XML Schema Documentation

Table of Contents

- Schema Document Properties
- Global Definitions
  - Simple Type: **AveragingInOutEnum**
  - Simple Type: **AveragingMethodEnum**
  - Simple Type: **BusinessDayConventionEnum**
  - Simple Type: **CalculationAgentPartyEnum**
  - Simple Type: **CommissionDenominationEnum**
  - Simple Type: **CompoundingMethodEnum**
  - Simple Type: **DayTypeEnum**
  - Simple Type: **DifferenceSeverityEnum**
  - Simple Type: **DifferenceTypeEnum**
  - Simple Type: **DiscountingTypeEnum**
  - Simple Type: **DividendAmountTypeEnum**
  - Simple Type: **DividendDateReferenceEnum**
  - Simple Type: **DividendEntitlementEnum**
  - Simple Type: **DividendPeriodEnum**
  - Simple Type: **ExerciseStyleEnum**
  - Simple Type: **FraDiscountingEnum**
  - Simple Type: **FrequencyTypeEnum**
  - Simple Type: **FxBarrierTypeEnum**
  - Simple Type: **IndexEventConsequenceEnum**
  - Simple Type: **InterestShortfallCapEnum**
  - Simple Type: **LengthUnitEnum**
  - Simple Type: **MethodOfAdjustmentEnum**
  - Simple Type: **NationalisationOrInsolvencyOrDelistingEventEnum**
  - Simple Type: **NegativeInterestRateTreatmentEnum**
  - Simple Type: **NotionalAdjustmentEnum**
  - Simple Type: **ObligationCategoryEnum**
  - Simple Type: **OptionTypeEnum**
  - Simple Type: **PayRelativeToEnum**
  - Simple Type: **PayerReceiverEnum**
  - Simple Type: **PayoutEnum**
  - Simple Type: **PeriodEnum**
  - Simple Type: **PremiumQuoteBasisEnum**
  - Simple Type: **PremiumTypeEnum**
  - Simple Type: **PriceExpressionEnum**
  - Simple Type: **QuotationRateTypeEnum**
  - Simple Type: **QuotationSideEnum**
  - Simple Type: **QuoteBasisEnum**
  - Simple Type: **RateTreatmentEnum**
  - Simple Type: **RealisedVarianceMethodEnum**
  - Simple Type: **ResetRelativeToEnum**
  - Simple Type: **ReturnTypeEnum**
  - Simple Type: **RollConventionEnum**
  - Simple Type: **RoundingDirectionEnum**
  - Simple Type: **SettlementTypeEnum**
  - Simple Type: **ShareExtraordinaryEventEnum**
  - Simple Type: **SideRateBasisEnum**
  - Simple Type: **StandardSettlementStyleEnum**
  - Simple Type: **StepRelativeToEnum**
  - Simple Type: **StrikeQuoteBasisEnum**
  - Simple Type: **StubPeriodTypeEnum**
  - Simple Type: **TimeTypeEnum**
  - Simple Type: **TouchConditionEnum**
  - Simple Type: **TriggerConditionEnum**
  - Simple Type: **ValuationMethodEnum**
  - Simple Type: **WeeklyRollConventionEnum**
- Legend
- Glossary

top

Schema Document Properties

Target Namespace	http://www.fpml.org/2007/FpML-4-4
Version	$Revision: 2586 $
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.

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
fpml-annotation	http://www.fpml.org/annotation
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: 2586 $"
attributeFormDefault="unqualified" elementFormDefault="qualified">
   ...
</xsd:schema>
```

top

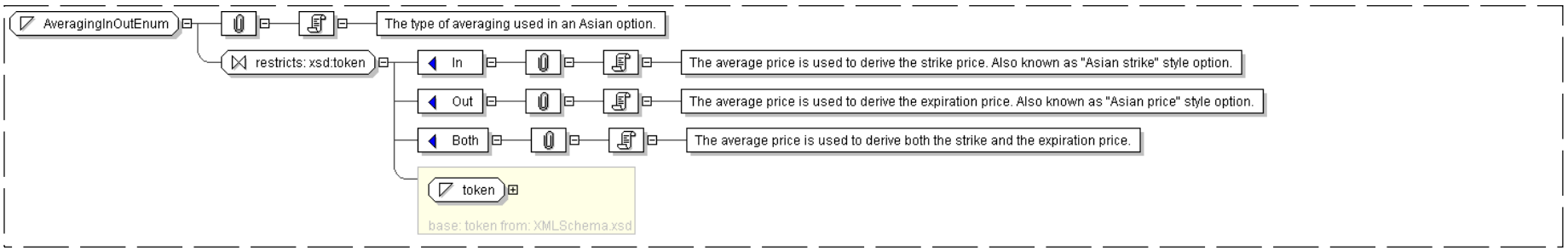
Global Definitions

Simple Type: AveragingInOutEnum

Super-types:	xsd:token < **AveragingInOutEnum** (by restriction)
Sub-types:	None

Name	AveragingInOutEnum
Content	• Base XSD Type: token • *value* comes from list: {'In'|'Out'|'Both'}
Documentation	The type of averaging used in an Asian option.

Diagram



Schema Component Representation

```
<xsd:simpleType name="AveragingInOutEnum">
   <xsd:restriction base=" xsd:token ">
      <xsd:enumeration value="In"/>
      <xsd:enumeration value="Out"/>
      <xsd:enumeration value="Both"/>
   </xsd:restriction>
</xsd:simpleType>
```

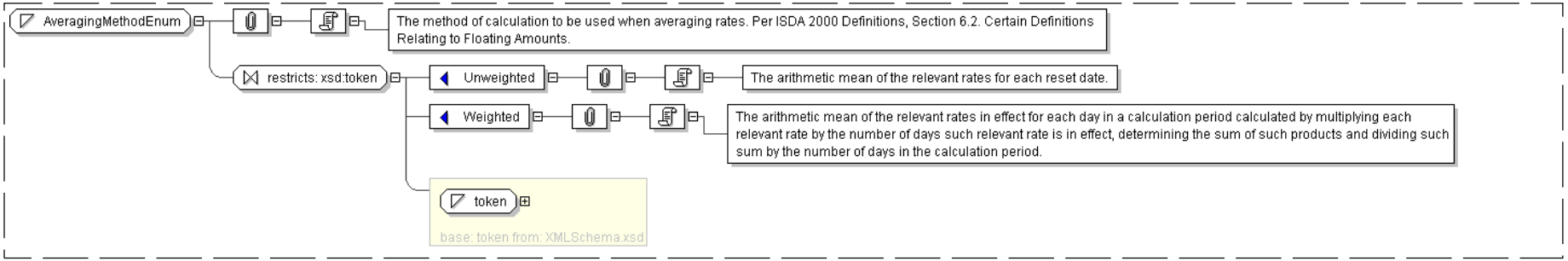
top

Simple Type: AveragingMethodEnum

Super-types:	xsd:token < **AveragingMethodEnum** (by restriction)
Sub-types:	None

Name	AveragingMethodEnum
Content	• Base XSD Type: token • *value* comes from list: {'Unweighted'|'Weighted'}
Documentation	The method of calculation to be used when averaging rates. Per ISDA 2000 Definitions, Section 6.2. Certain Definitions Relating to Floating Amounts.

Diagram



Schema Component Representation

```
<xsd:simpleType name="AveragingMethodEnum">
   <xsd:restriction base=" xsd:token ">
      <xsd:enumeration value="Unweighted"/>
      <xsd:enumeration value="Weighted"/>
   </xsd:restriction>
</xsd:simpleType>
```

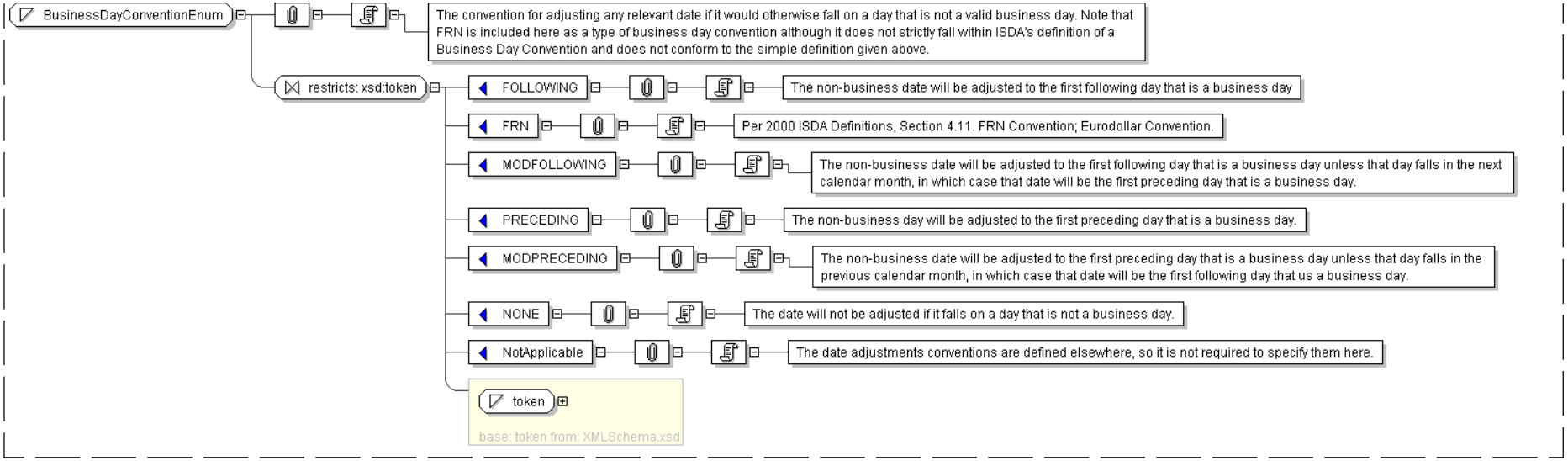
top

Simple Type: BusinessDayConventionEnum

Super-types:	xsd:token < **BusinessDayConventionEnum** (by restriction)
Sub-types:	None

Name	BusinessDayConventionEnum
Content	• Base XSD Type: token • *value* comes from list: {'FOLLOWING'|'FRN'|'MODFOLLOWING'|'PRECEDING'|'MODPRECEDING'|'NONE'|'NotApplicable'}
Documentation	The convention for adjusting any relevant date if it would otherwise fall on a day that is not a valid business day. Note that FRN is included here as a type of business day convention although it does not strictly fall within ISDA's definition of a Business Day Convention and does not conform to the simple definition given above.

Diagram



Schema Component Representation

```
<xsd:simpleType name="BusinessDayConventionEnum">
   <xsd:restriction base=" xsd:token ">
      <xsd:enumeration value="FOLLOWING"/>
      <xsd:enumeration value="FRN"/>
      <xsd:enumeration value="MODFOLLOWING"/>
      <xsd:enumeration value="PRECEDING"/>
      <xsd:enumeration value="MODPRECEDING"/>
      <xsd:enumeration value="NONE"/>
      <xsd:enumeration value="NotApplicable"/>
   </xsd:restriction>
</xsd:simpleType>
```

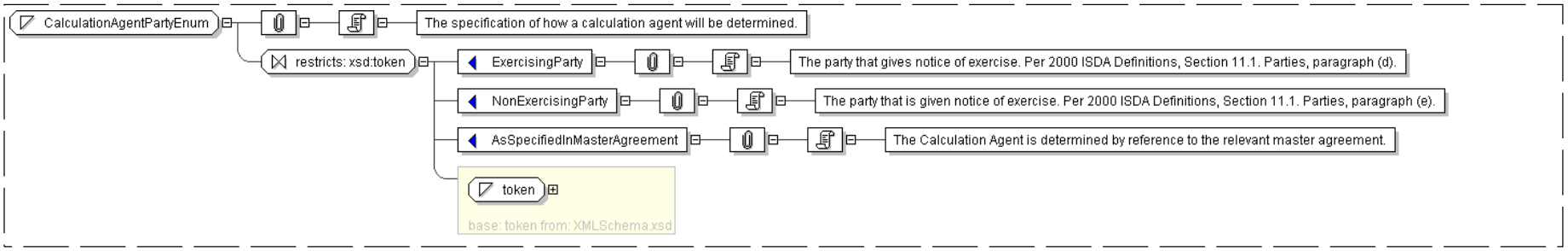
top

Simple Type: CalculationAgentPartyEnum

Super-types:	xsd:token < **CalculationAgentPartyEnum** (by restriction)
Sub-types:	None

Name	CalculationAgentPartyEnum
Content	• Base XSD Type: token • *value* comes from list: {'ExercisingParty'|'NonExercisingParty'|'AsSpecifiedInMasterAgreement'}
Documentation	The specification of how a calculation agent will be determined.

Diagram



Schema Component Representation

```
<xsd:simpleType name="CalculationAgentPartyEnum">
```

```
   <xsd:restriction base=" xsd:token ">
      <xsd:enumeration value="ExercisingParty"/>
      <xsd:enumeration value="NonExercisingParty"/>
      <xsd:enumeration value="AsSpecifiedInMasterAgreement"/>
   </xsd:restriction>
</xsd:simpleType>
```

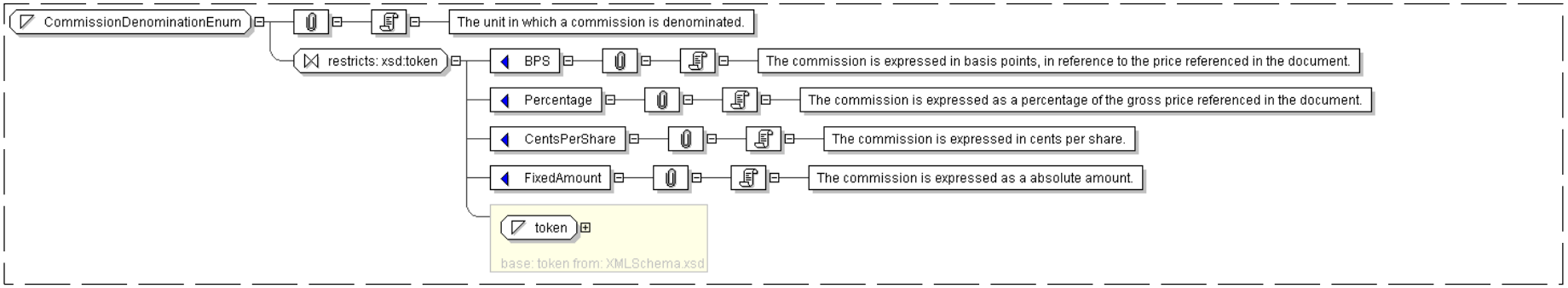
top

Simple Type: CommissionDenominationEnum

Super-types:	xsd:token < **CommissionDenominationEnum** (by restriction)
Sub-types:	None

Name	CommissionDenominationEnum
Content	• Base XSD Type: token • *value* comes from list: {'BPS'|'Percentage'|'CentsPerShare'|'FixedAmount'}
Documentation	The unit in which a commission is denominated.

Diagram



Schema Component Representation

```
<xsd:simpleType name="CommissionDenominationEnum">
   <xsd:restriction base=" xsd:token ">
      <xsd:enumeration value="BPS"/>
      <xsd:enumeration value="Percentage"/>
      <xsd:enumeration value="CentsPerShare"/>
      <xsd:enumeration value="FixedAmount"/>
   </xsd:restriction>
</xsd:simpleType>
```

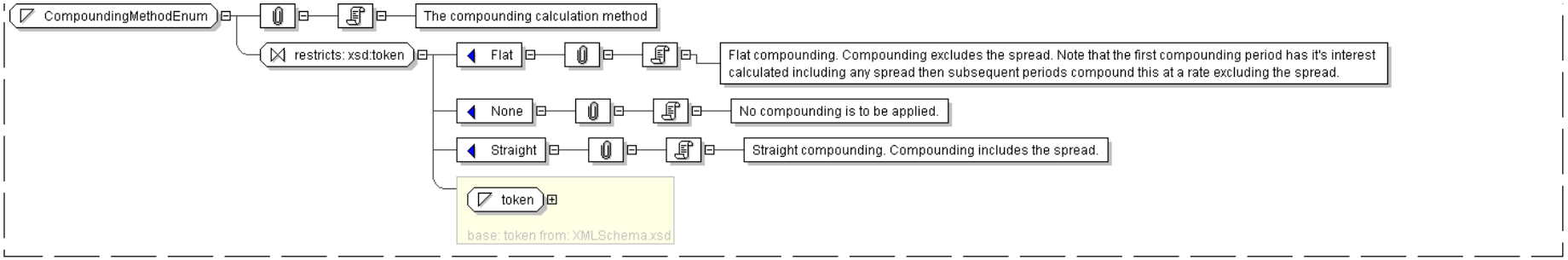
top

Simple Type: CompoundingMethodEnum

Super-types:	xsd:token < **CompoundingMethodEnum** (by restriction)
Sub-types:	None

Name	CompoundingMethodEnum
Content	• Base XSD Type: token • *value* comes from list: {'Flat'|'None'|'Straight'}
Documentation	The compounding calculation method

Diagram



Schema Component Representation

```
<xsd:simpleType name="CompoundingMethodEnum">
   <xsd:restriction base=" xsd:token ">
      <xsd:enumeration value="Flat"/>
      <xsd:enumeration value="None"/>
      <xsd:enumeration value="Straight"/>
   </xsd:restriction>
</xsd:simpleType>
```

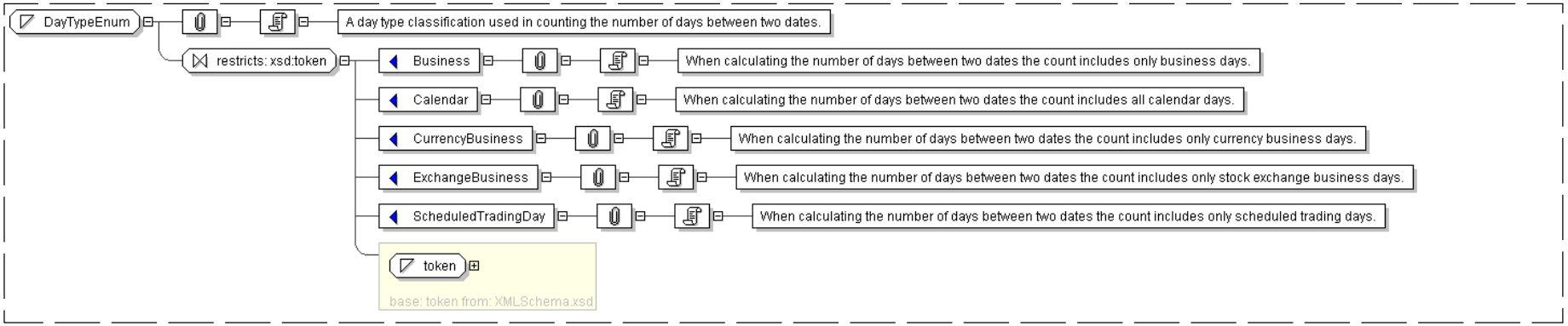
top

Simple Type: DayTypeEnum

Super-types:	xsd:token < **DayTypeEnum** (by restriction)
Sub-types:	None

Name	DayTypeEnum
Content	• Base XSD Type: token • *value* comes from list: {'Business'|'Calendar'|'CurrencyBusiness'|'ExchangeBusiness'|'ScheduledTradingDay'}
Documentation	A day type classification used in counting the number of days between two dates.

Diagram



Schema Component Representation

```
<xsd:simpleType name="DayTypeEnum">
   <xsd:restriction base=" xsd:token ">
      <xsd:enumeration value="Business"/>
      <xsd:enumeration value="Calendar"/>
      <xsd:enumeration value="CurrencyBusiness"/>
      <xsd:enumeration value="ExchangeBusiness"/>
      <xsd:enumeration value="ScheduledTradingDay"/>
   </xsd:restriction>
</xsd:simpleType>
```

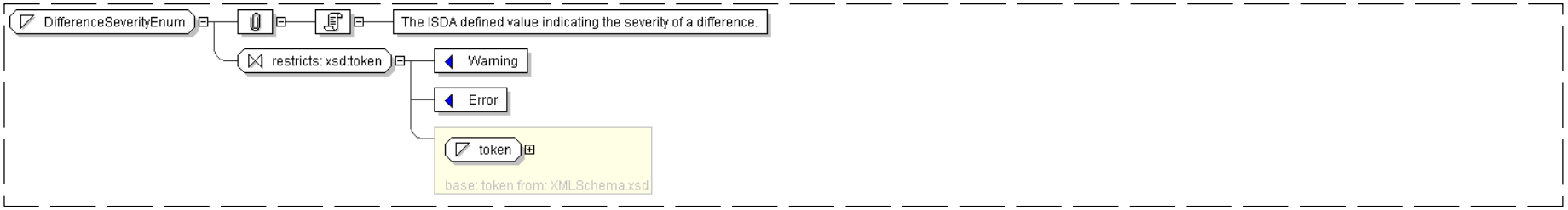
top

Simple Type: DifferenceSeverityEnum

Super-types:	xsd:token < **DifferenceSeverityEnum** (by restriction)
Sub-types:	None

Name	DifferenceSeverityEnum
Content	• Base XSD Type: token • *value* comes from list: {'Warning'|'Error'}
Documentation	The ISDA defined value indicating the severity of a difference.

Diagram



Schema Component Representation

```
<xsd:simpleType name="DifferenceSeverityEnum">
   <xsd:restriction base=" xsd:token ">
      <xsd:enumeration value="Warning"/>
      <xsd:enumeration value="Error"/>
   </xsd:restriction>
</xsd:simpleType>
```

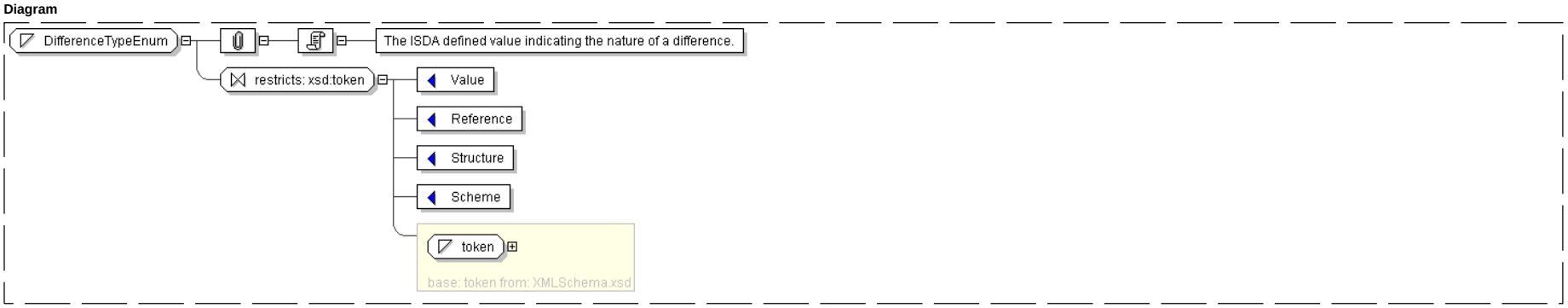
top

Simple Type: DifferenceTypeEnum

Super-types:	xsd:token < **DifferenceTypeEnum** (by restriction)
Sub-types:	None

Name	DifferenceTypeEnum
Content	• Base XSD Type: token • *value* comes from list: {'Value'|'Reference'|'Structure'|'Scheme'}
Documentation	The ISDA defined value indicating the nature of a difference.

Diagram



Schema Component Representation

```
<xsd:simpleType name="DifferenceTypeEnum">
   <xsd:restriction base=" xsd:token ">
      <xsd:enumeration value="Value"/>
```

```
      <xsd:enumeration value="Reference"/>
      <xsd:enumeration value="Structure"/>
      <xsd:enumeration value="Scheme"/>
   </xsd:restriction>
</xsd:simpleType>
```

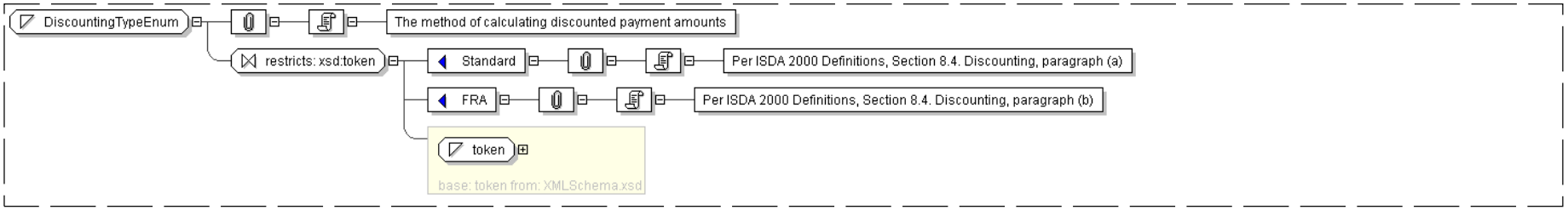
top

Simple Type: DiscountingTypeEnum

Super-types:	xsd:token < **DiscountingTypeEnum** (by restriction)
Sub-types:	None

Name	DiscountingTypeEnum
Content	• Base XSD Type: token • *value* comes from list: {'Standard'|'FRA'}
Documentation	The method of calculating discounted payment amounts

Diagram



Schema Component Representation

```
<xsd:simpleType name="DiscountingTypeEnum">
   <xsd:restriction base=" xsd:token ">
      <xsd:enumeration value="Standard"/>
      <xsd:enumeration value="FRA"/>
   </xsd:restriction>
</xsd:simpleType>
```

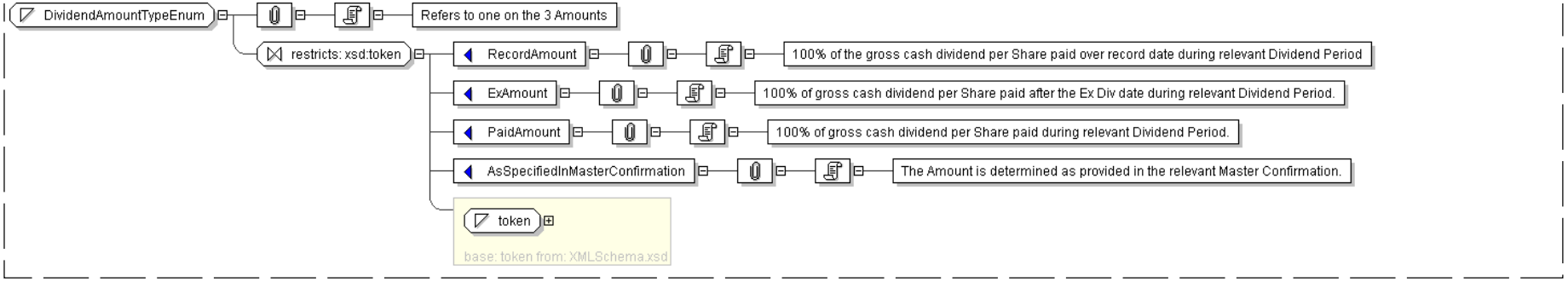
top

Simple Type: DividendAmountTypeEnum

Super-types:	xsd:token < **DividendAmountTypeEnum** (by restriction)
Sub-types:	None

Name	DividendAmountTypeEnum
Content	• Base XSD Type: token • *value* comes from list: {'RecordAmount'|'ExAmount'|'PaidAmount'|'AsSpecifiedInMasterConfirmation'}
Documentation	Refers to one on the 3 Amounts

Diagram



Schema Component Representation

```
<xsd:simpleType name="DividendAmountTypeEnum">
   <xsd:restriction base=" xsd:token ">
      <xsd:enumeration value="RecordAmount"/>
      <xsd:enumeration value="ExAmount"/>
      <xsd:enumeration value="PaidAmount"/>
      <xsd:enumeration value="AsSpecifiedInMasterConfirmation"/>
   </xsd:restriction>
</xsd:simpleType>
```

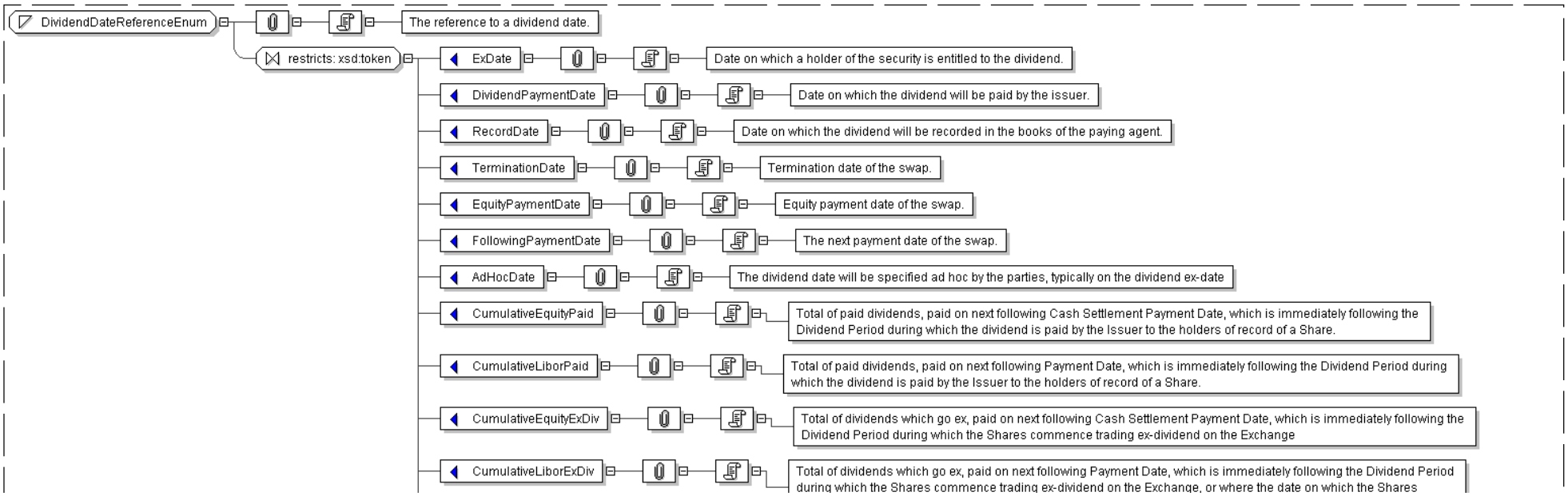
top

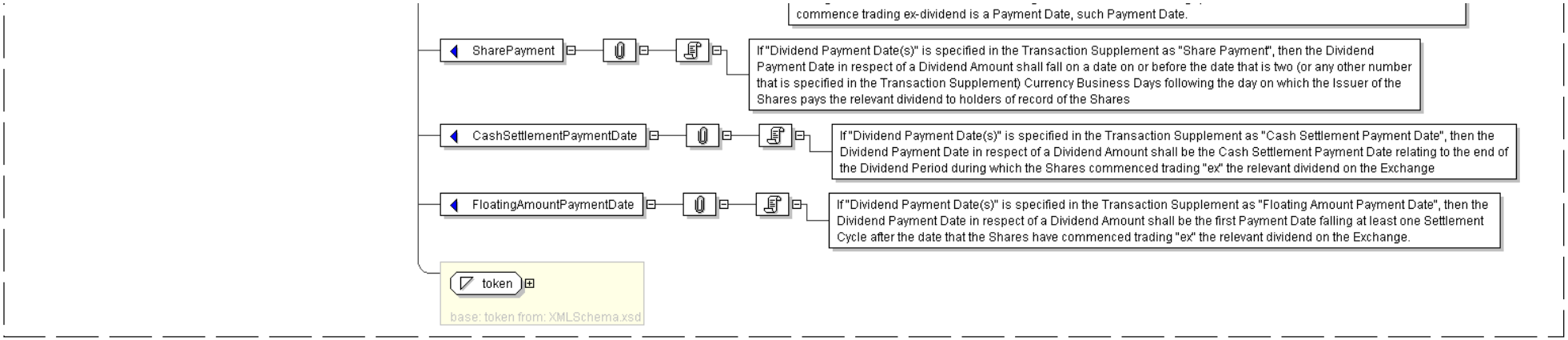
Simple Type: DividendDateReferenceEnum

Super-types:	xsd:token < **DividendDateReferenceEnum** (by restriction)
Sub-types:	None

Name	DividendDateReferenceEnum
Content	• Base XSD Type: token • *value* comes from list: {'ExDate'|'DividendPaymentDate'|'RecordDate'|'TerminationDate'|'EquityPaymentDate'|'FollowingPaymentDate'|'AdHocDate'|'CumulativeEquityPaid'|'CumulativeLiborPaid'|'CumulativeEquityExDiv'|'CumulativeLiborExDiv'|'SharePayment'|'CashSettlementPaymentDate'|'FloatingAmountPaymentDate'}
Documentation	The reference to a dividend date.

Diagram





Schema Component Representation

```
<xsd:simpleType name="DividendDateReferenceEnum">
   <xsd:restriction base=" xsd:token ">
      <xsd:enumeration value="ExDate"/>
      <xsd:enumeration value="DividendPaymentDate"/>
      <xsd:enumeration value="RecordDate"/>
      <xsd:enumeration value="TerminationDate"/>
      <xsd:enumeration value="EquityPaymentDate"/>
      <xsd:enumeration value="FollowingPaymentDate"/>
      <xsd:enumeration value="AdHocDate"/>
      <xsd:enumeration value="CumulativeEquityPaid"/>
      <xsd:enumeration value="CumulativeLiborPaid"/>
      <xsd:enumeration value="CumulativeEquityExDiv"/>
      <xsd:enumeration value="CumulativeLiborExDiv"/>
      <xsd:enumeration value="SharePayment"/>
      <xsd:enumeration value="CashSettlementPaymentDate"/>
      <xsd:enumeration value="FloatingAmountPaymentDate"/>
   </xsd:restriction>
</xsd:simpleType>
```

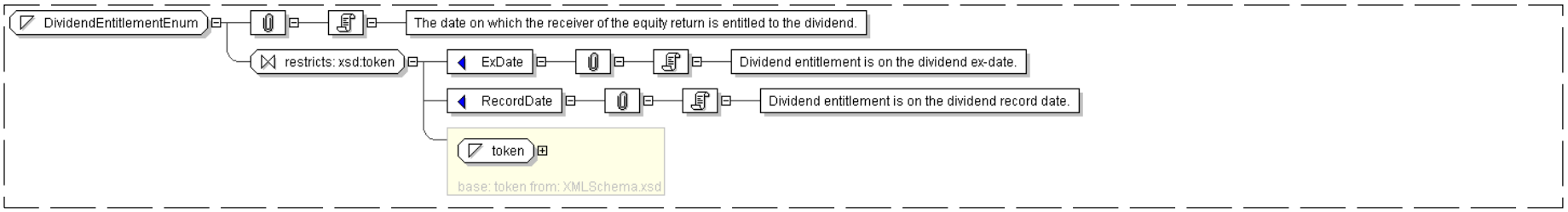
top

Simple Type: DividendEntitlementEnum

Super-types:	xsd:token < **DividendEntitlementEnum** (by restriction)
Sub-types:	None

Name	DividendEntitlementEnum
Content	• Base XSD Type: token • *value* comes from list: {'ExDate'|'RecordDate'}
Documentation	The date on which the receiver of the equity return is entitled to the dividend.

Diagram



Schema Component Representation

```
<xsd:simpleType name="DividendEntitlementEnum">
   <xsd:restriction base=" xsd:token ">
```

```
      <xsd:enumeration value="ExDate"/>
      <xsd:enumeration value="RecordDate"/>
   </xsd:restriction>
</xsd:simpleType>
```

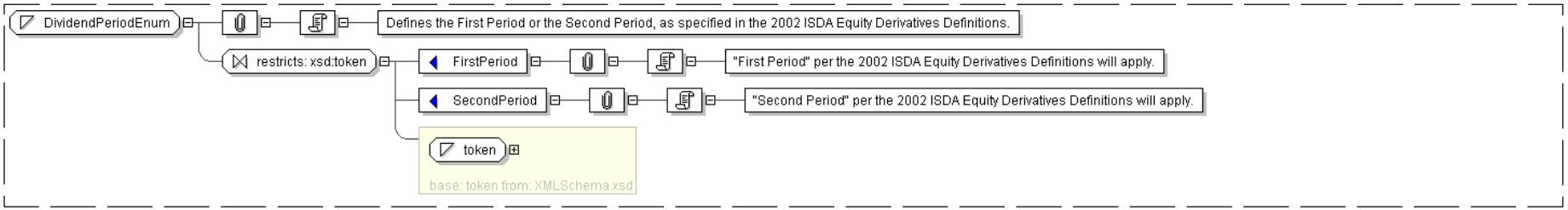
top

Simple Type: DividendPeriodEnum

Super-types:	xsd:token < **DividendPeriodEnum** (by restriction)
Sub-types:	None

Name	DividendPeriodEnum
Content	• Base XSD Type: token • *value* comes from list: {'FirstPeriod'|'SecondPeriod'}
Documentation	Defines the First Period or the Second Period, as specified in the 2002 ISDA Equity Derivatives Definitions.

Diagram



Schema Component Representation

```
<xsd:simpleType name="DividendPeriodEnum">
   <xsd:restriction base=" xsd:token ">
      <xsd:enumeration value="FirstPeriod"/>
      <xsd:enumeration value="SecondPeriod"/>
   </xsd:restriction>
</xsd:simpleType>
```

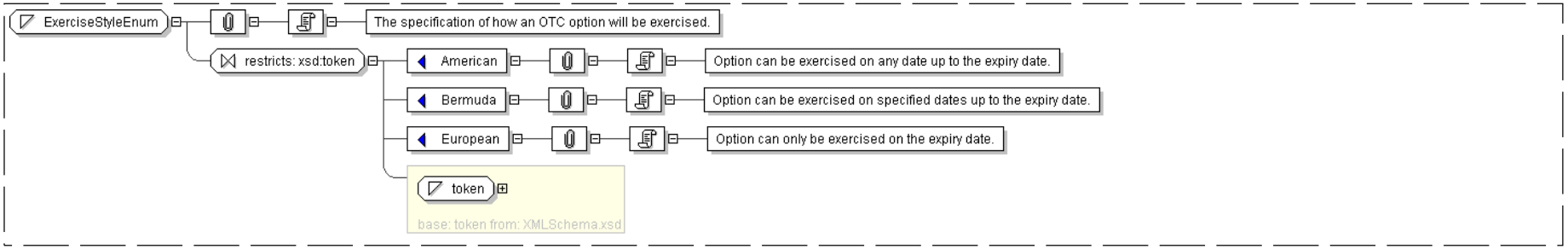
top

Simple Type: ExerciseStyleEnum

Super-types:	xsd:token < **ExerciseStyleEnum** (by restriction)
Sub-types:	None

Name	ExerciseStyleEnum
Content	• Base XSD Type: token • *value* comes from list: {'American'|'Bermuda'|'European'}
Documentation	The specification of how an OTC option will be exercised.

Diagram



Schema Component Representation

```
<xsd:simpleType name="ExerciseStyleEnum">
   <xsd:restriction base=" xsd:token ">
      <xsd:enumeration value="American"/>
      <xsd:enumeration value="Bermuda"/>
      <xsd:enumeration value="European"/>
   </xsd:restriction>
</xsd:simpleType>
```

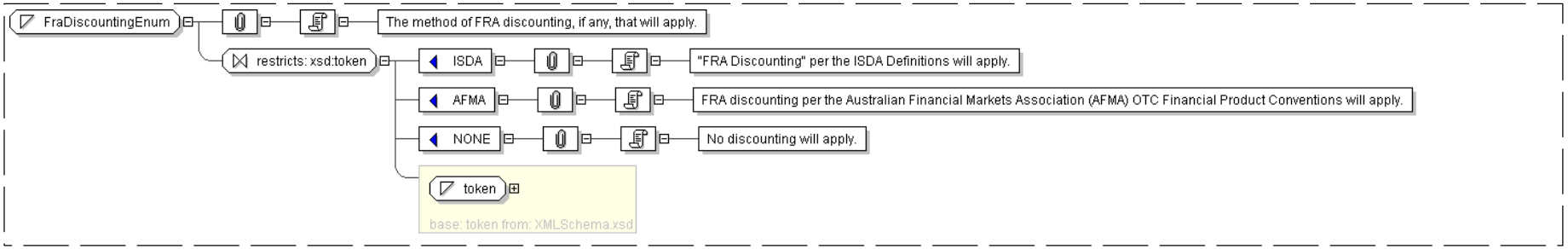
top

Simple Type: FraDiscountingEnum

Super-types:	xsd:token < **FraDiscountingEnum** (by restriction)
Sub-types:	None

Name	FraDiscountingEnum
Content	• Base XSD Type: token • *value* comes from list: {'ISDA'|'AFMA'|'NONE'}
Documentation	The method of FRA discounting, if any, that will apply.

Diagram



Schema Component Representation

```
<xsd:simpleType name="FraDiscountingEnum">
   <xsd:restriction base=" xsd:token ">
      <xsd:enumeration value="ISDA"/>
      <xsd:enumeration value="AFMA"/>
      <xsd:enumeration value="NONE"/>
   </xsd:restriction>
</xsd:simpleType>
```

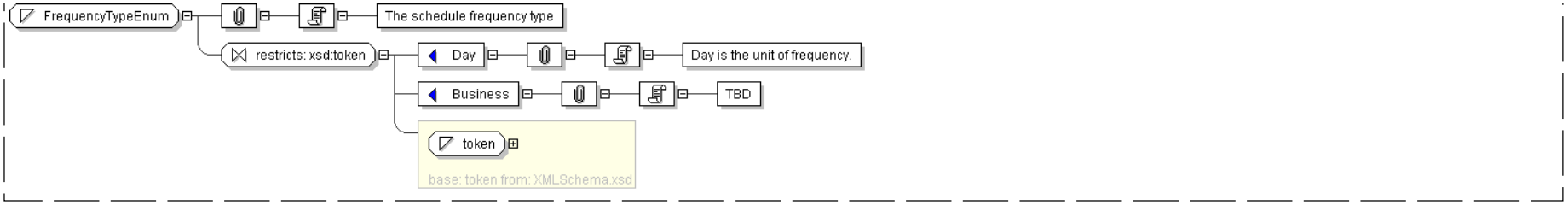
top

Simple Type: FrequencyTypeEnum

Super-types:	xsd:token < **FrequencyTypeEnum** (by restriction)
Sub-types:	None

Name	FrequencyTypeEnum
Content	• Base XSD Type: token • *value* comes from list: {'Day'|'Business'}
Documentation	The schedule frequency type

Diagram



Schema Component Representation

```
<xsd:simpleType name="FrequencyTypeEnum">
   <xsd:restriction base=" xsd:token ">
      <xsd:enumeration value="Day"/>
      <xsd:enumeration value="Business"/>
   </xsd:restriction>
</xsd:simpleType>
```

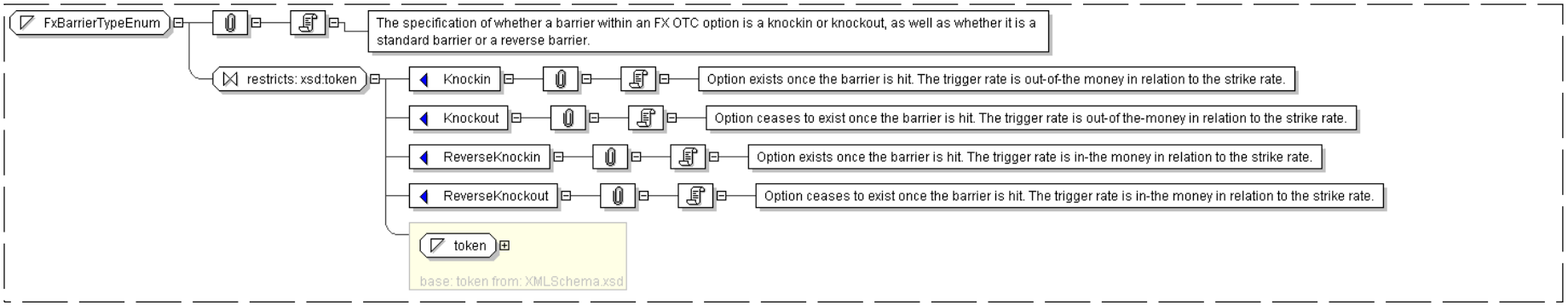
top

Simple Type: FxBarrierTypeEnum

Super-types:	xsd:token < **FxBarrierTypeEnum** (by restriction)
Sub-types:	None

Name	FxBarrierTypeEnum
Content	• Base XSD Type: token • *value* comes from list: {'Knockin'|'Knockout'|'ReverseKnockin'|'ReverseKnockout'}
Documentation	The specification of whether a barrier within an FX OTC option is a knockin or knockout, as well as whether it is a standard barrier or a reverse barrier.

Diagram



Schema Component Representation

```
<xsd:simpleType name="FxBarrierTypeEnum">
   <xsd:restriction base=" xsd:token ">
      <xsd:enumeration value="Knockin"/>
      <xsd:enumeration value="Knockout"/>
      <xsd:enumeration value="ReverseKnockin"/>
      <xsd:enumeration value="ReverseKnockout"/>
   </xsd:restriction>
</xsd:simpleType>
```

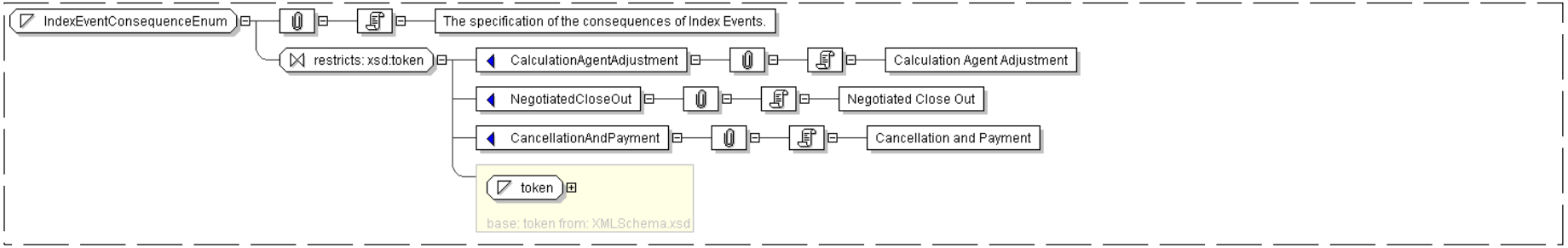
top

Simple Type: IndexEventConsequenceEnum

Super-types:	xsd:token < **IndexEventConsequenceEnum** (by restriction)
Sub-types:	None

Name	IndexEventConsequenceEnum
Content	• Base XSD Type: token • *value* comes from list: {'CalculationAgentAdjustment'|'NegotiatedCloseOut'|'CancellationAndPayment'}
Documentation	The specification of the consequences of Index Events.

Diagram



Schema Component Representation

```
<xsd:simpleType name="IndexEventConsequenceEnum">
   <xsd:restriction base=" xsd:token ">
      <xsd:enumeration value="CalculationAgentAdjustment"/>
      <xsd:enumeration value="NegotiatedCloseOut"/>
      <xsd:enumeration value="CancellationAndPayment"/>
   </xsd:restriction>
</xsd:simpleType>
```

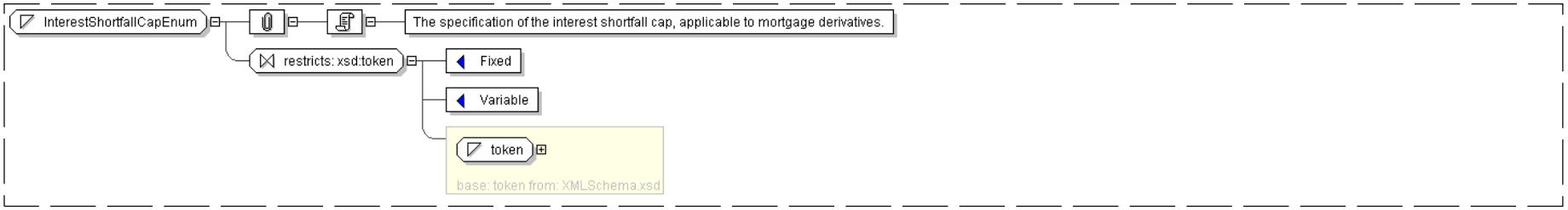
top

Simple Type: InterestShortfallCapEnum

Super-types:	xsd:token < **InterestShortfallCapEnum** (by restriction)
Sub-types:	None

Name	InterestShortfallCapEnum
Content	• Base XSD Type: token • *value* comes from list: {'Fixed'|'Variable'}
Documentation	The specification of the interest shortfall cap, applicable to mortgage derivatives.

Diagram



Schema Component Representation

```
<xsd:simpleType name="InterestShortfallCapEnum">
   <xsd:restriction base=" xsd:token ">
      <xsd:enumeration value="Fixed"/>
      <xsd:enumeration value="Variable"/>
   </xsd:restriction>
</xsd:simpleType>
```

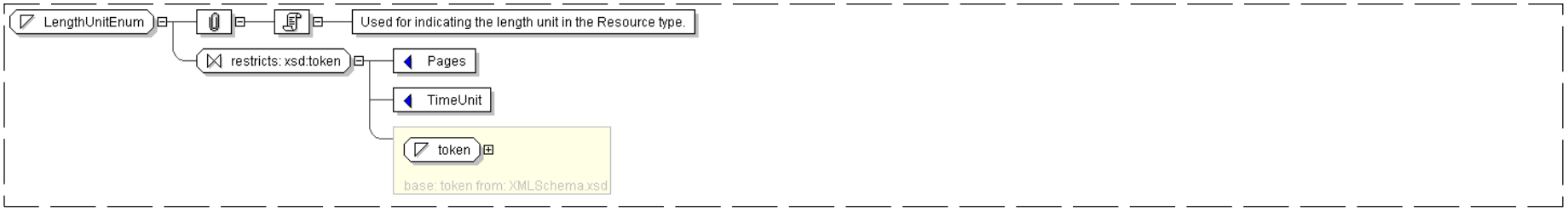
top

Simple Type: LengthUnitEnum

Super-types:	xsd:token < **LengthUnitEnum** (by restriction)
Sub-types:	None

Name	LengthUnitEnum
Content	• Base XSD Type: token • *value* comes from list: {'Pages'|'TimeUnit'}
Documentation	Used for indicating the length unit in the Resource type.

Diagram



Schema Component Representation

```
<xsd:simpleType name="LengthUnitEnum">
   <xsd:restriction base=" xsd:token ">
      <xsd:enumeration value="Pages"/>
      <xsd:enumeration value="TimeUnit"/>
   </xsd:restriction>
</xsd:simpleType>
```

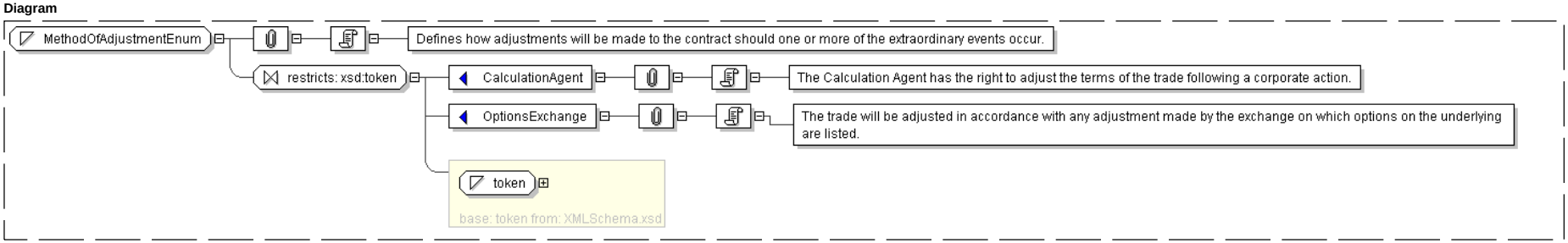
top

Simple Type: MethodOfAdjustmentEnum

Super-types:	xsd:token < **MethodOfAdjustmentEnum** (by restriction)
Sub-types:	None

Name	MethodOfAdjustmentEnum
Content	• Base XSD Type: token • *value* comes from list: {'CalculationAgent'|'OptionsExchange'}
Documentation	Defines how adjustments will be made to the contract should one or more of the extraordinary events occur.

Diagram



Schema Component Representation

```
<xsd:simpleType name="MethodOfAdjustmentEnum">
   <xsd:restriction base=" xsd:token ">
      <xsd:enumeration value="CalculationAgent"/>
      <xsd:enumeration value="OptionsExchange"/>
   </xsd:restriction>
</xsd:simpleType>
```

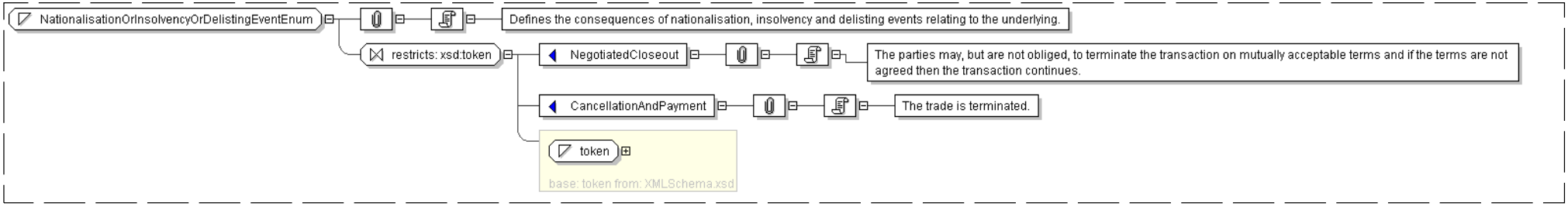
top

Simple Type: NationalisationOrInsolvencyOrDelistingEventEnum

Super-types:	xsd:token < **NationalisationOrInsolvencyOrDelistingEventEnum** (by restriction)
Sub-types:	None

Name	NationalisationOrInsolvencyOrDelistingEventEnum
Content	• Base XSD Type: token • *value* comes from list: {'NegotiatedCloseout'|'CancellationAndPayment'}
Documentation	Defines the consequences of nationalisation, insolvency and delisting events relating to the underlying.

Diagram



Schema Component Representation

```
<xsd:simpleType name="NationalisationOrInsolvencyOrDelistingEventEnum">
   <xsd:restriction base=" xsd:token ">
      <xsd:enumeration value="NegotiatedCloseout"/>
      <xsd:enumeration value="CancellationAndPayment"/>
   </xsd:restriction>
</xsd:simpleType>
```

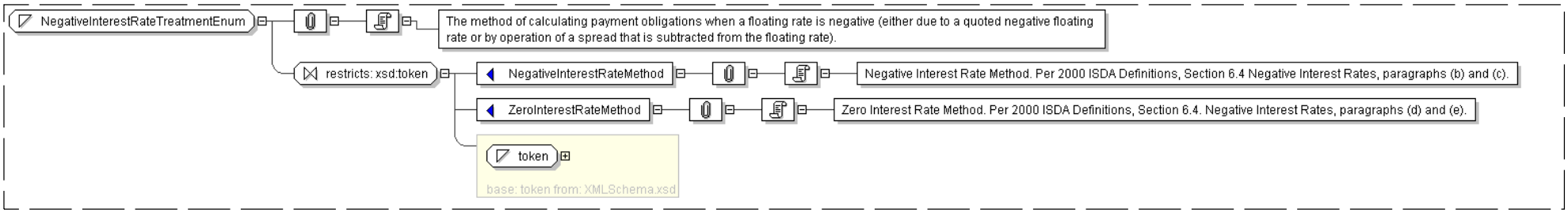
top

Simple Type: NegativeInterestRateTreatmentEnum

Super-types:	xsd:token < **NegativeInterestRateTreatmentEnum** (by restriction)
Sub-types:	None

Name	NegativeInterestRateTreatmentEnum
Content	• Base XSD Type: token • *value* comes from list: {'NegativeInterestRateMethod'|'ZeroInterestRateMethod'}
Documentation	The method of calculating payment obligations when a floating rate is negative (either due to a quoted negative floating rate or by operation of a spread that is subtracted from the floating rate).

Diagram



Schema Component Representation

```
<xsd:simpleType name="NegativeInterestRateTreatmentEnum">
   <xsd:restriction base=" xsd:token ">
```

```
      <xsd:enumeration value="NegativeInterestRateMethod"/>
      <xsd:enumeration value="ZeroInterestRateMethod"/>
   </xsd:restriction>
</xsd:simpleType>
```

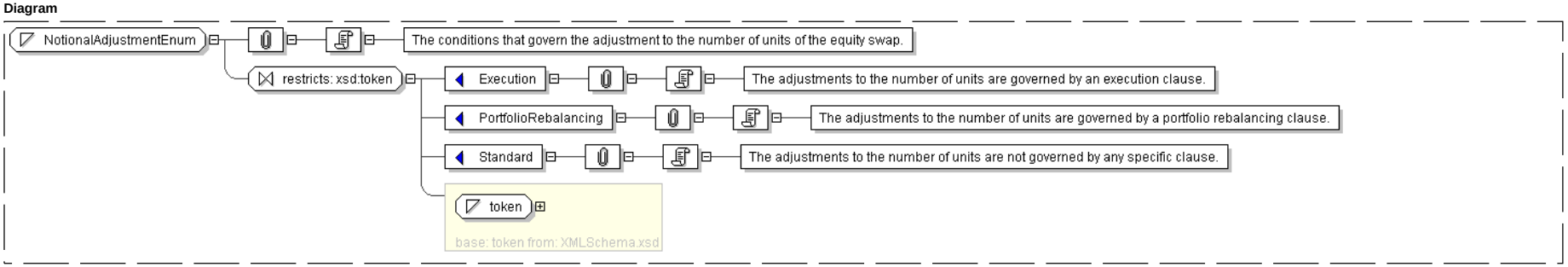
top

Simple Type: NotionalAdjustmentEnum

Super-types:	xsd:token < **NotionalAdjustmentEnum** (by restriction)
Sub-types:	None

Name	NotionalAdjustmentEnum
Content	• Base XSD Type: token • *value* comes from list: {'Execution'|'PortfolioRebalancing'|'Standard'}
Documentation	The conditions that govern the adjustment to the number of units of the equity swap.

Diagram



Schema Component Representation

```
<xsd:simpleType name="NotionalAdjustmentEnum">
   <xsd:restriction base=" xsd:token ">
      <xsd:enumeration value="Execution"/>
      <xsd:enumeration value="PortfolioRebalancing"/>
      <xsd:enumeration value="Standard"/>
   </xsd:restriction>
</xsd:simpleType>
```

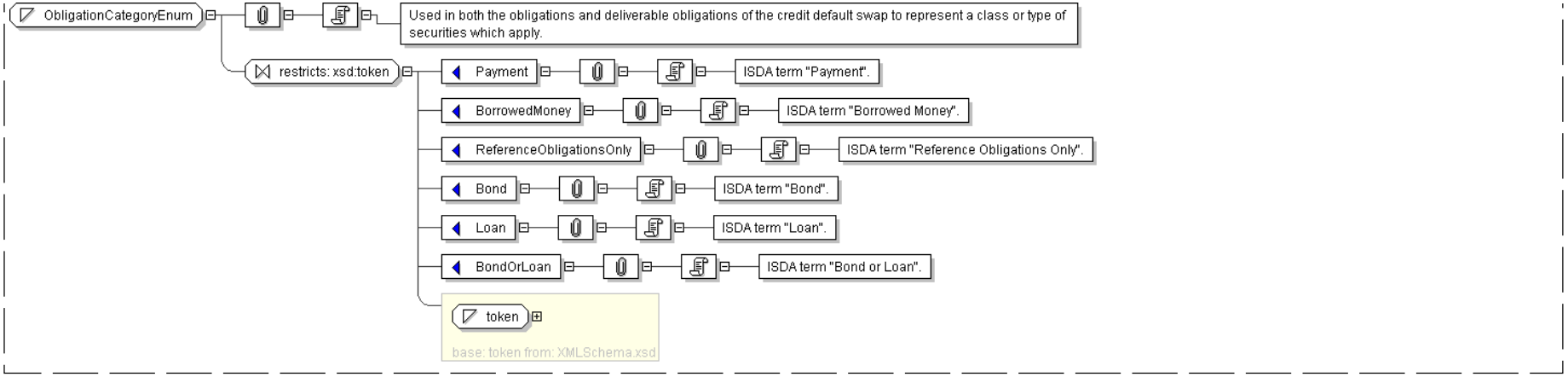
top

Simple Type: ObligationCategoryEnum

Super-types:	xsd:token < **ObligationCategoryEnum** (by restriction)
Sub-types:	None

Name	ObligationCategoryEnum
Content	• Base XSD Type: token • *value* comes from list: {'Payment'|'BorrowedMoney'|'ReferenceObligationsOnly'|'Bond'|'Loan'|'BondOrLoan'}
Documentation	Used in both the obligations and deliverable obligations of the credit default swap to represent a class or type of securities which apply.

Diagram



Schema Component Representation

```
<xsd:simpleType name="ObligationCategoryEnum">
   <xsd:restriction base=" xsd:token ">
      <xsd:enumeration value="Payment"/>
      <xsd:enumeration value="BorrowedMoney"/>
      <xsd:enumeration value="ReferenceObligationsOnly"/>
      <xsd:enumeration value="Bond"/>
      <xsd:enumeration value="Loan"/>
      <xsd:enumeration value="BondOrLoan"/>
   </xsd:restriction>
</xsd:simpleType>
```

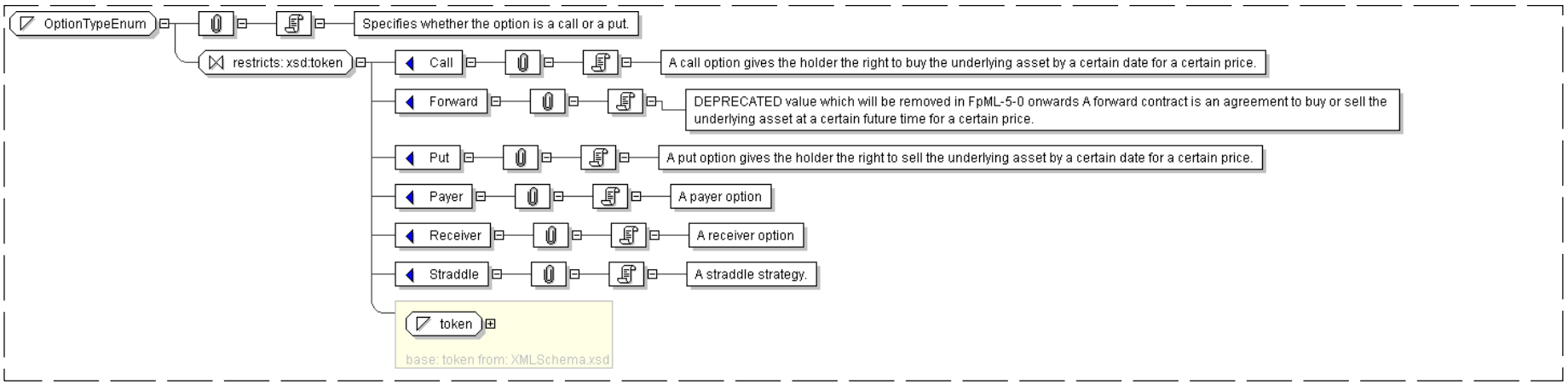
top

Simple Type: OptionTypeEnum

Super-types:	xsd:token < **OptionTypeEnum** (by restriction)
Sub-types:	None

Name	OptionTypeEnum
Content	• Base XSD Type: token • *value* comes from list: {'Call'|'Forward'|'Put'|'Payer'|'Receiver'|'Straddle'}
Documentation	Specifies whether the option is a call or a put.

Diagram



Schema Component Representation

```
<xsd:simpleType name="OptionTypeEnum">
   <xsd:restriction base=" xsd:token ">
      <xsd:enumeration value="Call"/>
      <xsd:enumeration value="Forward" deprecated="true" deprecatedReason="The optionType is to
      be used if the underlyer does not carry any mention of the resulting trade direction."/>
      <xsd:enumeration value="Put"/>
      <xsd:enumeration value="Payer"/>
      <xsd:enumeration value="Receiver"/>
      <xsd:enumeration value="Straddle"/>
   </xsd:restriction>
</xsd:simpleType>
```

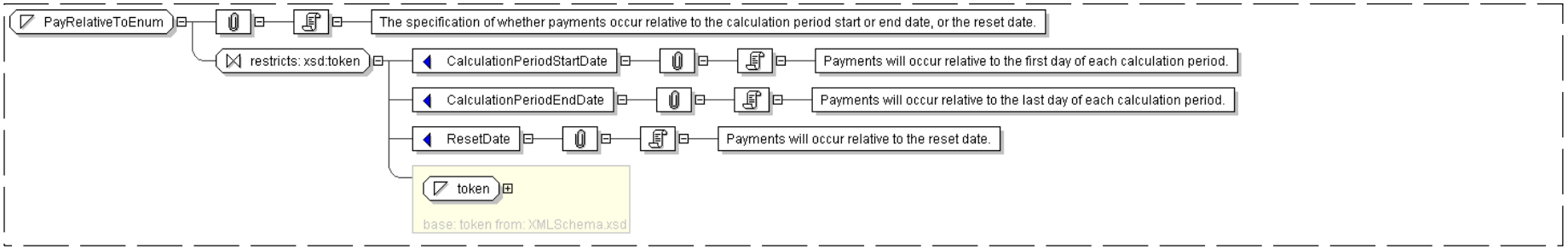
top

Simple Type: PayRelativeToEnum

Super-types:	xsd:token < **PayRelativeToEnum** (by restriction)
Sub-types:	None

Name	PayRelativeToEnum
Content	• Base XSD Type: token • *value* comes from list: {'CalculationPeriodStartDate'|'CalculationPeriodEndDate'|'ResetDate'}
Documentation	The specification of whether payments occur relative to the calculation period start or end date, or the reset date.

Diagram



Schema Component Representation

```
<xsd:simpleType name="PayRelativeToEnum">
   <xsd:restriction base=" xsd:token ">
      <xsd:enumeration value="CalculationPeriodStartDate"/>
      <xsd:enumeration value="CalculationPeriodEndDate"/>
      <xsd:enumeration value="ResetDate"/>
   </xsd:restriction>
</xsd:simpleType>
```

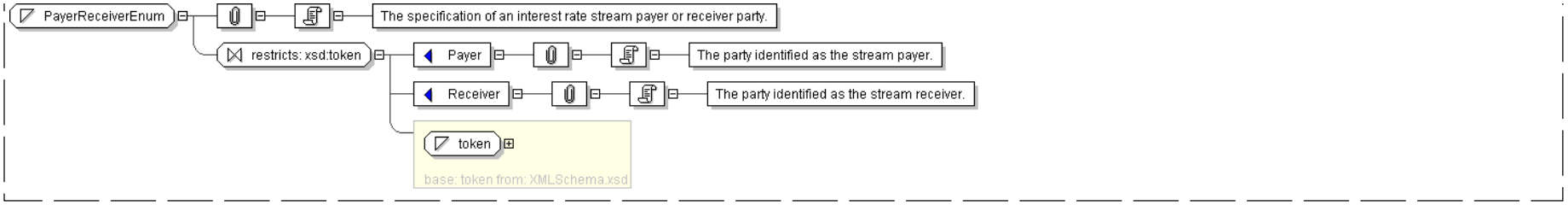
top

Simple Type: PayerReceiverEnum

Super-types:	xsd:token < **PayerReceiverEnum** (by restriction)
Sub-types:	None

Name	PayerReceiverEnum
Content	• Base XSD Type: token • *value* comes from list: {'Payer'|'Receiver'}
Documentation	The specification of an interest rate stream payer or receiver party.

Diagram



Schema Component Representation

```
<xsd:simpleType name="PayerReceiverEnum">
   <xsd:restriction base=" xsd:token ">
      <xsd:enumeration value="Payer"/>
      <xsd:enumeration value="Receiver"/>
   </xsd:restriction>
</xsd:simpleType>
```

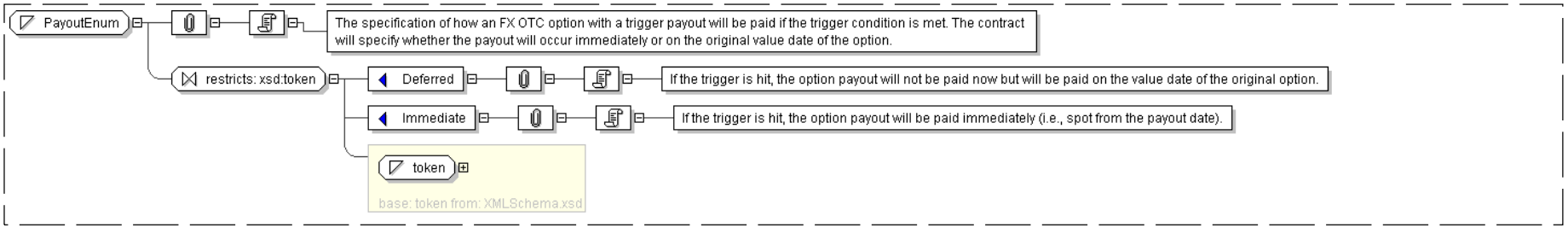
top

Simple Type: PayoutEnum

Super-types:	xsd:token < **PayoutEnum** (by restriction)
Sub-types:	None

Name	PayoutEnum
Content	• Base XSD Type: token • *value* comes from list: {'Deferred'|'Immediate'}
Documentation	The specification of how an FX OTC option with a trigger payout will be paid if the trigger condition is met. The contract will specify whether the payout will occur immediately or on the original value date of the option.

Diagram



Schema Component Representation

```
<xsd:simpleType name="PayoutEnum">
   <xsd:restriction base=" xsd:token ">
      <xsd:enumeration value="Deferred"/>
      <xsd:enumeration value="Immediate"/>
   </xsd:restriction>
</xsd:simpleType>
```

top

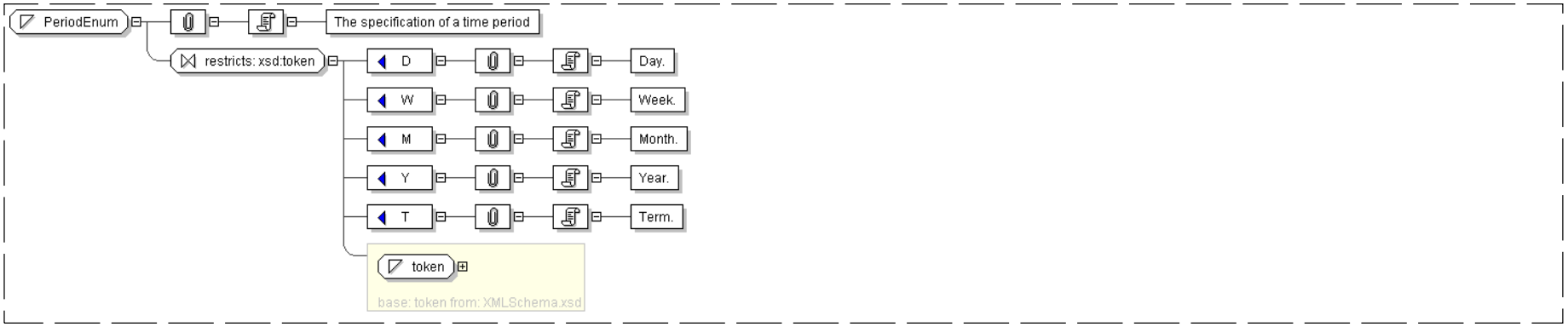
Simple Type: PeriodEnum

Super-types:	xsd:token < **PeriodEnum** (by restriction)
Sub-types:	None

Name	PeriodEnum
Content	• Base XSD Type: token • *value* comes from list: {'D'|'W'|'M'|'Y'|'T'}

Documentation	The specification of a time period

Diagram



Schema Component Representation

```
<xsd:simpleType name="PeriodEnum">
   <xsd:restriction base=" xsd:token ">
      <xsd:enumeration value="D"/>
      <xsd:enumeration value="W"/>
      <xsd:enumeration value="M"/>
      <xsd:enumeration value="Y"/>
      <xsd:enumeration value="T"/>
   </xsd:restriction>
</xsd:simpleType>
```

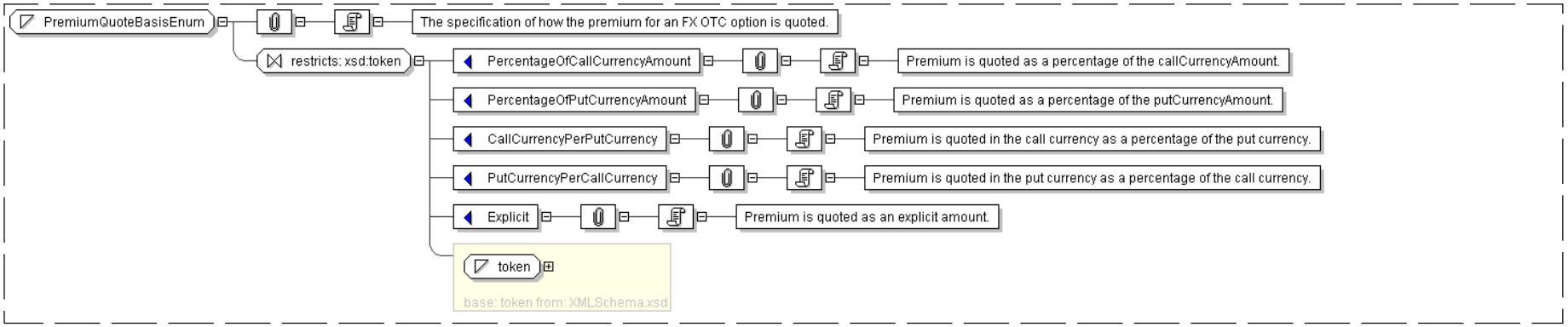
top

Simple Type: PremiumQuoteBasisEnum

Super-types:	xsd:token < **PremiumQuoteBasisEnum** (by restriction)
Sub-types:	None

Name	PremiumQuoteBasisEnum
Content	• Base XSD Type: token • *value* comes from list: {'PercentageOfCallCurrencyAmount'|'PercentageOfPutCurrencyAmount'|'CallCurrencyPerPutCurrency'|'PutCurrencyPerCallCurrency'|'Explicit'}
Documentation	The specification of how the premium for an FX OTC option is quoted.

Diagram



Schema Component Representation

```
<xsd:simpleType name="PremiumQuoteBasisEnum">
   <xsd:restriction base=" xsd:token ">
```

```
      <xsd:enumeration value="PercentageOfCallCurrencyAmount"/>
      <xsd:enumeration value="PercentageOfPutCurrencyAmount"/>
      <xsd:enumeration value="CallCurrencyPerPutCurrency"/>
      <xsd:enumeration value="PutCurrencyPerCallCurrency"/>
      <xsd:enumeration value="Explicit"/>
   </xsd:restriction>
</xsd:simpleType>
```

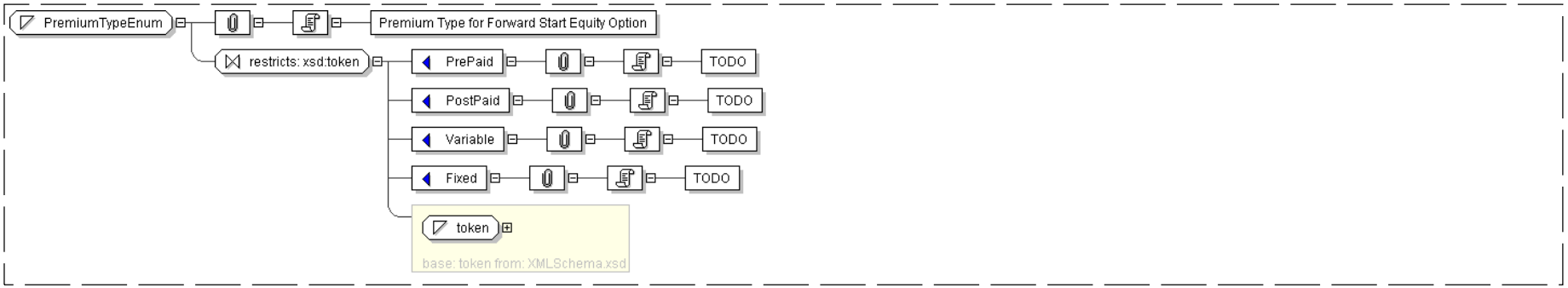
top

Simple Type: PremiumTypeEnum

Super-types:	xsd:token < **PremiumTypeEnum** (by restriction)
Sub-types:	None

Name	PremiumTypeEnum
Content	• Base XSD Type: token • *value* comes from list: {'PrePaid'|'PostPaid'|'Variable'|'Fixed'}
Documentation	Premium Type for Forward Start Equity Option

Diagram



Schema Component Representation

```
<xsd:simpleType name="PremiumTypeEnum">
   <xsd:restriction base=" xsd:token ">
      <xsd:enumeration value="PrePaid"/>
      <xsd:enumeration value="PostPaid"/>
      <xsd:enumeration value="Variable"/>
      <xsd:enumeration value="Fixed"/>
   </xsd:restriction>
</xsd:simpleType>
```

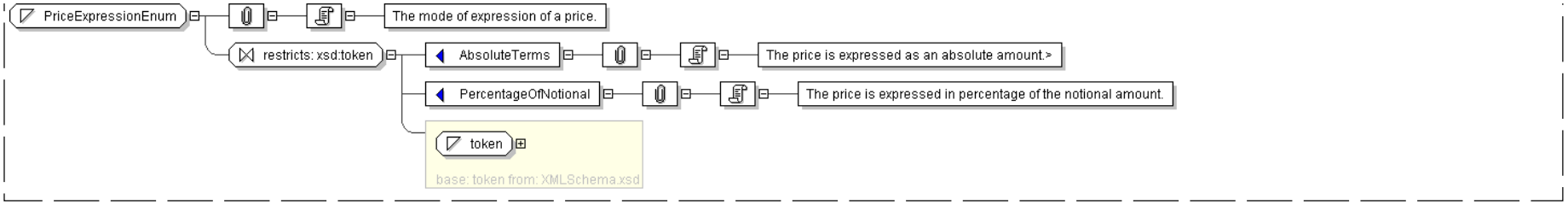
top

Simple Type: PriceExpressionEnum

Super-types:	xsd:token < **PriceExpressionEnum** (by restriction)
Sub-types:	None

Name	PriceExpressionEnum
Content	• Base XSD Type: token • *value* comes from list: {'AbsoluteTerms'|'PercentageOfNotional'}
Documentation	The mode of expression of a price.

Diagram



Schema Component Representation

```
<xsd:simpleType name="PriceExpressionEnum">
   <xsd:restriction base=" xsd:token ">
      <xsd:enumeration value="AbsoluteTerms"/>
      <xsd:enumeration value="PercentageOfNotional"/>
   </xsd:restriction>
</xsd:simpleType>
```

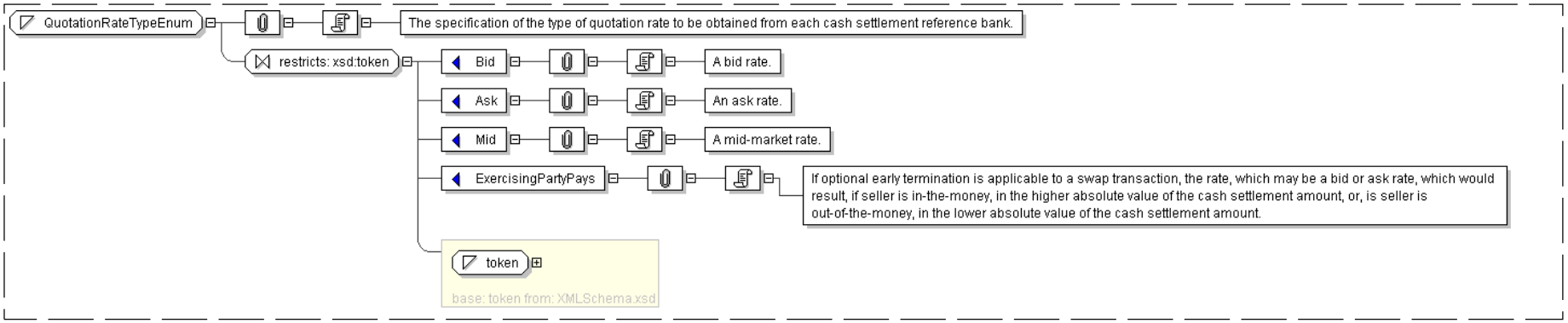
top

Simple Type: QuotationRateTypeEnum

Super-types:	xsd:token < **QuotationRateTypeEnum** (by restriction)
Sub-types:	None

Name	QuotationRateTypeEnum
Content	• Base XSD Type: token • *value* comes from list: {'Bid'|'Ask'|'Mid'|'ExercisingPartyPays'}
Documentation	The specification of the type of quotation rate to be obtained from each cash settlement reference bank.

Diagram



Schema Component Representation

```
<xsd:simpleType name="QuotationRateTypeEnum">
   <xsd:restriction base=" xsd:token ">
      <xsd:enumeration value="Bid"/>
      <xsd:enumeration value="Ask"/>
      <xsd:enumeration value="Mid"/>
      <xsd:enumeration value="ExercisingPartyPays"/>
   </xsd:restriction>
</xsd:simpleType>
```

top

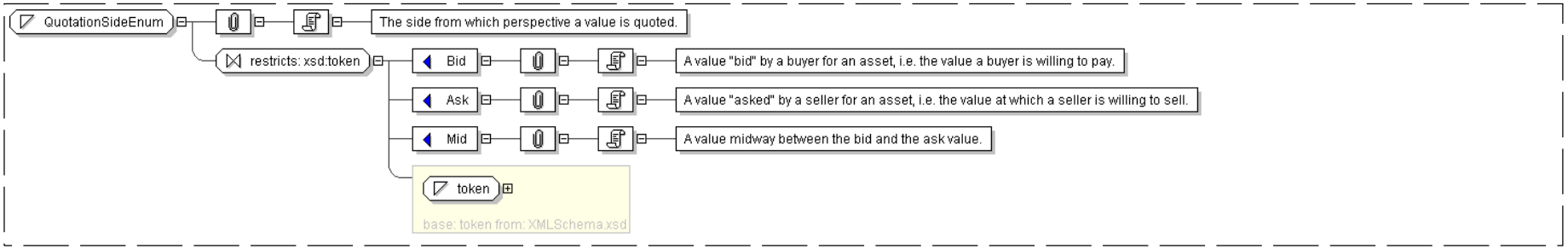
Simple Type: QuotationSideEnum

Super-types:	xsd:token < **QuotationSideEnum** (by restriction)

Sub-types:	None

Name	QuotationSideEnum
Content	• Base XSD Type: token • *value* comes from list: {'Bid'|'Ask'|'Mid'}
Documentation	The side from which perspective a value is quoted.

Diagram



Schema Component Representation

```
<xsd:simpleType name="QuotationSideEnum">
   <xsd:restriction base=" xsd:token ">
      <xsd:enumeration value="Bid"/>
      <xsd:enumeration value="Ask"/>
      <xsd:enumeration value="Mid"/>
   </xsd:restriction>
</xsd:simpleType>
```

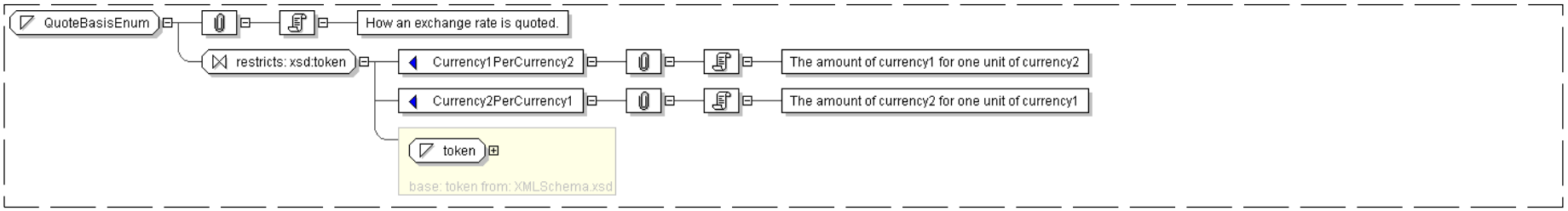
top

Simple Type: QuoteBasisEnum

Super-types:	xsd:token < **QuoteBasisEnum** (by restriction)
Sub-types:	None

Name	QuoteBasisEnum
Content	• Base XSD Type: token • *value* comes from list: {'Currency1PerCurrency2'|'Currency2PerCurrency1'}
Documentation	How an exchange rate is quoted.

Diagram



Schema Component Representation

```
<xsd:simpleType name="QuoteBasisEnum">
   <xsd:restriction base=" xsd:token ">
      <xsd:enumeration value="Currency1PerCurrency2"/>
      <xsd:enumeration value="Currency2PerCurrency1"/>
   </xsd:restriction>
</xsd:simpleType>
```

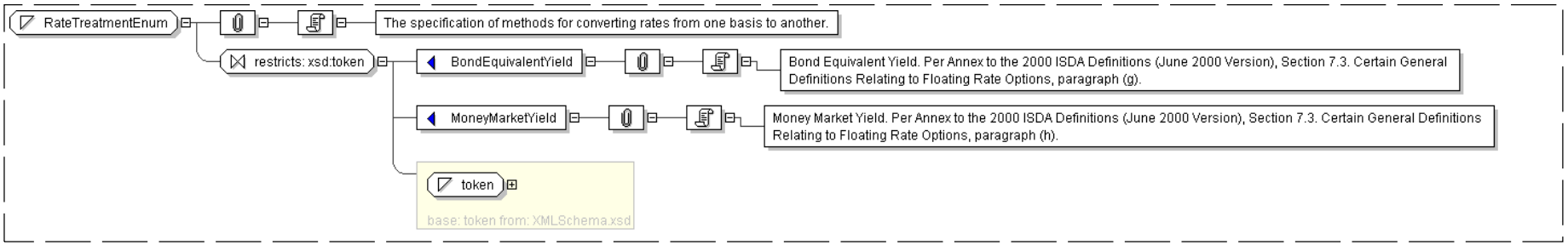
top

Simple Type: RateTreatmentEnum

Super-types:	xsd:token < **RateTreatmentEnum** (by restriction)
Sub-types:	None

Name	RateTreatmentEnum
Content	• Base XSD Type: token • *value* comes from list: {'BondEquivalentYield'|'MoneyMarketYield'}
Documentation	The specification of methods for converting rates from one basis to another.

Diagram



Schema Component Representation

```
<xsd:simpleType name="RateTreatmentEnum">
   <xsd:restriction base=" xsd:token ">
      <xsd:enumeration value="BondEquivalentYield"/>
      <xsd:enumeration value="MoneyMarketYield"/>
   </xsd:restriction>
</xsd:simpleType>
```

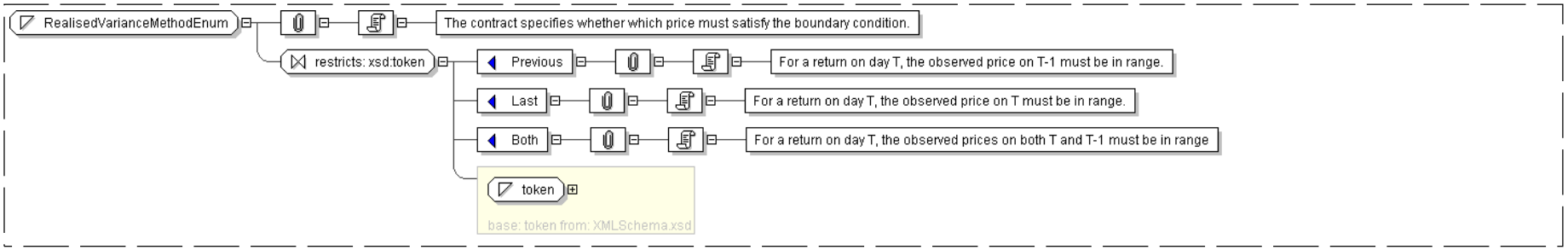
top

Simple Type: RealisedVarianceMethodEnum

Super-types:	xsd:token < **RealisedVarianceMethodEnum** (by restriction)
Sub-types:	None

Name	RealisedVarianceMethodEnum
Content	• Base XSD Type: token • *value* comes from list: {'Previous'|'Last'|'Both'}
Documentation	The contract specifies whether which price must satisfy the boundary condition.

Diagram



Schema Component Representation

```
<xsd:simpleType name="RealisedVarianceMethodEnum">
   <xsd:restriction base=" xsd:token ">
      <xsd:enumeration value="Previous"/>
```

```
      <xsd:enumeration value="Last"/>
      <xsd:enumeration value="Both"/>
   </xsd:restriction>
</xsd:simpleType>
```

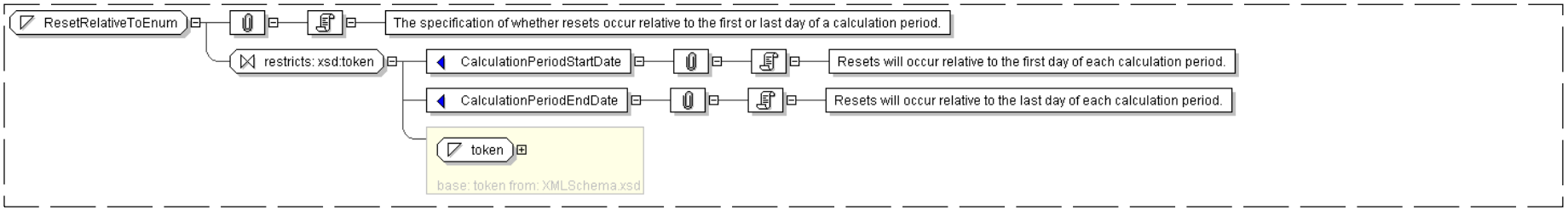
top

Simple Type: ResetRelativeToEnum

Super-types:	xsd:token < **ResetRelativeToEnum** (by restriction)
Sub-types:	None

Name	ResetRelativeToEnum
Content	• Base XSD Type: token • *value* comes from list: {'CalculationPeriodStartDate'|'CalculationPeriodEndDate'}
Documentation	The specification of whether resets occur relative to the first or last day of a calculation period.

Diagram



Schema Component Representation

```
<xsd:simpleType name="ResetRelativeToEnum">
   <xsd:restriction base=" xsd:token ">
      <xsd:enumeration value="CalculationPeriodStartDate"/>
      <xsd:enumeration value="CalculationPeriodEndDate"/>
   </xsd:restriction>
</xsd:simpleType>
```

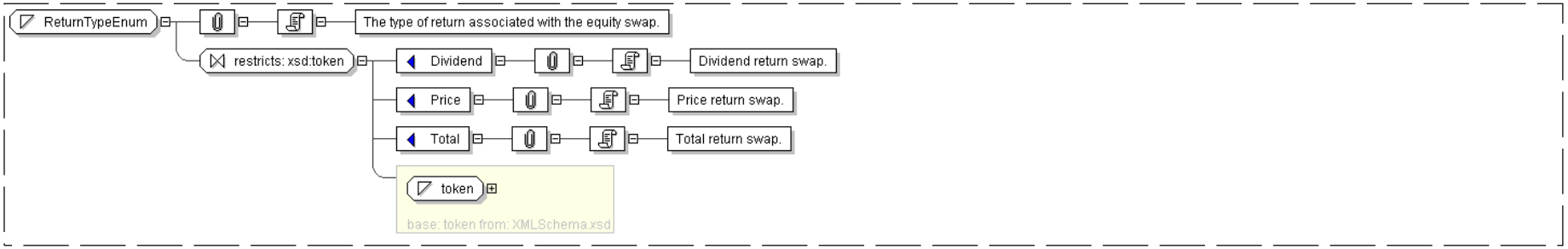
top

Simple Type: ReturnTypeEnum

Super-types:	xsd:token < **ReturnTypeEnum** (by restriction)
Sub-types:	None

Name	ReturnTypeEnum
Content	• Base XSD Type: token • *value* comes from list: {'Dividend'|'Price'|'Total'}
Documentation	The type of return associated with the equity swap.

Diagram



Schema Component Representation

```
<xsd:simpleType name="ReturnTypeEnum">
   <xsd:restriction base=" xsd:token ">
      <xsd:enumeration value="Dividend"/>
      <xsd:enumeration value="Price"/>
      <xsd:enumeration value="Total"/>
   </xsd:restriction>
</xsd:simpleType>
```

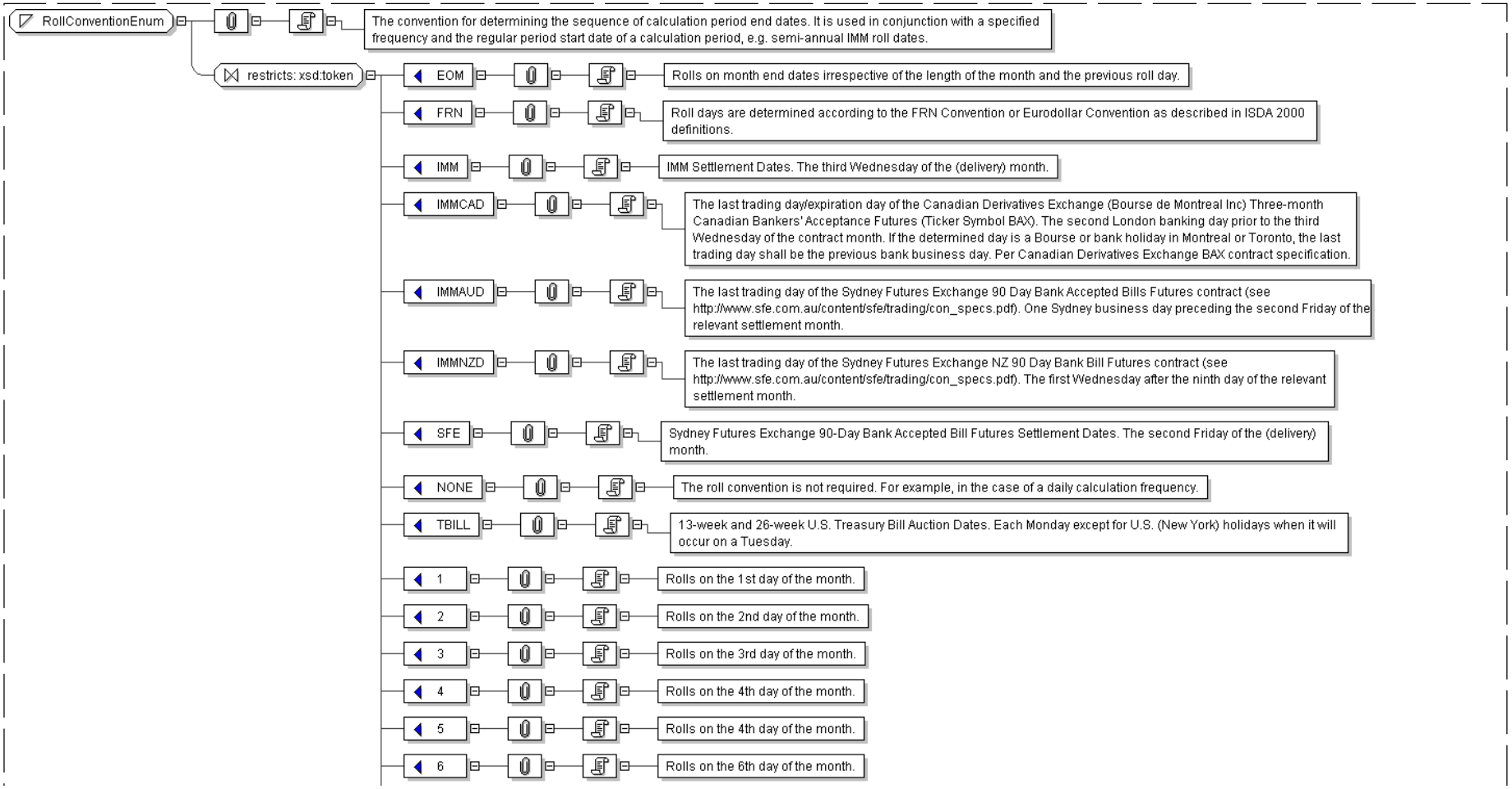
top

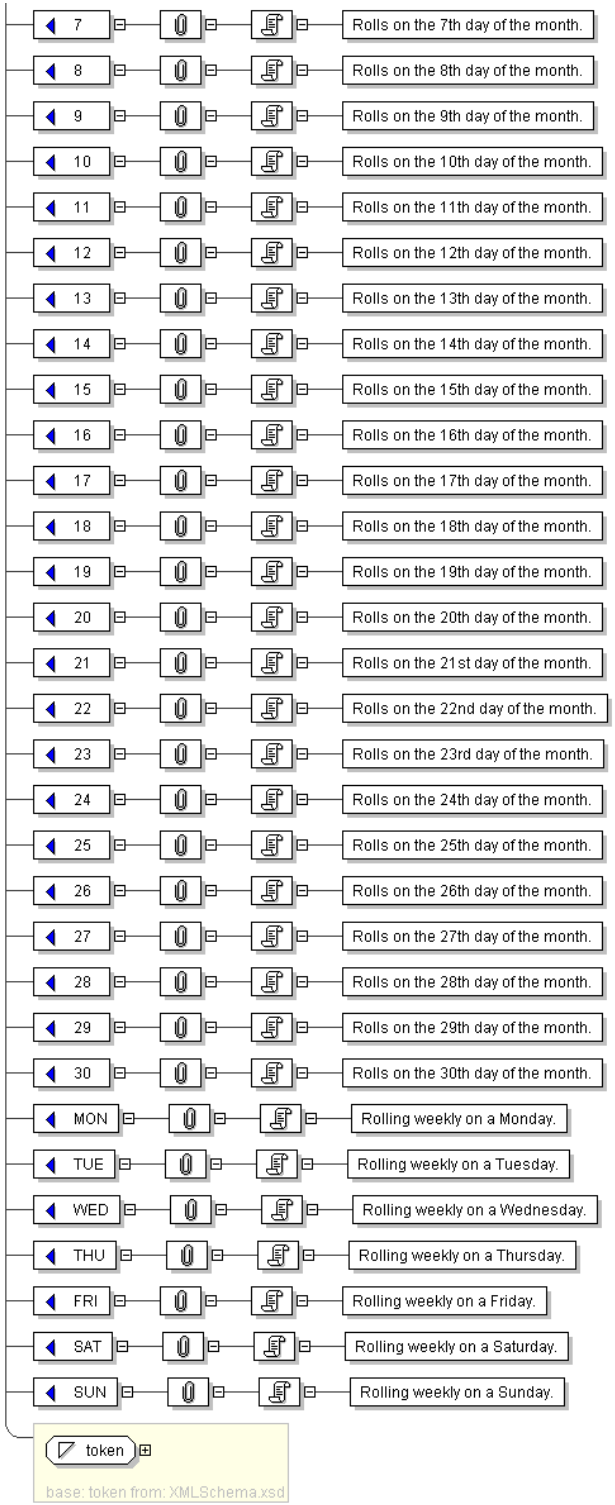
Simple Type: RollConventionEnum

Super-types:	xsd:token < **RollConventionEnum** (by restriction)
Sub-types:	None

Name	RollConventionEnum
Content	• Base XSD Type: token • *value* comes from list: {'EOM'|'FRN'|'IMM'|'IMMCAD'|'IMMAUD'|'IMMNZD'|'SFE'|'NONE'|'TBILL'|'1'|'2'|'3'|'4'|'5'|'6'|'7'|'8'|'9'|'10'|'11'|'12'|'13'|'14'|'15'|'16'|'17'|'18'|'19'|'20'|'21'|'22'|'23'|'24'|'25'|'26'|'27'|'28'|'29'|'30'|'MON'|'TUE'|'WED'|'THU'|'FRI'|'SAT'|'SUN'}
Documentation	The convention for determining the sequence of calculation period end dates. It is used in conjunction with a specified frequency and the regular period start date of a calculation period, e.g. semi-annual IMM roll dates.

Diagram




7
Rolls on the 7th day of the month.
8
Rolls on the 8th day of the month.
9
Rolls on the 9th day of the month.
10
Rolls on the 10th day of the month.
11
Rolls on the 11th day of the month.
12
Rolls on the 12th day of the month.
13
Rolls on the 13th day of the month.
14
Rolls on the 14th day of the month.
15
Rolls on the 15th day of the month.
16
Rolls on the 16th day of the month.
17
Rolls on the 17th day of the month.
18
Rolls on the 18th day of the month.
19
Rolls on the 19th day of the month.
20
Rolls on the 20th day of the month.
21
Rolls on the 21st day of the month.
22
Rolls on the 22nd day of the month.
23
Rolls on the 23rd day of the month.
24
Rolls on the 24th day of the month.
25
Rolls on the 25th day of the month.
26
Rolls on the 26th day of the month.
27
Rolls on the 27th day of the month.
28
Rolls on the 28th day of the month.
29
Rolls on the 29th day of the month.
30
Rolls on the 30th day of the month.
MON
Rolling weekly on a Monday.
TUE
Rolling weekly on a Tuesday.
WED
Rolling weekly on a Wednesday.
THU
Rolling weekly on a Thursday.
FRI
Rolling weekly on a Friday.
SAT
Rolling weekly on a Saturday.
SUN
Rolling weekly on a Sunday.
token
base: token from: XMLSchema.xsd

Schema Component Representation

```
<xsd:simpleType name="RollConventionEnum">
   <xsd:restriction base=" xsd:token ">
      <xsd:enumeration value="EOM"/>
      <xsd:enumeration value="FRN"/>
      <xsd:enumeration value="IMM"/>
      <xsd:enumeration value="IMMCAD"/>
      <xsd:enumeration value="IMMAUD"/>
      <xsd:enumeration value="IMMNZD"/>
      <xsd:enumeration value="SFE"/>
      <xsd:enumeration value="NONE"/>
      <xsd:enumeration value="TBILL"/>
      <xsd:enumeration value="1"/>
      <xsd:enumeration value="2"/>
      <xsd:enumeration value="3"/>
      <xsd:enumeration value="4"/>
      <xsd:enumeration value="5"/>
      <xsd:enumeration value="6"/>
      <xsd:enumeration value="7"/>
      <xsd:enumeration value="8"/>
      <xsd:enumeration value="9"/>
      <xsd:enumeration value="10"/>
      <xsd:enumeration value="11"/>
      <xsd:enumeration value="12"/>
      <xsd:enumeration value="13"/>
      <xsd:enumeration value="14"/>
      <xsd:enumeration value="15"/>
      <xsd:enumeration value="16"/>
      <xsd:enumeration value="17"/>
      <xsd:enumeration value="18"/>
      <xsd:enumeration value="19"/>
      <xsd:enumeration value="20"/>
      <xsd:enumeration value="21"/>
      <xsd:enumeration value="22"/>
      <xsd:enumeration value="23"/>
      <xsd:enumeration value="24"/>
      <xsd:enumeration value="25"/>
      <xsd:enumeration value="26"/>
      <xsd:enumeration value="27"/>
      <xsd:enumeration value="28"/>
      <xsd:enumeration value="29"/>
      <xsd:enumeration value="30"/>
      <xsd:enumeration value="MON"/>
      <xsd:enumeration value="TUE"/>
      <xsd:enumeration value="WED"/>
      <xsd:enumeration value="THU"/>
      <xsd:enumeration value="FRI"/>
      <xsd:enumeration value="SAT"/>
      <xsd:enumeration value="SUN"/>
   </xsd:restriction>
</xsd:simpleType>
```

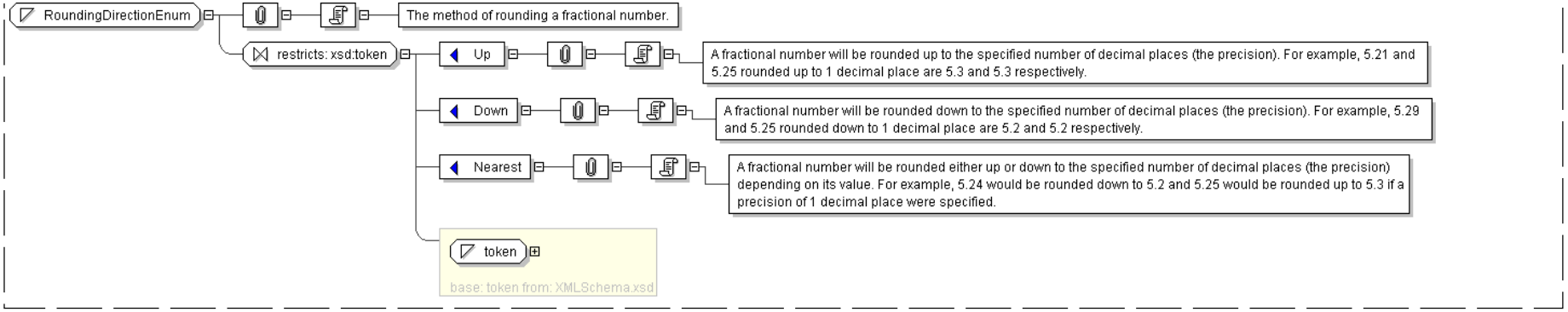
top

Simple Type: RoundingDirectionEnum

Super-types:	xsd:token < **RoundingDirectionEnum** (by restriction)
Sub-types:	None

Name	RoundingDirectionEnum
Content	• Base XSD Type: token • *value* comes from list: {'Up'|'Down'|'Nearest'}
Documentation	The method of rounding a fractional number.

Diagram



Schema Component Representation

```
<xsd:simpleType name="RoundingDirectionEnum">
   <xsd:restriction base=" xsd:token ">
      <xsd:enumeration value="Up"/>
      <xsd:enumeration value="Down"/>
      <xsd:enumeration value="Nearest"/>
   </xsd:restriction>
</xsd:simpleType>
```

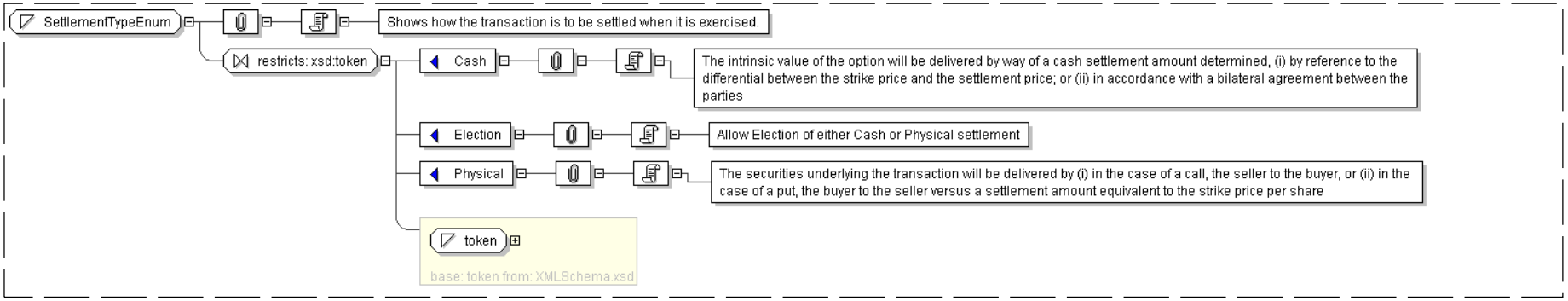
top

Simple Type: SettlementTypeEnum

Super-types:	xsd:token < **SettlementTypeEnum** (by restriction)
Sub-types:	None

Name	SettlementTypeEnum
Content	• Base XSD Type: token • *value* comes from list: {'Cash'|'Election'|'Physical'}
Documentation	Shows how the transaction is to be settled when it is exercised.

Diagram



Schema Component Representation

```
<xsd:simpleType name="SettlementTypeEnum">
   <xsd:restriction base=" xsd:token ">
      <xsd:enumeration value="Cash"/>
      <xsd:enumeration value="Election"/>
      <xsd:enumeration value="Physical"/>
   </xsd:restriction>
</xsd:simpleType>
```

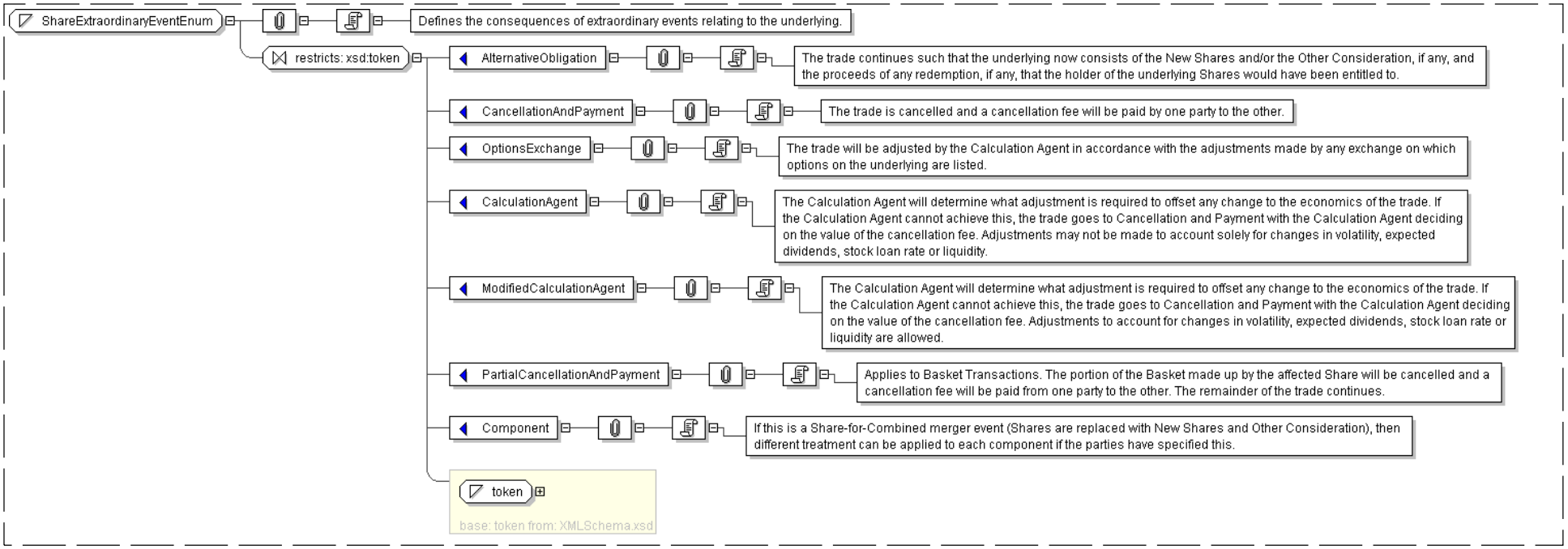
top

Simple Type: ShareExtraordinaryEventEnum

Super-types:	xsd:token < **ShareExtraordinaryEventEnum** (by restriction)
Sub-types:	None

Name	ShareExtraordinaryEventEnum
Content	• Base XSD Type: token • *value* comes from list: {'AlternativeObligation'|'CancellationAndPayment'|'OptionsExchange'|'CalculationAgent'|'ModifiedCalculationAgent'|'PartialCancellationAndPayment'|'Component'}
Documentation	Defines the consequences of extraordinary events relating to the underlying.

Diagram



Schema Component Representation

```
<xsd:simpleType name="ShareExtraordinaryEventEnum">
   <xsd:restriction base=" xsd:token ">
      <xsd:enumeration value="AlternativeObligation"/>
      <xsd:enumeration value="CancellationAndPayment"/>
      <xsd:enumeration value="OptionsExchange"/>
      <xsd:enumeration value="CalculationAgent"/>
      <xsd:enumeration value="ModifiedCalculationAgent"/>
      <xsd:enumeration value="PartialCancellationAndPayment"/>
      <xsd:enumeration value="Component"/>
   </xsd:restriction>
</xsd:simpleType>
```

top

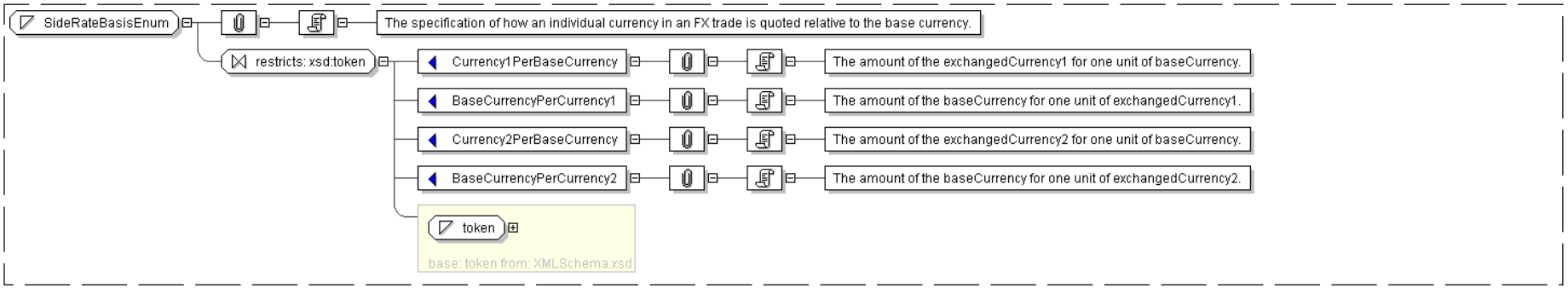
Simple Type: SideRateBasisEnum

Super-types:	xsd:token < **SideRateBasisEnum** (by restriction)
Sub-types:	None

Name	SideRateBasisEnum

Content	• Base XSD Type: token • *value* comes from list: {'Currency1PerBaseCurrency'|'BaseCurrencyPerCurrency1'|'Currency2PerBaseCurrency'|'BaseCurrencyPerCurrency2'}
Documentation	The specification of how an individual currency in an FX trade is quoted relative to the base currency.

Diagram



Schema Component Representation

```
<xsd:simpleType name="SideRateBasisEnum">
   <xsd:restriction base=" xsd:token ">
      <xsd:enumeration value="Currency1PerBaseCurrency"/>
      <xsd:enumeration value="BaseCurrencyPerCurrency1"/>
      <xsd:enumeration value="Currency2PerBaseCurrency"/>
      <xsd:enumeration value="BaseCurrencyPerCurrency2"/>
   </xsd:restriction>
</xsd:simpleType>
```

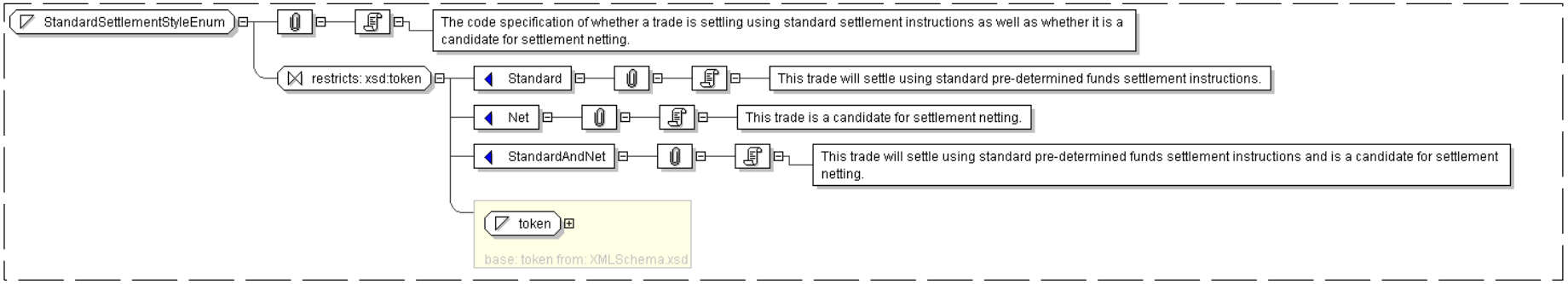
top

Simple Type: StandardSettlementStyleEnum

Super-types:	xsd:token < **StandardSettlementStyleEnum** (by restriction)
Sub-types:	None

Name	StandardSettlementStyleEnum
Content	• Base XSD Type: token • *value* comes from list: {'Standard'|'Net'|'StandardAndNet'}
Documentation	The code specification of whether a trade is settling using standard settlement instructions as well as whether it is a candidate for settlement netting.

Diagram



Schema Component Representation

```
<xsd:simpleType name="StandardSettlementStyleEnum">
   <xsd:restriction base=" xsd:token ">
      <xsd:enumeration value="Standard"/>
      <xsd:enumeration value="Net"/>
      <xsd:enumeration value="StandardAndNet"/>
```

```
    </xsd:restriction>
</xsd:simpleType>
```

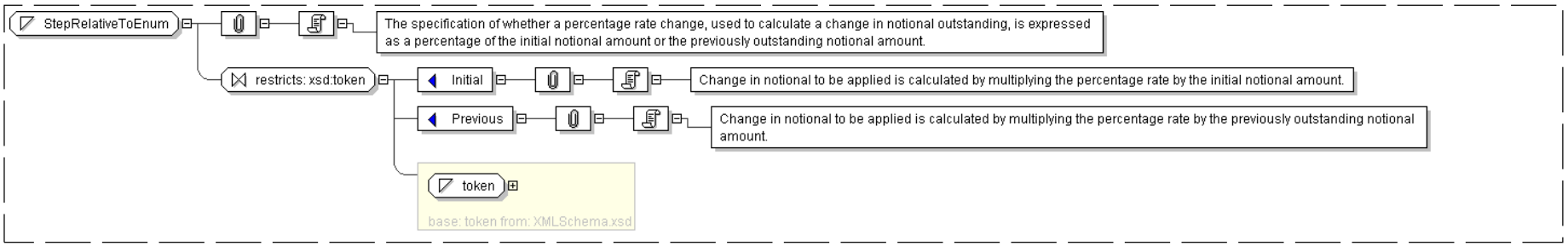
top

Simple Type: StepRelativeToEnum

Super-types:	xsd:token < **StepRelativeToEnum** (by restriction)
Sub-types:	None

Name	StepRelativeToEnum
Content	• Base XSD Type: token • *value* comes from list: {'Initial'|'Previous'}
Documentation	The specification of whether a percentage rate change, used to calculate a change in notional outstanding, is expressed as a percentage of the initial notional amount or the previously outstanding notional amount.

Diagram



Schema Component Representation

```
<xsd:simpleType name="StepRelativeToEnum">
   <xsd:restriction base=" xsd:token ">
      <xsd:enumeration value="Initial"/>
      <xsd:enumeration value="Previous"/>
   </xsd:restriction>
</xsd:simpleType>
```

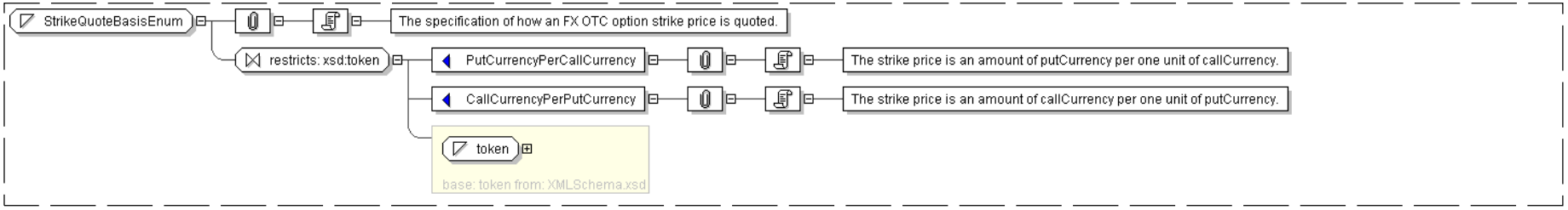
top

Simple Type: StrikeQuoteBasisEnum

Super-types:	xsd:token < **StrikeQuoteBasisEnum** (by restriction)
Sub-types:	None

Name	StrikeQuoteBasisEnum
Content	• Base XSD Type: token • *value* comes from list: {'PutCurrencyPerCallCurrency'|'CallCurrencyPerPutCurrency'}
Documentation	The specification of how an FX OTC option strike price is quoted.

Diagram



Schema Component Representation

```
<xsd:simpleType name="StrikeQuoteBasisEnum">
   <xsd:restriction base=" xsd:token ">
      <xsd:enumeration value="PutCurrencyPerCallCurrency"/>
      <xsd:enumeration value="CallCurrencyPerPutCurrency"/>
   </xsd:restriction>
</xsd:simpleType>
```

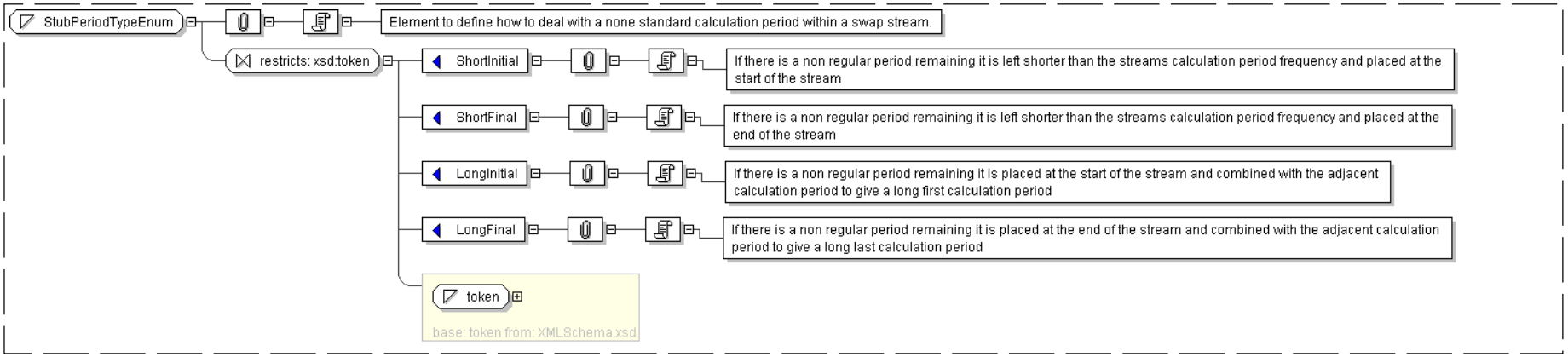
top

Simple Type: StubPeriodTypeEnum

Super-types:	xsd:token < **StubPeriodTypeEnum** (by restriction)
Sub-types:	None

Name	StubPeriodTypeEnum
Content	• Base XSD Type: token • *value* comes from list: {'ShortInitial'|'ShortFinal'|'LongInitial'|'LongFinal'}
Documentation	Element to define how to deal with a none standard calculation period within a swap stream.

Diagram



Schema Component Representation

```
<xsd:simpleType name="StubPeriodTypeEnum">
   <xsd:restriction base=" xsd:token ">
      <xsd:enumeration value="ShortInitial"/>
      <xsd:enumeration value="ShortFinal"/>
      <xsd:enumeration value="LongInitial"/>
      <xsd:enumeration value="LongFinal"/>
   </xsd:restriction>
</xsd:simpleType>
```

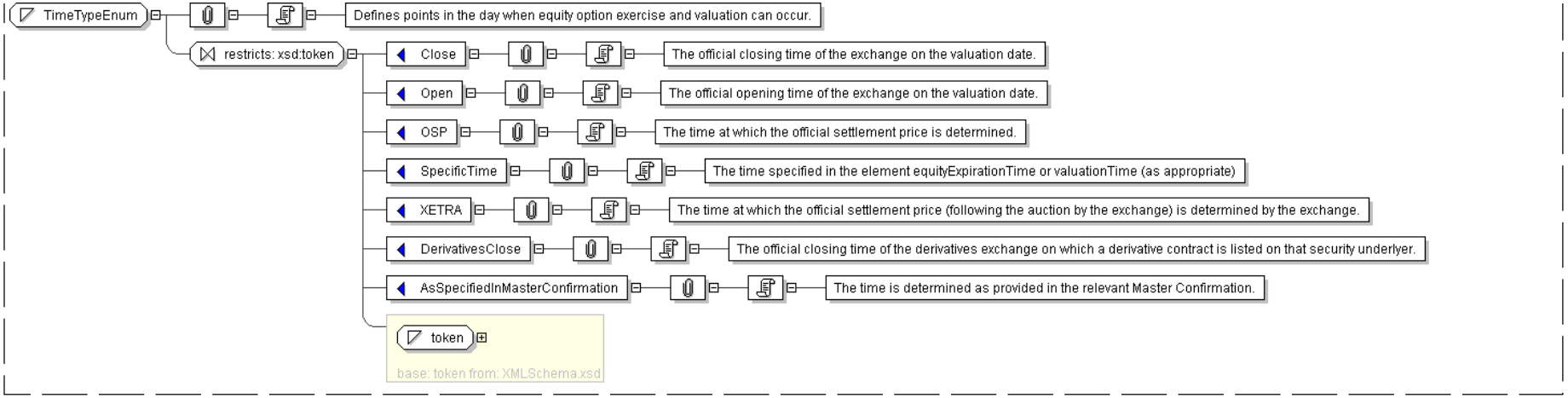
top

Simple Type: TimeTypeEnum

Super-types:	xsd:token < **TimeTypeEnum** (by restriction)
Sub-types:	None

Name	TimeTypeEnum
Content	• Base XSD Type: token • *value* comes from list: {'Close'|'Open'|'OSP'|'SpecificTime'|'XETRA'|'DerivativesClose'|'AsSpecifiedInMasterConfirmation'}
Documentation	Defines points in the day when equity option exercise and valuation can occur.

Diagram



Schema Component Representation

```
<xsd:simpleType name="TimeTypeEnum">
   <xsd:restriction base=" xsd:token ">
      <xsd:enumeration value="Close"/>
      <xsd:enumeration value="Open"/>
      <xsd:enumeration value="OSP"/>
      <xsd:enumeration value="SpecificTime"/>
      <xsd:enumeration value="XETRA"/>
      <xsd:enumeration value="DerivativesClose"/>
      <xsd:enumeration value="AsSpecifiedInMasterConfirmation"/>
   </xsd:restriction>
</xsd:simpleType>
```

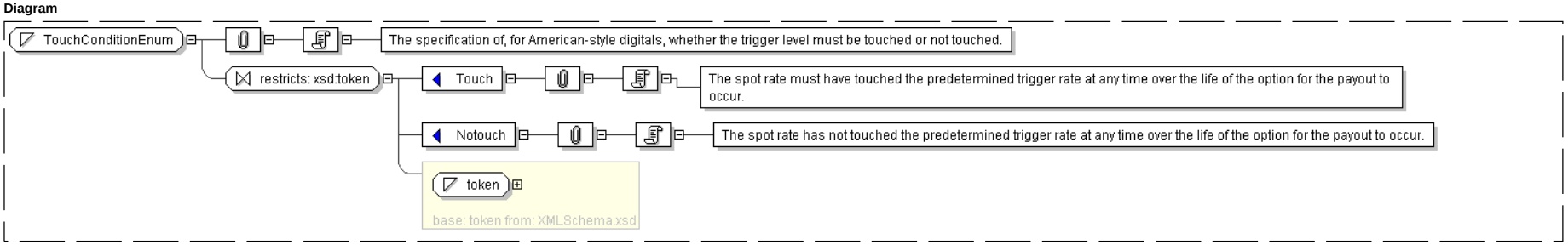
top

Simple Type: TouchConditionEnum

Super-types:	xsd:token < **TouchConditionEnum** (by restriction)
Sub-types:	None

Name	TouchConditionEnum
Content	• Base XSD Type: token • *value* comes from list: {'Touch'|'Notouch'}
Documentation	The specification of, for American-style digitals, whether the trigger level must be touched or not touched.

Diagram



Schema Component Representation

```
<xsd:simpleType name="TouchConditionEnum">
   <xsd:restriction base=" xsd:token ">
      <xsd:enumeration value="Touch"/>
      <xsd:enumeration value="Notouch"/>
   </xsd:restriction>
</xsd:simpleType>
```

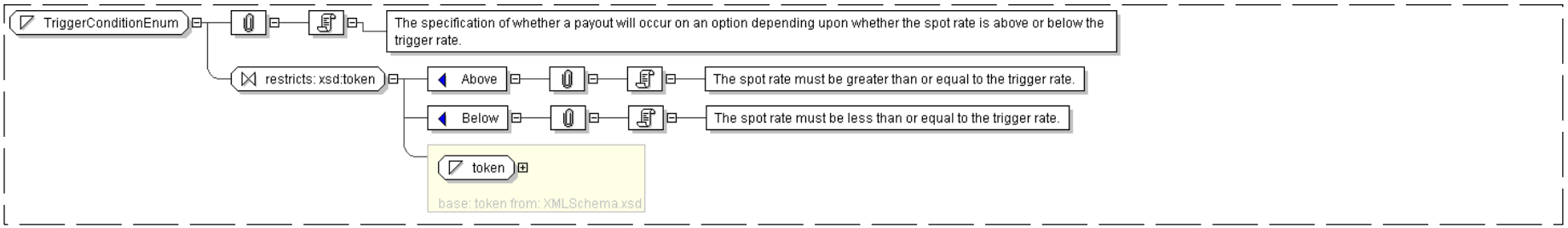
top

Simple Type: TriggerConditionEnum

Super-types:	xsd:token < **TriggerConditionEnum** (by restriction)
Sub-types:	None

Name	TriggerConditionEnum
Content	• Base XSD Type: token • *value* comes from list: {'Above'|'Below'}
Documentation	The specification of whether a payout will occur on an option depending upon whether the spot rate is above or below the trigger rate.

Diagram



Schema Component Representation

```
<xsd:simpleType name="TriggerConditionEnum">
   <xsd:restriction base=" xsd:token ">
      <xsd:enumeration value="Above"/>
      <xsd:enumeration value="Below"/>
   </xsd:restriction>
</xsd:simpleType>
```

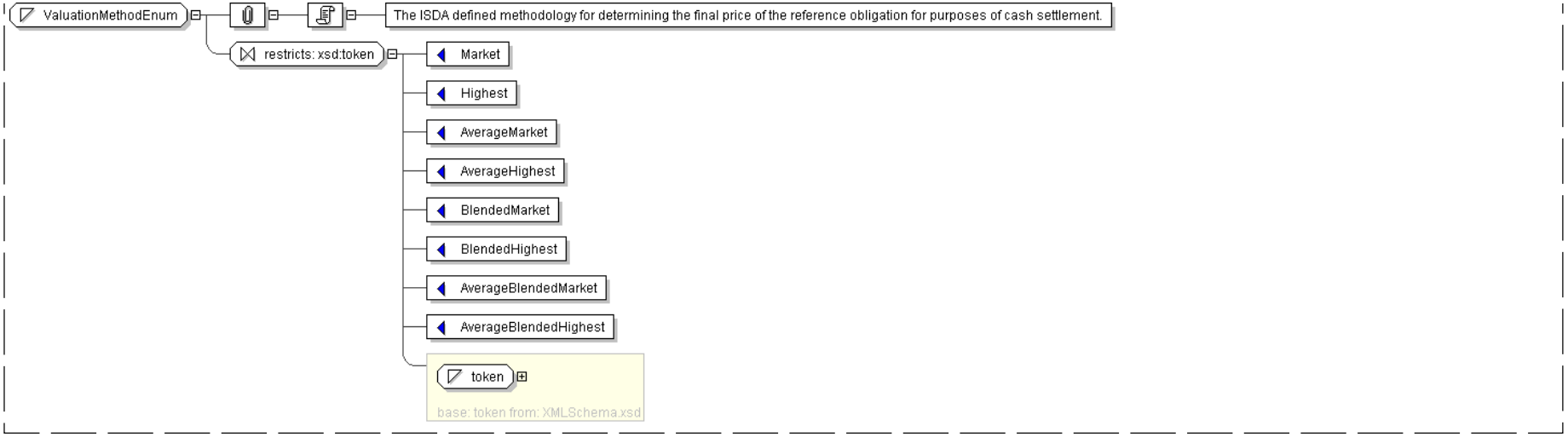
top

Simple Type: ValuationMethodEnum

Super-types:	xsd:token < **ValuationMethodEnum** (by restriction)
Sub-types:	None

Name	ValuationMethodEnum
Content	• Base XSD Type: token • *value* comes from list: {'Market'|'Highest'|'AverageMarket'|'AverageHighest'|'BlendedMarket'|'BlendedHighest'|'AverageBlendedMarket'|'AverageBlendedHighest'}
Documentation	The ISDA defined methodology for determining the final price of the reference obligation for purposes of cash settlement.

Diagram



Schema Component Representation

```
<xsd:simpleType name="ValuationMethodEnum">
   <xsd:restriction base=" xsd:token ">
      <xsd:enumeration value="Market"/>
      <xsd:enumeration value="Highest"/>
      <xsd:enumeration value="AverageMarket"/>
      <xsd:enumeration value="AverageHighest"/>
      <xsd:enumeration value="BlendedMarket"/>
      <xsd:enumeration value="BlendedHighest"/>
      <xsd:enumeration value="AverageBlendedMarket"/>
      <xsd:enumeration value="AverageBlendedHighest"/>
   </xsd:restriction>
</xsd:simpleType>
```

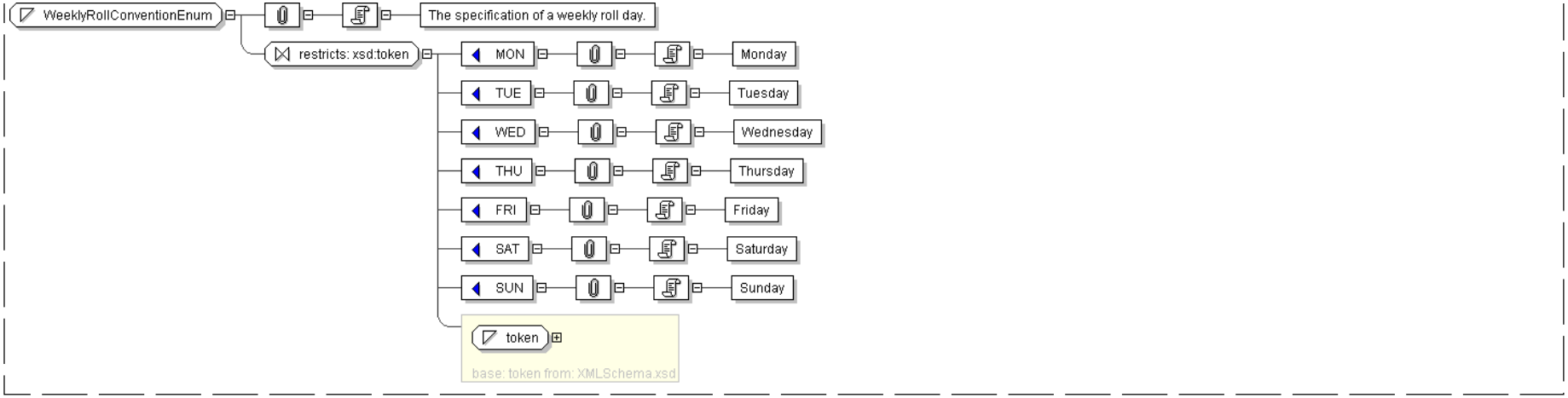
top

Simple Type: WeeklyRollConventionEnum

Super-types:	xsd:token < **WeeklyRollConventionEnum** (by restriction)
Sub-types:	None

Name	WeeklyRollConventionEnum
Content	• Base XSD Type: token • *value* comes from list: {'MON'|'TUE'|'WED'|'THU'|'FRI'|'SAT'|'SUN'}
Documentation	The specification of a weekly roll day.

Diagram



Schema Component Representation

```
<xsd:simpleType name="WeeklyRollConventionEnum">
   <xsd:restriction base=" xsd:token ">
      <xsd:enumeration value="MON"/>
      <xsd:enumeration value="TUE"/>
      <xsd:enumeration value="WED"/>
      <xsd:enumeration value="THU"/>
      <xsd:enumeration value="FRI"/>
      <xsd:enumeration value="SAT"/>
      <xsd:enumeration value="SUN"/>
   </xsd:restriction>
</xsd:simpleType>
```

top

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.