
Used by (from the same schema document)	Complex Type **PositionMatchResult**
Abstract	no

XML Instance Representation

```
<...>
   <positionId> PositionId </positionId> [1]
   'A version-independent identifier for the position, possibly based on trade identifier.'

   <version> xsd:positiveInteger </version> [0..1]
   'A version identifier. Version identifiers must be ascending, i.e. higher numbers imply
   newer versions. There is no requirement that version identifiers for a position be
   sequential or small, so for example timestamp-based version identifiers could be used.'

Start Group: PositionWithoutId.model [0..1]
      <reportingRoles> ReportingRoles </reportingRoles> [0..1]
      'Information about the roles of the parties with respect to reporting the positions.'

      <constituent> PositionConstituent </constituent> [1]
      'The components that create this position.'

      <scheduledDate> ScheduledDate </scheduledDate> [0..*]
      'Position level schedule date, such as final payment dates, in a simple and flexible format.'

      <valuation> AssetValuation </valuation> [0..*]
      'Valuation reported for the position, such as NPV or accrued interest. The asset/
      object references in the valuations should refer to the deal or components of the deal in
      the position, e.g. legs, streams, or underlyers.'

End Group: PositionWithoutId.model
   <matchId> MatchId </matchId> [0..1]
   'A unique identifier assigned by the matching service to each set of matched positions.'
```

```
    <difference> TradeDifference </difference> [0..*]
    'A type used to record the details of a difference between two sides of a position.'

</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="PositionProposedMatch">
   <xsd:sequence>
      <xsd:group ref=" PositionIdAndVersion.model "/>
      <xsd:group ref=" PositionWithoutId.model " minOccurs="0"/>
      <xsd:element name="matchId" type=" MatchId " minOccurs="0"/>
      <xsd:element name="difference" type=" TradeDifference " minOccurs="0" maxOccurs="unbounded"/>
   </xsd:sequence>
</xsd:complexType>
```

top

Complex Type: PositionReference

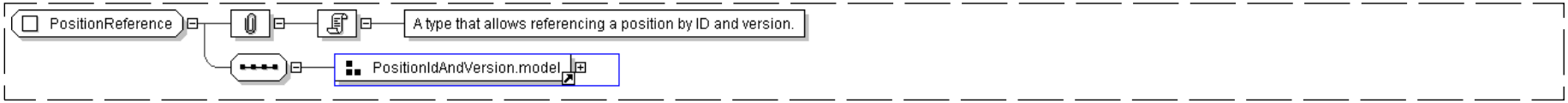
Super-types:	None
Sub-types:	None

Name	PositionReference
Used by (from the same schema document)	Complex Type **DefinePosition** , Complex Type **PositionsAcknowledged** , Complex Type **PositionsAcknowledged** , Complex Type **PositionsAsserted**
Abstract	no
Documentation	A type that allows referencing a position by ID and version.

XML Instance Representation

```
<...>
    <positionId> PositionId </positionId> [1]
    'A version-independent identifier for the position, possibly based on trade identifier.'

    <version> xsd:positiveInteger </version> [0..1]
    'A version identifier. Version identifiers must be ascending, i.e. higher numbers imply
    newer versions. There is no requirement that version identifiers for a position be
    sequential or small, so for example timestamp-based version identifiers could be used.'

</...>
```

Diagram



Schema Component Representation

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

top

Complex Type: PositionsAcknowledged

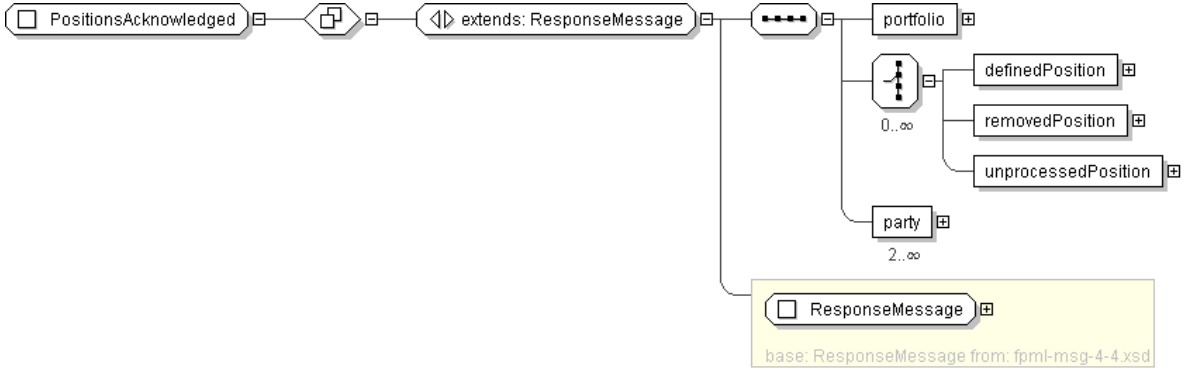
Super-types:	ResponseMessage < **PositionsAcknowledged** (by extension)
Sub-types:	None

Name	PositionsAcknowledged
Abstract	no

XML Instance Representation

```
<...
version=" xsd:token (value comes from list: {'4-0'|'4-1'|'4-2'|'4-3'|'4-4'}) [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="2 [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..*]
   <portfolio> PortfolioDefinition </portfolio> [1]
   'Contains the portfolio definition.'

Start Choice [0..*]
      <definedPosition> PositionReference </definedPosition> [1]
      <removedPosition> PositionReference </removedPosition> [1]
      <unprocessedPosition> UnprocessedPosition </unprocessedPosition> [1]
End Choice
   <party> Party </party> [2..*]
</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="PositionsAcknowledged">
   <xsd:complexContent>
      <xsd:extension base=" ResponseMessage ">
         <xsd:sequence>
            <xsd:element name="portfolio" type=" PortfolioDefinition "/>
            <xsd:choice minOccurs="0" maxOccurs="unbounded">
               <xsd:element name="definedPosition" type=" PositionReference "/>
               <xsd:element name="removedPosition" type=" PositionReference "/>
               <xsd:element name="unprocessedPosition" type=" UnprocessedPosition "/>
            </xsd:choice>
            <xsd:element name="party" type=" Party " minOccurs="2" maxOccurs="unbounded"/>
         </xsd:sequence>
      </xsd:extension>
   </xsd:complexContent>
</xsd:complexType>
```

top

Complex Type: PositionsAsserted

Super-types:	RequestMessage < **PositionsAsserted** (by extension)
Sub-types:	None

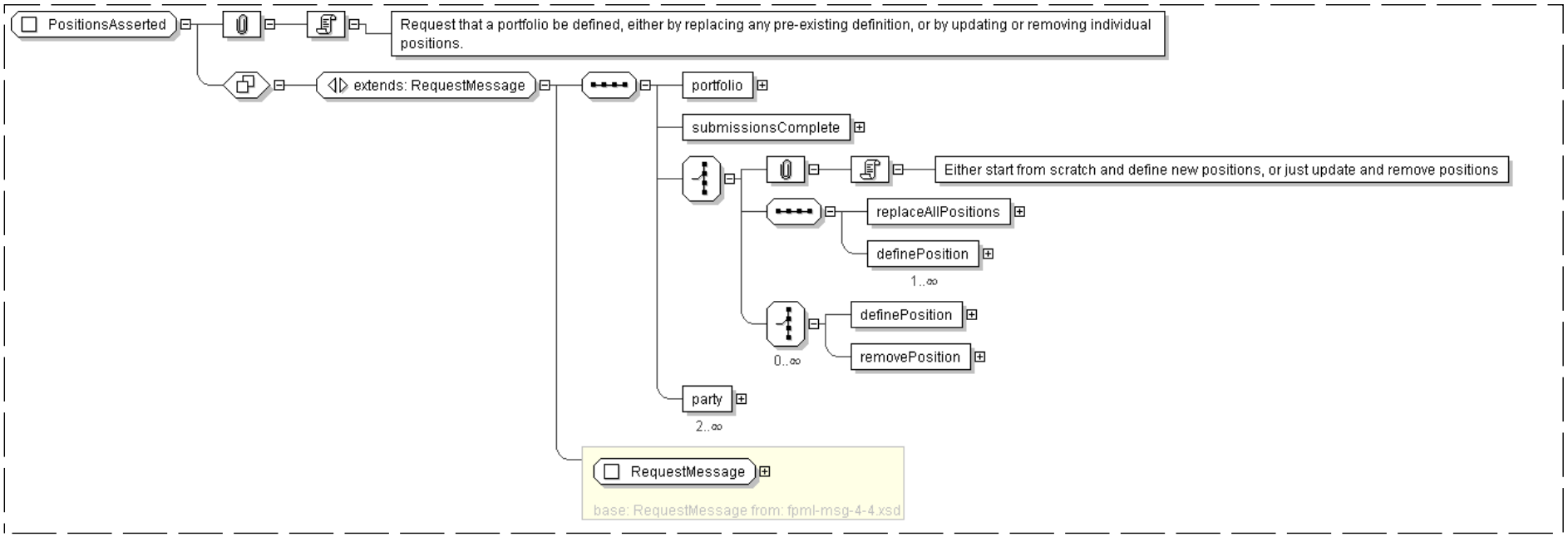
Name	PositionsAsserted
Abstract	no
Documentation	Request that a portfolio be defined, either by replacing any pre-existing definition, or by updating or removing individual positions.

XML Instance Representation

```
<...
version=" xsd:token (value comes from list: {'4-0'|'4-1'|'4-2'|'4-3'|'4-4'}) [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="2 [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..*]
   <portfolio> InitialPortfolioDefinition </portfolio> [1]
   'Contains the portfolio definition.'

   <submissionsComplete> xsd:boolean </submissionsComplete> [1]
   'Indicates whether all portfolio updates have been submitted for this as-of date'

Start Choice [1]
'Either start from scratch and define new positions, or just update and remove positions'


      <replaceAllPositions> Empty </replaceAllPositions> [1]
      'Indicates that this message replaces all previous positions for this portfolio.'

      <definePosition> DefinePosition </definePosition> [1..*]
      'Used to specify a new position.'

Start Choice [0..*]
         <definePosition> DefinePosition </definePosition> [1]
         'Used to specify a position, whether it is a new or updated position.'

         <removePosition> PositionReference </removePosition> [1]
         'Used to remove a position from the portfolio.'

End Choice
End Choice
   <party> Party </party> [2..*]
</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="PositionsAsserted">
   <xsd:complexContent>
      <xsd:extension base=" RequestMessage ">
         <xsd:sequence>
            <xsd:element name="portfolio" type=" InitialPortfolioDefinition "/>
            <xsd:element name="submissionsComplete" type=" xsd:boolean "/>
            <xsd:choice>
               <xsd:sequence>
                  <xsd:element name="replaceAllPositions" type=" Empty "/>
                  <xsd:element name="definePosition" type=" DefinePosition " maxOccurs="unbounded"/>
               </xsd:sequence>
               <xsd:choice minOccurs="0" maxOccurs="unbounded">
                  <xsd:element name="definePosition" type=" DefinePosition "/>
                  <xsd:element name="removePosition" type=" PositionReference "/>
               </xsd:choice>
            </xsd:choice>
            <xsd:element name="party" type=" Party " minOccurs="2" maxOccurs="unbounded"/>
         </xsd:sequence>
      </xsd:extension>
   </xsd:complexContent>
</xsd:complexType>
```

top

Complex Type: PositionsMatchResults

Super-types:	ResponseMessage < **PositionsMatchResults** (by extension)
Sub-types:	None

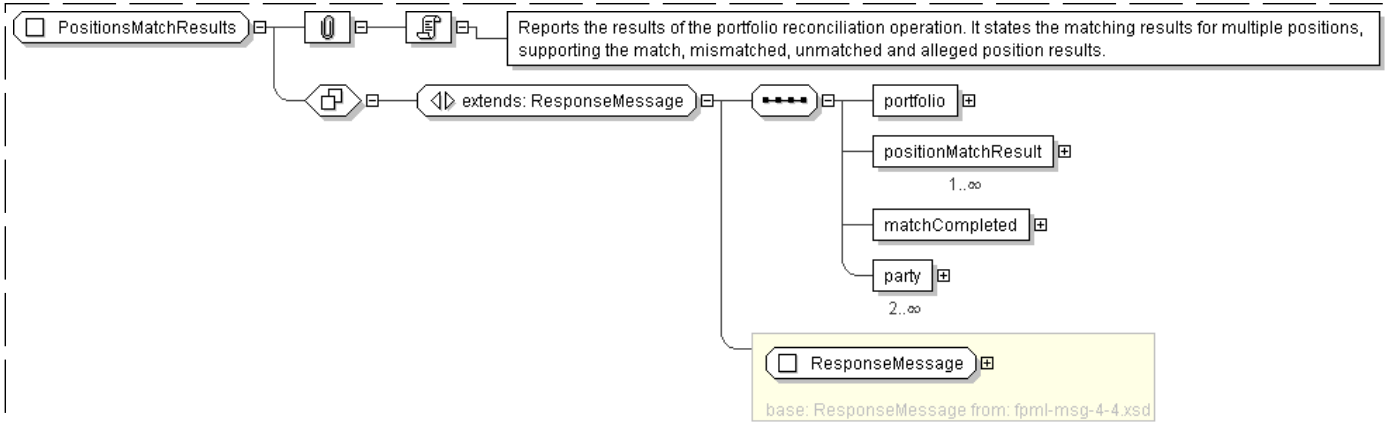
Name	PositionsMatchResults
Abstract	no
Documentation	Reports the results of the portfolio reconciliation operation. It states the matching results for multiple positions, supporting the match, mismatched, unmatched and alleged position results.

XML Instance Representation

```
<...
version=" xsd:token (value comes from list: {'4-0'|'4-1'|'4-2'|'4-3'|'4-4'}) [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="2 [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..*]
   <portfolio> PortfolioDefinition </portfolio> [1]
   'Contains the portfolio definition.'

   <positionMatchResult> PositionMatchResult </positionMatchResult> [1..*]
```

```
    <matchCompleted> xsd:boolean </matchCompleted> [1]
    'Flag indicating whether the Matching Service has finished sending all matching results.'

    <party> Party </party> [2..*]
</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="PositionsMatchResults">
   <xsd:complexContent>
      <xsd:extension base=" ResponseMessage ">
         <xsd:sequence>
            <xsd:element name="portfolio" type=" PortfolioDefinition "/>
            <xsd:element name="positionMatchResult" type=" PositionMatchResult " maxOccurs="unbounded"/>
            <xsd:element name="matchCompleted" type=" xsd:boolean "/>
            <xsd:element name="party" type=" Party " minOccurs="2" maxOccurs="unbounded"/>
         </xsd:sequence>
      </xsd:extension>
   </xsd:complexContent>
</xsd:complexType>
```

top

Complex Type: RequestPortfolio

Super-types:	RequestMessage < **RequestPortfolio** (by extension)
Sub-types:	None

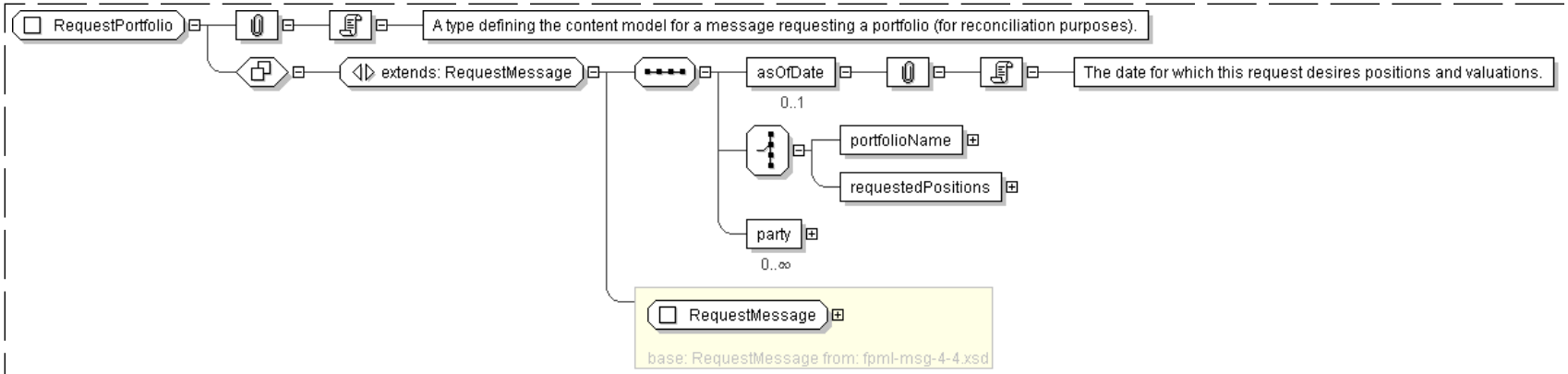
Name	RequestPortfolio
Abstract	no
Documentation	A type defining the content model for a message requesting a portfolio (for reconciliation purposes).

XML Instance Representation

```
<...
version=" xsd:token (value comes from list: {'4-0'|'4-1'|'4-2'|'4-3'|'4-4'}) [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="2 [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..*]
   <asOfDate> ... </asOfDate> [0..1]
   'The date for which this request desires positions and valuations.'

Start Choice [1]
      <portfolioName> xsd:normalizedString </portfolioName> [1]
      'The name of the portfolio that is requested.'

      <requestedPositions> RequestedPositions </requestedPositions> [1]
      'The name of the data set (portfolio, product type, etc.) that this request corresponds
      to. Describes the desired report.'

End Choice
   <party> Party </party> [0..*]
</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="RequestPortfolio">
   <xsd:complexContent>
      <xsd:extension base=" RequestMessage ">
         <xsd:sequence>
            <xsd:element name="asOfDate" minOccurs="0"/>
            <xsd:choice>
               <xsd:element name="portfolioName" type=" xsd:normalizedString "/>
               <xsd:element name="requestedPositions" type=" RequestedPositions "/>
            </xsd:choice>
            <xsd:element name="party" type=" Party " minOccurs="0" maxOccurs="unbounded"/>
         </xsd:sequence>
      </xsd:extension>
   </xsd:complexContent>
</xsd:complexType>
```

top

Complex Type: StepReference

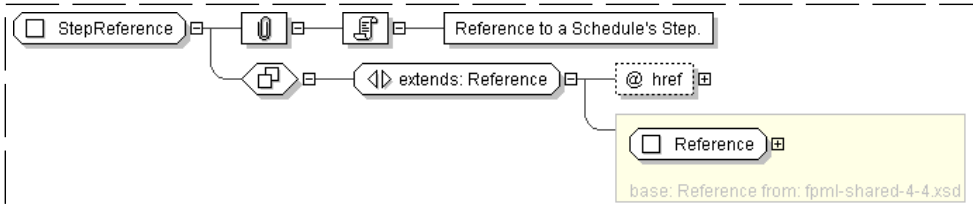
Super-types:	Reference < **StepReference** (by extension)
Sub-types:	None

Name	StepReference
Used by (from the same schema document)	Complex Type **CashflowCalculationPeriod**
Abstract	no
Documentation	Reference to a Schedule's Step.

XML Instance Representation

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

Diagram



Schema Component Representation

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

top

Complex Type: TradeCashflowsAsserted

Super-types:	NotificationMessage < **TradeCashflowsAsserted** (by extension)
Sub-types:	None

Name	TradeCashflowsAsserted
Abstract	no
Documentation	Message for assertion of payments to be reconciled. Notification message that submits cashflows that need to be reconciled per payment date at the trade level.

XML Instance Representation

```
<...
version=" xsd:token (value comes from list: {'4-0'|'4-1'|'4-2'|'4-3'|'4-4'}) [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="2 [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..*]
   <asOfDate> xsd:dateTime </asOfDate> [0..1]
```

'The date and time at which the set of cashflows was defined.'

```
   <tradeCashflowsId> TradeCashflowsId </tradeCashflowsId> [1]
```

'Unique identifier assigned by the party asserting the set of cashflows to be reconciled.'

```
   <tradeIdentifyingItems> TradeIdentifyingItems </tradeIdentifyingItems> [1]
```

'Structure that holds reference to the trade through the tradeId and optionally some trade-specific elements for identifying the trade in the case of trades that have not been negotiated through electronic platforms and for which the counterparty\'s trade ID has not been captured.'

```
   <adjustedPaymentDate> xsd:date </adjustedPaymentDate> [1]
```

'The adjusted date in which the payments are being paid/received.'

```
   <payment> PaymentMatching </payment> [1..*]
```

'Specifies the payment that is exposed to the matching process. Usually there will be a single payment but for cross-currency swaps a different payment per currency shall be provided.'

```
   <matchId> MatchId </matchId> [0..1]
```

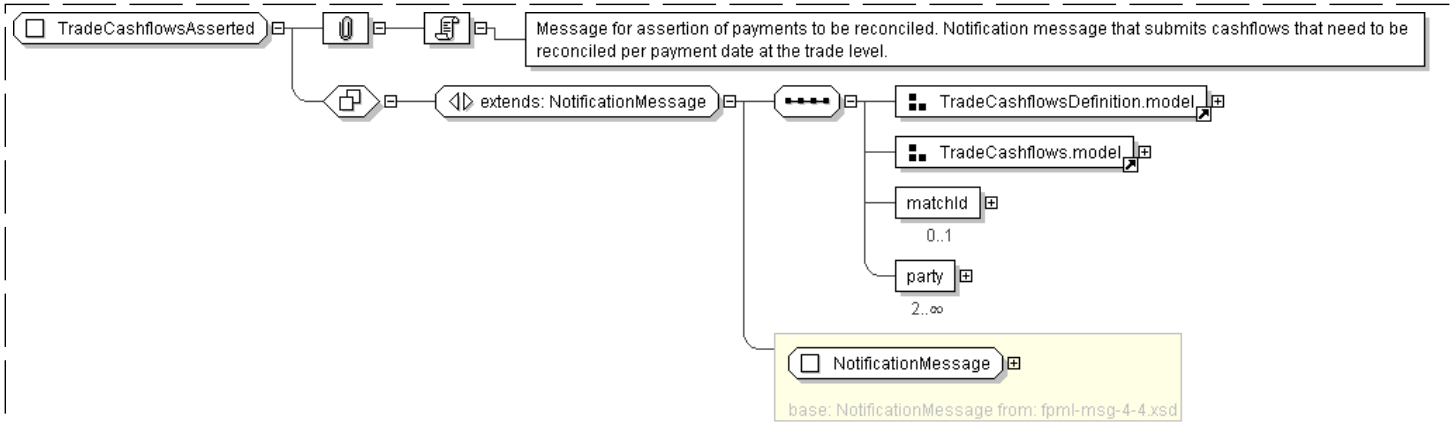
'A unique identifier assigned by either party, or matching service, as agreed, to each set of matched cashflows.'

```
   <party> Party </party> [2..*]
```

'One party element for each of the principal parties and any other party that is referenced.'

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

Diagram



Schema Component Representation

```
<xsd:complexType name="TradeCashflowsAsserted">
   <xsd:complexContent>
      <xsd:extension base=" NotificationMessage ">
```

```
        <xsd:sequence>
           <xsd:group ref=" TradeCashflowsDefinition.model "/>
           <xsd:group ref=" TradeCashflows.model "/>
           <xsd:element name="matchId" type=" MatchId " minOccurs="0"/>
           <xsd:element name="party" type=" Party " minOccurs="2" maxOccurs="unbounded"/>
        </xsd:sequence>
     </xsd:extension>
   </xsd:complexContent>
</xsd:complexType>
```

top

Complex Type: TradeCashflowsId

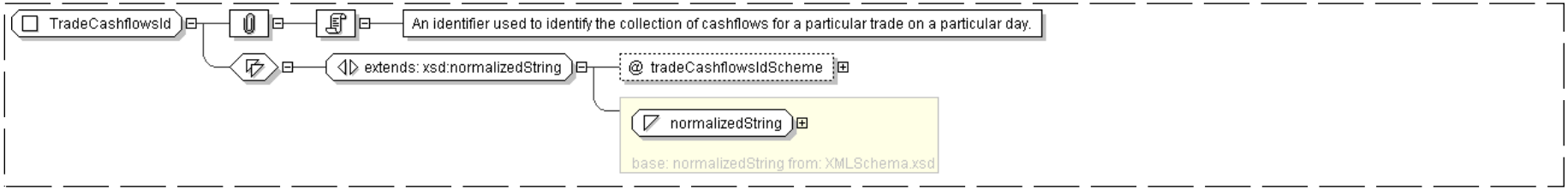
Super-types:	xsd:normalizedString < **TradeCashflowsId** (by extension)
Sub-types:	None

Name	TradeCashflowsId
Used by (from the same schema document)	Model Group **IdAndTradeCashflows.model** , Model Group **TradeCashflowsDefinition.model**
Abstract	no
Documentation	An identifier used to identify the collection of cashflows for a particular trade on a particular day.

XML Instance Representation

```
<...
tradeCashflowsIdScheme=" xsd:anyURI [0..1]">
xsd:normalizedString
</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="TradeCashflowsId">
   <xsd:simpleContent>
      <xsd:extension base=" xsd:normalizedString ">
         <xsd:attribute name="tradeCashflowsIdScheme" type=" xsd:anyURI "/>
      </xsd:extension>
   </xsd:simpleContent>
</xsd:complexType>
```

top

Complex Type: TradeCashflowsMatchResult

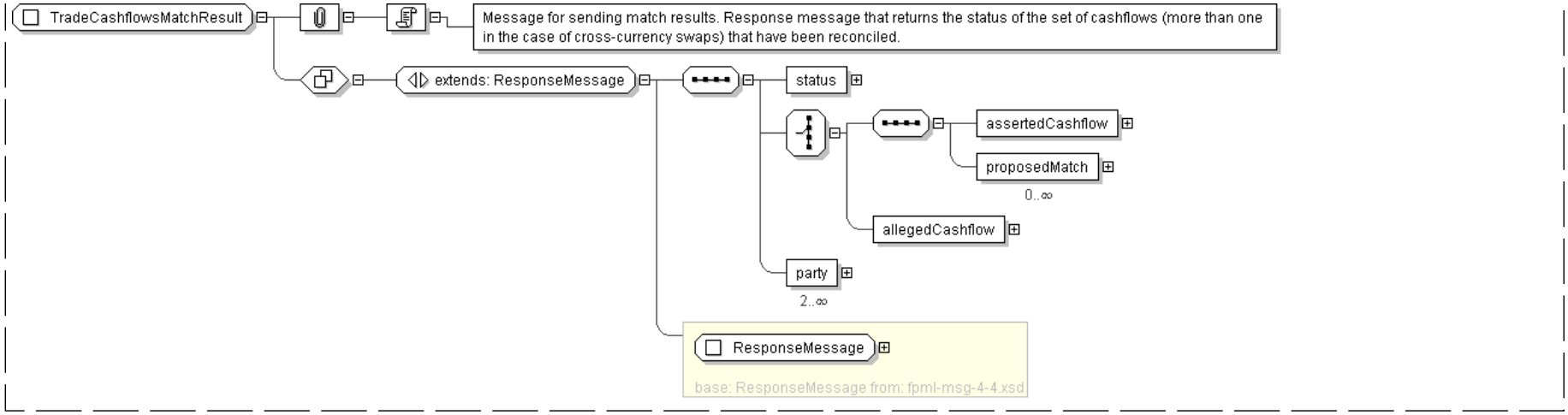
Super-types:	ResponseMessage < **TradeCashflowsMatchResult** (by extension)
Sub-types:	None

Name	TradeCashflowsMatchResult
Abstract	no
Documentation	Message for sending match results. Response message that returns the status of the set of cashflows (more than one in the case of cross-currency swaps) that have been reconciled.

XML Instance Representation

```
<...
version=" xsd:token (value comes from list: {'4-0'|'4-1'|'4-2'|'4-3'|'4-4'}) [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="2 [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..*]
   <status> TradeCashflowsStatus </status> [1]
   'Reconciliation status of the set of cashflows.'

Start Choice [1]
      <assertedCashflow> AssertedCashflow </assertedCashflow> [1]
      'Cashflow (or set of cashflows for cross-currency swap) asserted by one of the parties.'

      <proposedMatch> TradeCashflowsProposedMatch </proposedMatch> [0..*]
      '"Other side\'s" cashflow that meets the minimimum matching criteria and is proposed as
      match to the cashflow that is being asserted.'

      <allegedCashflow> AllegedCashflow </allegedCashflow> [1]
      'Cashflow (or set of cashflows for cross-currency swap) asserted by the \"other side\'s
      \" party.'

End Choice
   <party> Party </party> [2..*]
   'One party element for each of the principal parties and any other party that is referenced.'

</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="TradeCashflowsMatchResult">
   <xsd:complexContent>
      <xsd:extension base=" ResponseMessage ">
         <xsd:sequence>
            <xsd:element name="status" type=" TradeCashflowsStatus "/>
            <xsd:choice>
               <xsd:sequence>
                  <xsd:element name="assertedCashflow" type=" AssertedCashflow "/>
                  <xsd:element name="proposedMatch" type=" TradeCashflowsProposedMatch "
                  minOccurs="0" maxOccurs="unbounded"/>
               </xsd:sequence>
               <xsd:element name="allegedCashflow" type=" AllegedCashflow "/>
            </xsd:choice>
            <xsd:element name="party" type=" Party " minOccurs="2" maxOccurs="unbounded"/>
         </xsd:sequence>
      </xsd:extension>
   </xsd:complexContent>
</xsd:complexType>
```

top

Complex Type: TradeCashflowsProposedMatch

Super-types:	None
Sub-types:	None

Name	TradeCashflowsProposedMatch
Used by (from the same schema document)	Complex Type **TradeCashflowsMatchResult**
Abstract	no
Documentation	"Other side's" cashflow that meets the minimimum matching criteria and is proposed as match to the cash flow that is being asserted.

XML Instance Representation

```
<...>
   <tradeCashflowsId> TradeCashflowsId </tradeCashflowsId> [1]
   'Unique identifier assigned by either party to a set of cashflows.'

Start Group: TradeCashflows.model [0..1]
      <tradeIdentifyingItems> TradeIdentifyingItems </tradeIdentifyingItems> [1]
      'Structure that holds reference to the trade through the tradeId and optionally some
      trade-specific elements for identifying the trade in the case of trades that have not
```

been negotiated through electronic platforms and for which the counterparty\'s trade ID has not been captured.'

<adjustedPaymentDate> xsd:date </adjustedPaymentDate> [1]

'The adjusted date in which the payments are being paid/received.'

<payment> PaymentMatching </payment> [1..*]

'Specifies the payment that is exposed to the matching process. Usually there will be a single payment but for cross-currency swaps a different payment per currency shall be provided.'

End Group: TradeCashflows.model

<matchId> MatchId </matchId> [1]

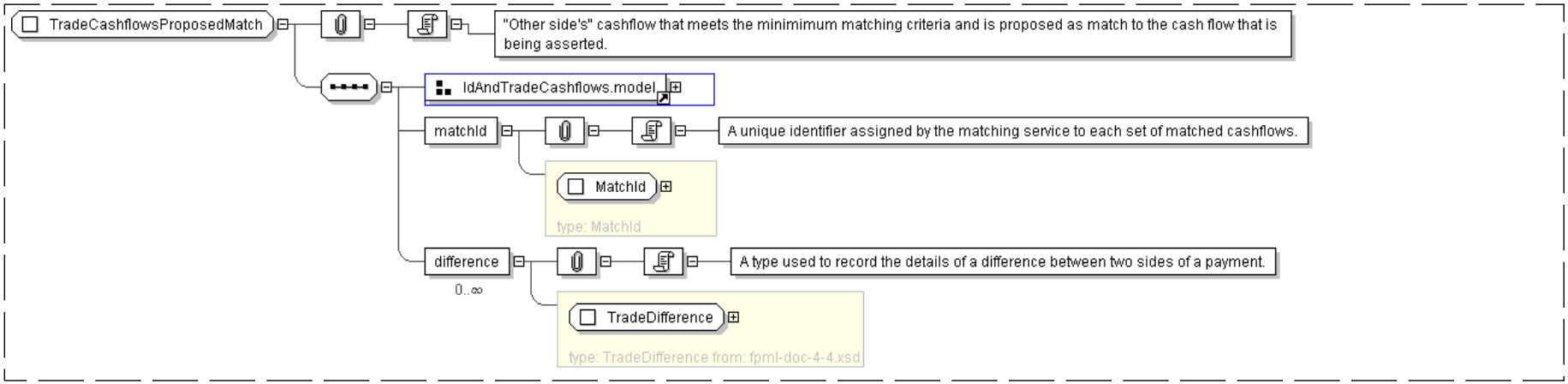
'A unique identifier assigned by the matching service to each set of matched cashflows.'

<difference> TradeDifference </difference> [0..*]

'A type used to record the details of a difference between two sides of a payment.'

</...>

Diagram



Schema Component Representation

```
<xsd:complexType name="TradeCashflowsProposedMatch">
   <xsd:sequence>
      <xsd:group ref=" IdAndTradeCashflows.model "/>
      <xsd:element name="matchId" type=" MatchId "/>
      <xsd:element name="difference" type=" TradeDifference " minOccurs="0" maxOccurs="unbounded"/>
   </xsd:sequence>
</xsd:complexType>
```

top

Complex Type: TradeCashflowsStatus

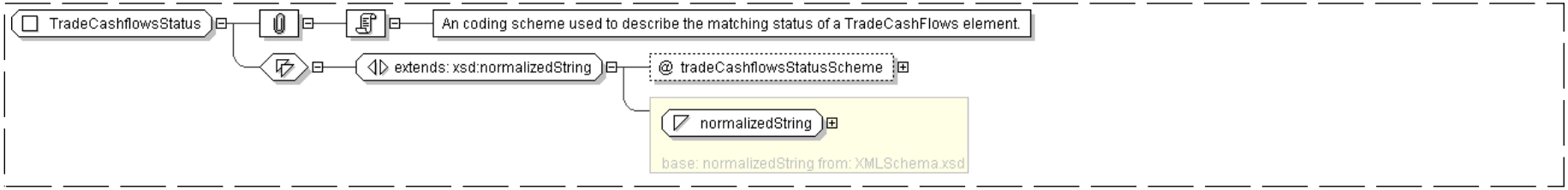
Super-types:	xsd:normalizedString < **TradeCashflowsStatus** (by extension)
Sub-types:	None

Name	TradeCashflowsStatus
Used by (from the same schema document)	Complex Type **TradeCashflowsMatchResult**
Abstract	no
Documentation	An coding scheme used to describe the matching status of a TradeCashFlows element.

XML Instance Representation

```
<...
tradeCashflowsStatusScheme=" xsd:anyURI [0..1]">
xsd:normalizedString
</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="TradeCashflowsStatus">
   <xsd:simpleContent>
      <xsd:extension base=" xsd:normalizedString ">
         <xsd:attribute name="tradeCashflowsStatusScheme" type=" xsd:anyURI " default="http://www.
         fpml.org/coding-scheme/trade-cashflows-status-1-0"/>
      </xsd:extension>
   </xsd:simpleContent>
</xsd:complexType>
```

top

Complex Type: TradeDetails

Super-types:	None
Sub-types:	None

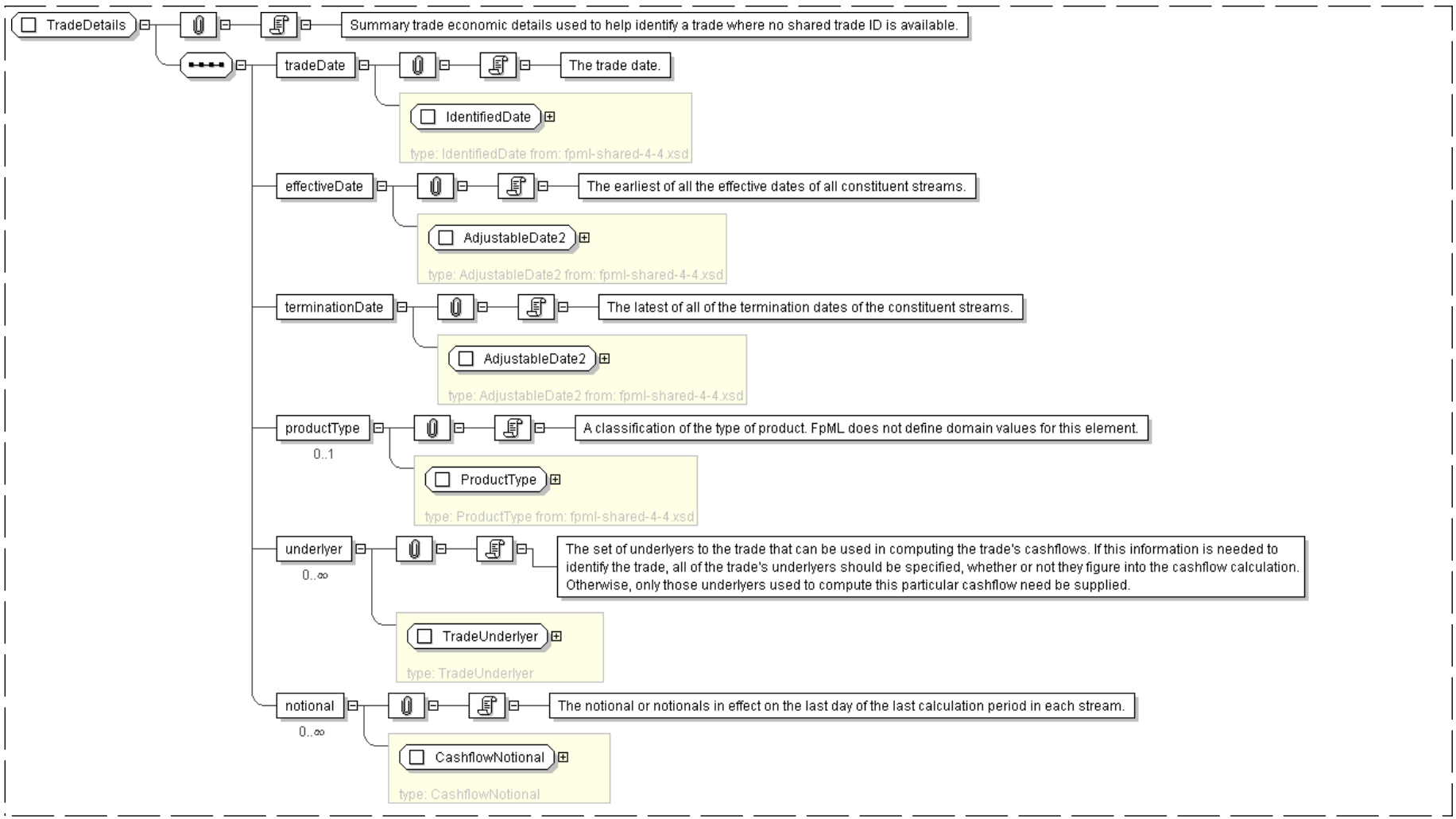
Name	TradeDetails
Used by (from the same schema document)	Complex Type **TradeIdentifyingItems**
Abstract	no
Documentation	Summary trade economic details used to help identify a trade where no shared trade ID is available.

XML Instance Representation

```
<...>
   <tradeDate> IdentifiedDate </tradeDate> [1]
   'The trade date.'

   <effectiveDate> AdjustableDate2 </effectiveDate> [1]
   'The earliest of all the effective dates of all constituent streams.'

   <terminationDate> AdjustableDate2 </terminationDate> [1]
   'The latest of all of the termination dates of the constituent streams.'

   <productType> ProductType </productType> [0..1]
   'A classification of the type of product. FpML does not define domain values for this element.'

   <underlyer> TradeUnderlyer </underlyer> [0..*]
   'The set of underlyers to the trade that can be used in computing the trade\'s cashflows.
   If this information is needed to identify the trade, all of the trade\'s underlyers should
   be specified, whether or not they figure into the cashflow calculation. Otherwise, only
   those underlyers used to compute this particular cashflow need be supplied.'

   <notional> CashflowNotional </notional> [0..*]
```

```
'The notional or notionals in effect on the last day of the last calculation period in
each stream.'

</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="TradeDetails">
   <xsd:sequence>
      <xsd:element name="tradeDate" type=" IdentifiedDate "/>
      <xsd:element name="effectiveDate" type=" AdjustableDate2 "/>
      <xsd:element name="terminationDate" type=" AdjustableDate2 "/>
      <xsd:element name="productType" type=" ProductType " minOccurs="0"/>
      <xsd:element name="underlyer" type=" TradeUnderlyer " minOccurs="0" maxOccurs="unbounded"/>
      <xsd:element name="notional" type=" CashflowNotional " minOccurs="0" maxOccurs="unbounded"/>
   </xsd:sequence>
</xsd:complexType>
```

top

Complex Type: TradeIdentifyingItems

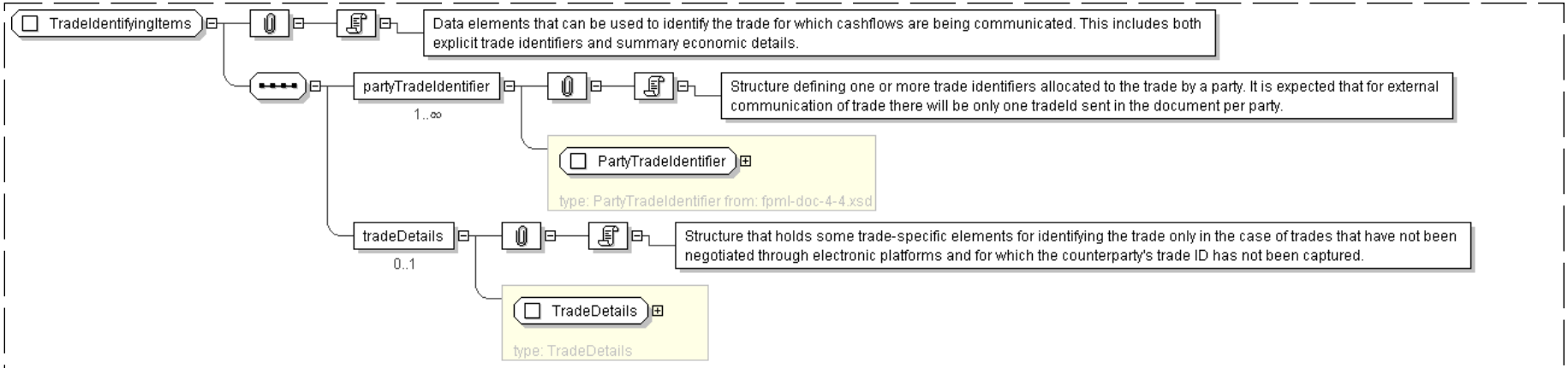
Super-types:	None
Sub-types:	None

Name	TradeIdentifyingItems
Used by (from the same schema document)	Model Group **TradeCashflows.model**
Abstract	no
Documentation	Data elements that can be used to identify the trade for which cashflows are being communicated. This includes both explicit trade identifiers and summary economic details.

XML Instance Representation

```
<...>
   <partyTradeIdentifier> PartyTradeIdentifier </partyTradeIdentifier> [1..*]
   'Structure defining one or more trade identifiers allocated to the trade by a party. It
   is expected that for external communication of trade there will be only one tradeId sent in
   the document per party.'

   <tradeDetails> TradeDetails </tradeDetails> [0..1]
   'Structure that holds some trade-specific elements for identifying the trade only in the
   case of trades that have not been negotiated through electronic platforms and for which
   the counterparty\'s trade ID has not been captured.'

</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="TradeIdentifyingItems">
   <xsd:sequence>
      <xsd:element name="partyTradeIdentifier" type=" PartyTradeIdentifier " maxOccurs="unbounded"/>
      <xsd:element name="tradeDetails" type=" TradeDetails " minOccurs="0"/>
   </xsd:sequence>
</xsd:complexType>
```

top

Complex Type: TradeUnderlyer

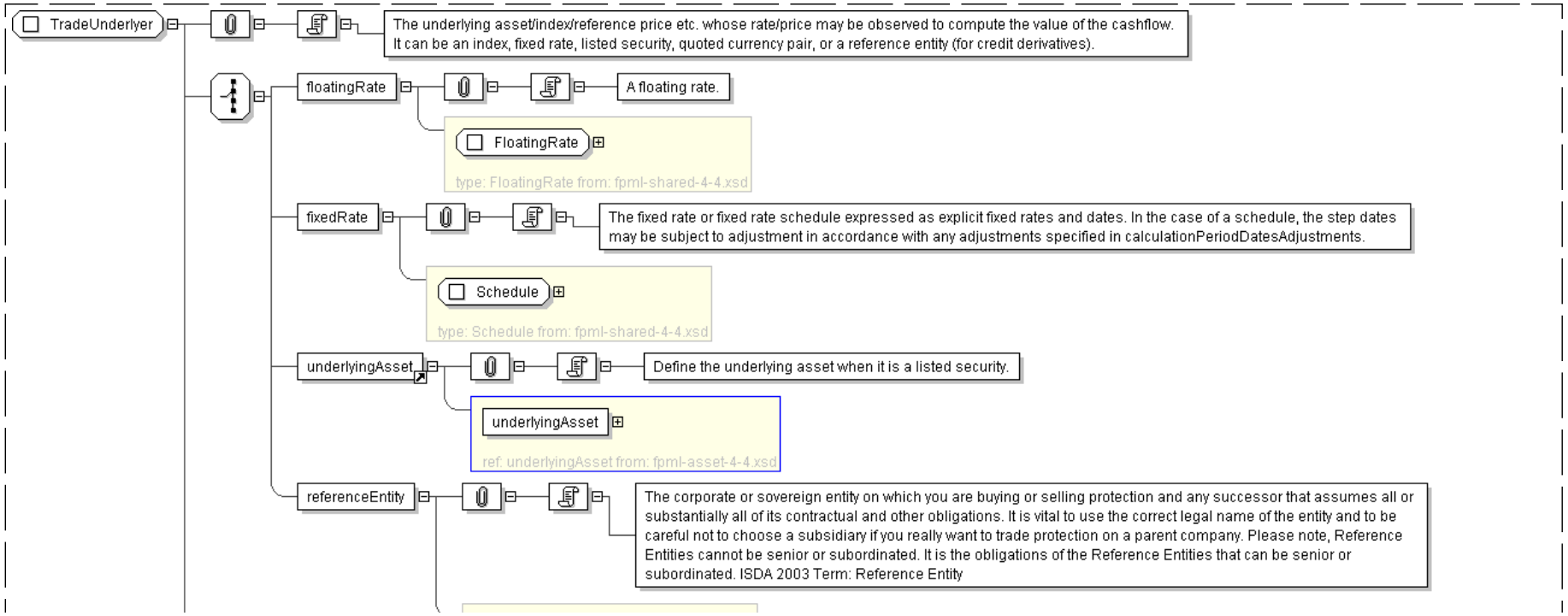
Super-types:	None
Sub-types:	None

Name	TradeUnderlyer
Used by (from the same schema document)	Complex Type **CashflowCalculationElements** , Complex Type **TradeDetails**
Abstract	no
Documentation	The underlying asset/index/reference price etc. whose rate/price may be observed to compute the value of the cashflow. It can be an index, fixed rate, listed security, quoted currency pair, or a reference entity (for credit derivatives).

XML Instance Representation

```
<...
id=" xsd:ID [0..1]">
Start Choice [1]
      <floatingRate> FloatingRate </floatingRate> [1]
      'A floating rate.'

      <fixedRate> Schedule </fixedRate> [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.'

      <underlyingAsset> ... </underlyingAsset> [1]
      'Define the underlying asset when it is a listed security.'

      <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'

End Choice
</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="TradeUnderlyer">
   <xsd:choice>
      <xsd:element name="floatingRate" type=" FloatingRate "/>
      <xsd:element name="fixedRate" type=" Schedule "/>
      <xsd:element ref=" underlyingAsset "/>
      <xsd:element name="referenceEntity" type=" LegalEntity "/>
   </xsd:choice>
   <xsd:attribute name="id" type=" xsd:ID "/>
</xsd:complexType>
```

top

Complex Type: TradeUnderlyerReference

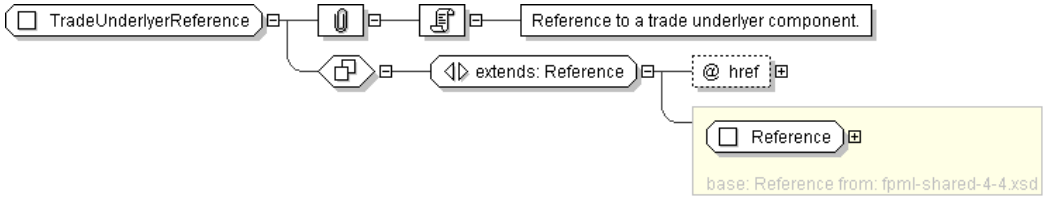
Super-types:	Reference < **TradeUnderlyerReference** (by extension)
Sub-types:	None

Name	TradeUnderlyerReference
Used by (from the same schema document)	Complex Type **CashflowObservation** , Complex Type **UnderlyerReferenceUnits**
Abstract	no
Documentation	Reference to a trade underlyer component.

XML Instance Representation

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

Diagram



Schema Component Representation

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

top

Complex Type: UnderlyerReferenceUnits

Super-types:	None

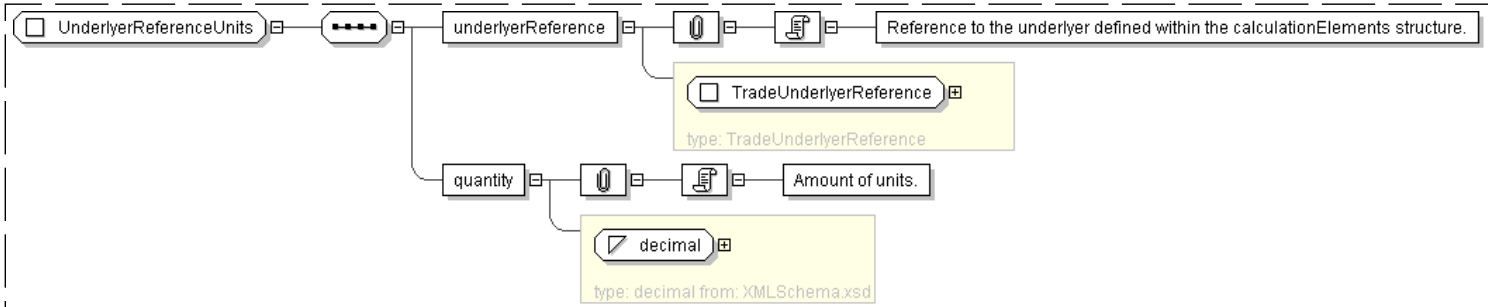
Sub-types:	None

Name	UnderlyerReferenceUnits
Used by (from the same schema document)	Complex Type **CashflowCalculationElements**
Abstract	no

XML Instance Representation

```
<...>
   <underlyerReference> TradeUnderlyerReference </underlyerReference> [1]
   'Reference to the underlyer defined within the calculationElements structure.'

   <quantity> xsd:decimal </quantity> [1]
   'Amount of units.'

</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="UnderlyerReferenceUnits">
   <xsd:sequence>
      <xsd:element name="underlyerReference" type=" TradeUnderlyerReference "/>
      <xsd:element name="quantity" type=" xsd:decimal "/>
   </xsd:sequence>
</xsd:complexType>
```

top

Complex Type: UnprocessedPosition

Super-types:	None
Sub-types:	None

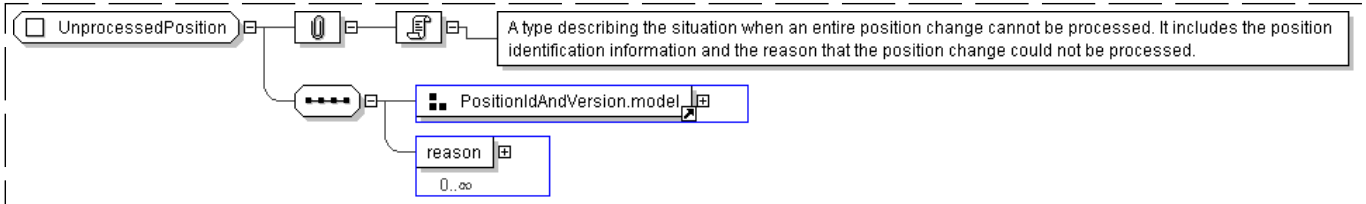
Name	UnprocessedPosition
Used by (from the same schema document)	Complex Type **PositionsAcknowledged**
Abstract	no
Documentation	A type describing the situation when an entire position change cannot be processed. It includes the position identification information and the reason that the position change could not be processed.

XML Instance Representation

```
<...>
   <positionId> PositionId </positionId> [1]
   'A version-independent identifier for the position, possibly based on trade identifier.'

   <version> xsd:positiveInteger </version> [0..1]
   'A version identifier. Version identifiers must be ascending, i.e. higher numbers imply
```

```
    newer versions. There is no requirement that version identifiers for a position be
    sequential or small, so for example timestamp-based version identifiers could be used.'

    <reason> Reason </reason> [0..*]
</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="UnprocessedPosition">
   <xsd:sequence>
      <xsd:group ref=" PositionIdAndVersion.model "/>
      <xsd:element name="reason" type=" Reason " minOccurs="0" maxOccurs="unbounded"/>
   </xsd:sequence>
</xsd:complexType>
```

top

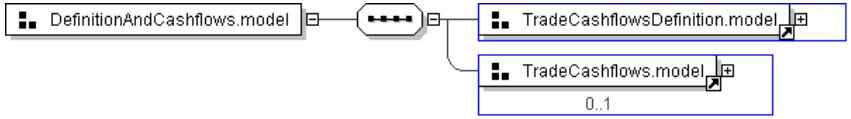
Model Group: DefinitionAndCashflows.model

Name	DefinitionAndCashflows.model
Used by (from the same schema document)	Complex Type **AllegedCashflow** , Complex Type **AssertedCashflow**

XML Instance Representation

```
<asOfDate> xsd:dateTime </asOfDate> [0..1]
'The date and time at which the set of cashflows was defined.'

<tradeCashflowsId> TradeCashflowsId </tradeCashflowsId> [1]
'Unique identifier assigned by the party asserting the set of cashflows to be reconciled.'

Start Group: TradeCashflows.model [0..1]
   <tradeIdentifyingItems> TradeIdentifyingItems </tradeIdentifyingItems> [1]
   'Structure that holds reference to the trade through the tradeId and optionally some
   trade-specific elements for identifying the trade in the case of trades that have not
   been negotiated through electronic platforms and for which the counterparty\'s trade ID has
   not been captured.'

   <adjustedPaymentDate> xsd:date </adjustedPaymentDate> [1]
   'The adjusted date in which the payments are being paid/received.'

   <payment> PaymentMatching </payment> [1..*]
   'Specifies the payment that is exposed to the matching process. Usually there will be a
   single payment but for cross-currency swaps a different payment per currency shall be provided.'

End Group: TradeCashflows.model
```

Diagram



Schema Component Representation

```
<xsd:group name="DefinitionAndCashflows.model">
   <xsd:sequence>
      <xsd:group ref=" TradeCashflowsDefinition.model "/>
      <xsd:group ref=" TradeCashflows.model " minOccurs="0"/>
   </xsd:sequence>
</xsd:group>
```

top

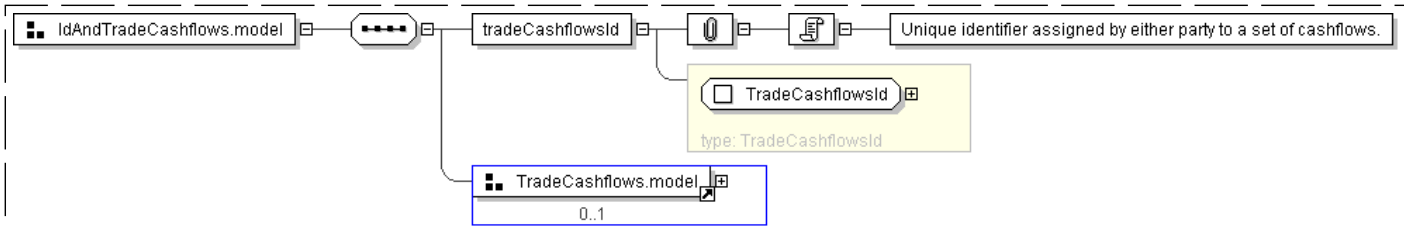
Model Group: IdAndTradeCashflows.model

Name	IdAndTradeCashflows.model
Used by (from the same schema document)	Complex Type **CancelTradeCashflows** , Complex Type **TradeCashflowsProposedMatch**

XML Instance Representation

```
<tradeCashflowsId> TradeCashflowsId </tradeCashflowsId> [1]
'Unique identifier assigned by either party to a set of cashflows.'

Start Group: TradeCashflows.model [0..1]
   <tradeIdentifyingItems> TradeIdentifyingItems </tradeIdentifyingItems> [1]
   'Structure that holds reference to the trade through the tradeId and optionally some
   trade-specific elements for identifying the trade in the case of trades that have not
   been negotiated through electronic platforms and for which the counterparty\'s trade ID has
   not been captured.'

   <adjustedPaymentDate> xsd:date </adjustedPaymentDate> [1]
   'The adjusted date in which the payments are being paid/received.'

   <payment> PaymentMatching </payment> [1..*]
   'Specifies the payment that is exposed to the matching process. Usually there will be a
   single payment but for cross-currency swaps a different payment per currency shall be provided.'

End Group: TradeCashflows.model
```

Diagram



Schema Component Representation

```
<xsd:group name="IdAndTradeCashflows.model">
   <xsd:sequence>
      <xsd:element name="tradeCashflowsId" type=" TradeCashflowsId "/>
      <xsd:group ref=" TradeCashflows.model " minOccurs="0"/>
   </xsd:sequence>
</xsd:group>
```

top

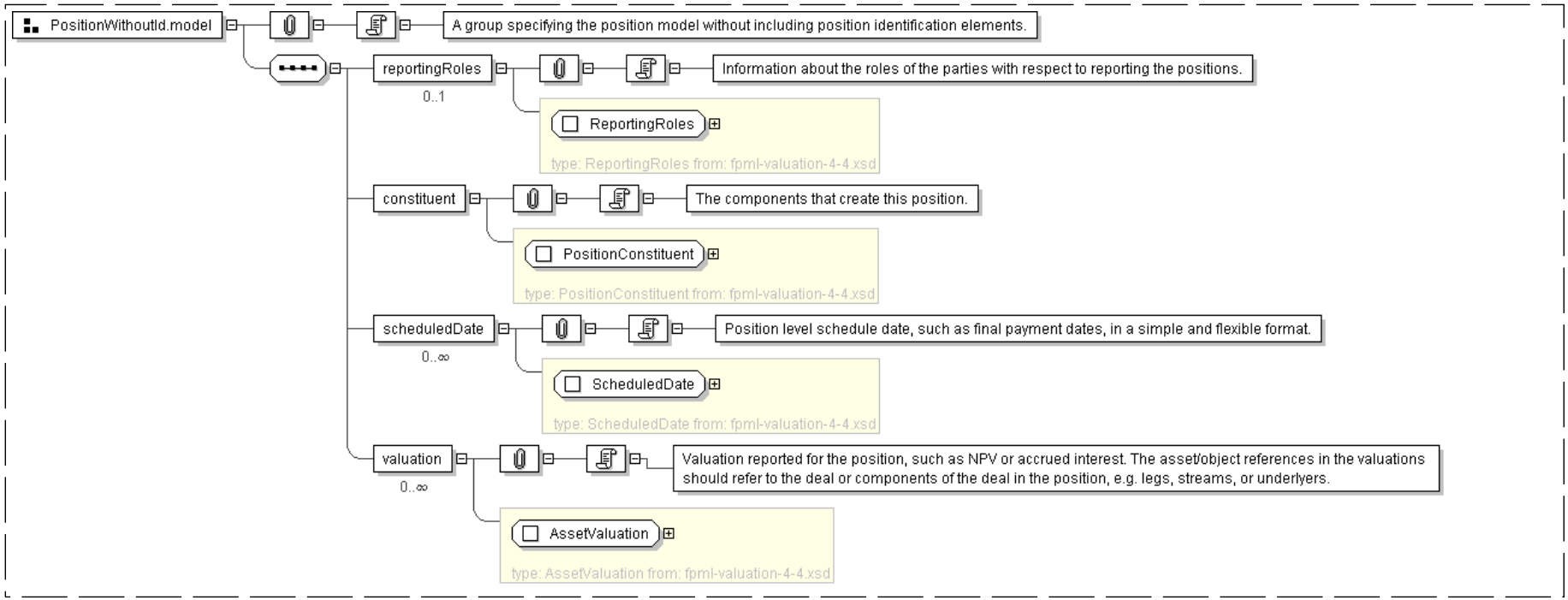
Model Group: PositionWithoutId.model

Name	PositionWithoutId.model
Used by (from the same schema document)	Complex Type **AssertedPosition** , Complex Type **PositionProposedMatch**
Documentation	A group specifying the position model without including position identification elements.

XML Instance Representation

```
<reportingRoles> ReportingRoles </reportingRoles> [0..1]
'Information about the roles of the parties with respect to reporting the positions.'

<constituent> PositionConstituent </constituent> [1]
'The components that create this position.'

<scheduledDate> ScheduledDate </scheduledDate> [0..*]
'Position level schedule date, such as final payment dates, in a simple and flexible format.'

<valuation> AssetValuation </valuation> [0..*]
'Valuation reported for the position, such as NPV or accrued interest. The asset/
object references in the valuations should refer to the deal or components of the deal in
the position, e.g. legs, streams, or underlyers.'
```

Diagram



Schema Component Representation

```
<xsd:group name="PositionWithoutId.model">
   <xsd:sequence>
      <xsd:element name="reportingRoles" type=" ReportingRoles " minOccurs="0"/>
      <xsd:element name="constituent" type=" PositionConstituent "/>
      <xsd:element name="scheduledDate" type=" ScheduledDate " minOccurs="0" maxOccurs="unbounded"/>
```

```
      <xsd:element name="valuation" type=" AssetValuation " minOccurs="0" maxOccurs="unbounded"/>
   </xsd:sequence>
</xsd:group>
```

top

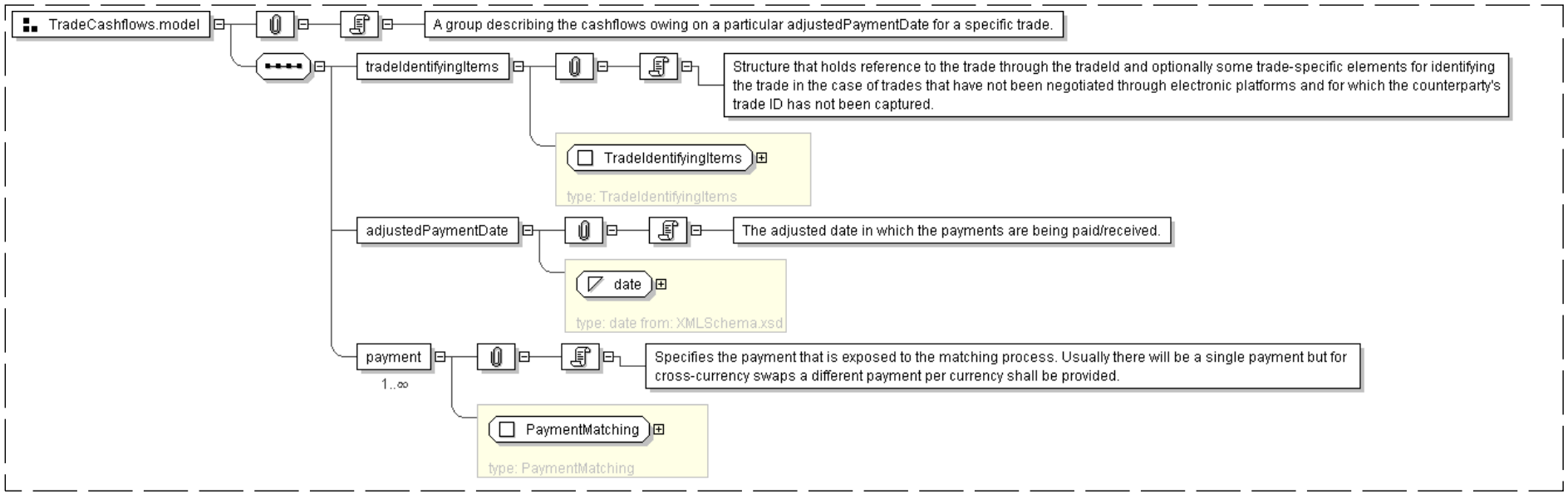
Model Group: TradeCashflows.model

Name	TradeCashflows.model
Used by (from the same schema document)	Complex Type **TradeCashflowsAsserted** , Model Group **DefinitionAndCashflows.model** , Model Group **IdAndTradeCashflows.model**
Documentation	A group describing the cashflows owing on a particular adjustedPaymentDate for a specific trade.

XML Instance Representation

```
<tradeIdentifyingItems> TradeIdentifyingItems </tradeIdentifyingItems> [1]
'Structure that holds reference to the trade through the tradeId and optionally some
trade-specific elements for identifying the trade in the case of trades that have not
been negotiated through electronic platforms and for which the counterparty\'s trade ID has
not been captured.'

<adjustedPaymentDate> xsd:date </adjustedPaymentDate> [1]
'The adjusted date in which the payments are being paid/received.'

<payment> PaymentMatching </payment> [1..*]
'Specifies the payment that is exposed to the matching process. Usually there will be a
single payment but for cross-currency swaps a different payment per currency shall be provided.'
```

Diagram



Schema Component Representation

```
<xsd:group name="TradeCashflows.model">
   <xsd:sequence>
      <xsd:element name="tradeIdentifyingItems" type=" TradeIdentifyingItems "/>
      <xsd:element name="adjustedPaymentDate" type=" xsd:date "/>
      <xsd:element name="payment" type=" PaymentMatching " maxOccurs="unbounded"/>
   </xsd:sequence>
```

```
</xsd:group>
```

top

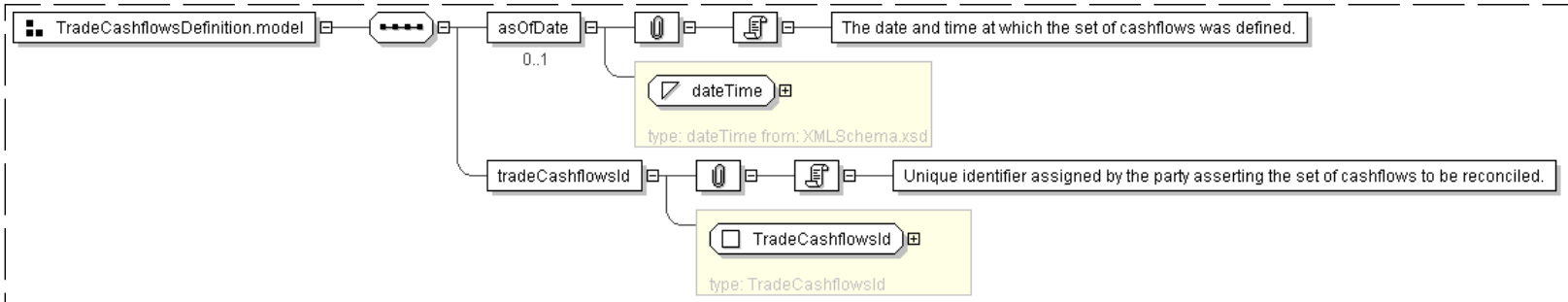
Model Group: TradeCashflowsDefinition.model

Name	TradeCashflowsDefinition.model
Used by (from the same schema document)	Complex Type **TradeCashflowsAsserted** , Model Group **DefinitionAndCashflows.model**

XML Instance Representation

```
<asOfDate> xsd:dateTime </asOfDate> [0..1]
'The date and time at which the set of cashflows was defined.'

<tradeCashflowsId> TradeCashflowsId </tradeCashflowsId> [1]
'Unique identifier assigned by the party asserting the set of cashflows to be reconciled.'
```

Diagram



Schema Component Representation

```
<xsd:group name="TradeCashflowsDefinition.model">
   <xsd:sequence>
      <xsd:element name="asOfDate" type=" xsd:dateTime " minOccurs="0"/>
      <xsd:element name="tradeCashflowsId" type=" TradeCashflowsId "/>
   </xsd:sequence>
</xsd:group>
```

top

Legend

Complex Type: **AusAddress**

Schema Component Type — 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 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.

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.

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.

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.

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.

top

Generated by <oXygen/> XML Editor using a modified version of xs3p that adds schema diagrams and chunking support.