XML Schema Documentation

Table of Contents

- Schema Document Properties
- Global Definitions
  - Complex Type: **AffectedTransactions**
  - Complex Type: **Novation**
  - Complex Type: **NovationNotificationMessage**
  - Complex Type: **NovationRequestMessage**
  - Complex Type: **NovationResponseMessage**
  - Complex Type: **PartialTerminationAmount**
  - Complex Type: **Termination**
  - Complex Type: **TradeAmendment**
  - Model Group: **NovationDetails.model**
  - Model Group: **NovationMessage.model**
  - Model Group: **TerminationDetails.model**
- Legend
- Glossary

top

Schema Document Properties

Target Namespace	**http://www.fpml.org/2007/FpML-4-4**
Version	$Revision: 2350 $
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-msg-4-4.xsd

Declared Namespaces

Prefix	Namespace
Default namespace	**http://www.fpml.org/2007/FpML-4-4**
xml	http://www.w3.org/XML/1998/namespace
ecore	http://www.eclipse.org/emf/2002/Ecore
fpml-annotation	http://www.fpml.org/annotation
fpml	**http://www.fpml.org/2007/FpML-4-4**
dsig	http://www.w3.org/2000/09/xmldsig#
xsd	http://www.w3.org/2001/XMLSchema

Schema Component Representation

```
<xsd:schema nsPrefix="fpml" package="org.fpml" documentRoot="FpML" targetNamespace="http://
www.fpml.org/2007/FpML-4-4" version="$Revision: 2350 $"
attributeFormDefault="unqualified" elementFormDefault="qualified">
   <xsd:include schemaLocation="fpml-msg-4-4.xsd"/>
   ...
</xsd:schema>
```

top

Global Definitions

Complex Type: AffectedTransactions

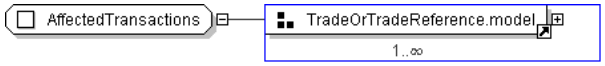
Super-types:	None
Sub-types:	None

Name	AffectedTransactions
Abstract	no

XML Instance Representation

```
<...>
Start Group: TradeOrTradeReference.model [1..*]
Start Choice [1]
        <trade> Trade </trade> [1]
        'An element that allows the full details of the trade to be used as a mechanism for
        identifying the trade for which the post-trade event pertains'

        <tradeReference> PartyTradeIdentifiers </tradeReference> [1]
        'A container since an individual trade can be referenced by two or more
        different partyTradeIdentifier elements - each allocated by a different party.'

End Choice
End Group: TradeOrTradeReference.model
</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="AffectedTransactions">
   <xsd:group ref=" TradeOrTradeReference.model " maxOccurs="unbounded"/>
</xsd:complexType>
```

top

Complex Type: Novation

Super-types:	Event < **Novation** (by extension)
Sub-types:	None

Name	Novation
Used by (from the same schema document)	Model Group **NovationMessage.model**
Abstract	no
Documentation	An event type that records the occurrence of a novation

XML Instance Representation

```
<...>
   <eventId> EventId </eventId> [0..*]
   ''

Start Choice [1]
Start Choice [1]
        <newTransactionReference> PartyTradeIdentifiers </newTransactionReference> [1]
        'Indicates a reference to the new transaction between the transferee and the remaining party.'

        <newTransaction> Trade </newTransaction> [1]
        'Indicates the new transaction between the transferee and the remaining party.'

End Choice
Start Choice [1]
        <oldTransactionReference> PartyTradeIdentifiers </oldTransactionReference> [1]
        'Indicates a reference to the original trade between the transferor and the remaining party.'

        <oldTransaction> Trade </oldTransaction> [1]
        'Indicates the original trade between the transferor and the remaining party.'
```

End Choice
Start Choice [0..1]

<newTransactionReference> PartyTradeIdentifiers </newTransactionReference> [1]

'Indicates a reference to the new transaction between the transferee and the remaining party.'

<newTransaction> Trade </newTransaction> [1]

'Indicates the new transaction between the transferee and the remaining party.'

End Choice
End Choice

<transferor> PartyReference </transferor> [1]

'A pointer style reference to a party identifier defined elsewhere in the document. In a three-way novation the party referenced is the Transferor (outgoing party) in the novation. The Transferor means a party which transfers by novation to a Transferee all of its rights, liabilities, duties and obligations with respect to a Remaining Party. In a four-way novation the party referenced is Transferor 1 which transfers by novation to Transferee 1 all of its rights, liabilities, duties and obligations with respect to Transferor 2. ISDA 2004 Novation Term: Transferor (three-way novation) or Transferor 1 (four-way novation).'

<transferee> PartyReference </transferee> [1]

'A pointer style reference to a party identifier defined elsewhere in the document. In a three-way novation the party referenced is the Transferee (incoming party) in the novation. Transferee means a party which accepts by way of novation all rights, liabilities, duties and obligations of a Transferor with respect to a Remaining Party. In a four-way novation the party referenced is Transferee 1 which accepts by way of novation the rights, liabilities, duties and obligations of Transferor 1. ISDA 2004 Novation Term: Transferee (three-way novation) or Transferee 1 (four-way novation).'

<remainingParty> PartyReference </remainingParty> [1]

'A pointer style reference to a party identifier defined elsewhere in the document. In a three-way novation the party referenced is the Remaining Party in the novation. Remaining Party means a party which consents to a Transferor\'s transfer by novation and the acceptance thereof by the Transferee of all of the Transferor\'s rights, liabilities, duties and obligations with respect to such Remaining Party under and with respect of the Novated Amount of a transaction. In a four-way novation the party referenced is Transferor 2 per the ISDA definition and acts in the role of a Transferor. Transferor 2 transfers by novation to Transferee 2 all of its rights, liabilities, duties and obligations with respect to Transferor 1. ISDA 2004 Novation Term: Remaining Party (three-way novation) or Transferor 2 (four-way novation).'

<otherRemainingParty> PartyReference </otherRemainingParty> [0..1]

'A pointer style reference to a party identifier defined elsewhere in the document. This element is not applicable in a three-way novation and should be omitted. In a four-way novation the party referenced is Transferee 2. Transferee 2 means a party which accepts by way of novation the rights, liabilities, duties and obligations of Transferor 2. ISDA 2004 Novation Term: Transferee 2 (four-way novation).'

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

'Specifies the date that one party\'s legal obligations with regard to a trade are transferred to another party. It corresponds to the Novation Date section of the 2004 ISDA Novation Definitions, section 1.16.'

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

'Specifies the date the parties agree to assign or novate a trade. If this element is not specified, the novationTradeDate will be deemed to be the novationDate. It corresponds to the Novation Trade Date section of the 2004 ISDA Novation Definitions, section 1.17.'

Start Choice [0..1]

<novatedAmount> Money </novatedAmount> [1]

'The amount which represents the portion of the Old Transaction being novated.'

<novatedNumberOfOptions> xsd:decimal </novatedNumberOfOptions> [1]

'The number of options which represent the portion of the Old Transaction being novated.'

End Choice

<remainingTrade> Trade </remainingTrade> [0..1]

'This element contains a description of the remaining portion of a partially novated trade.'

<fullFirstCalculationPeriod> xsd:boolean </fullFirstCalculationPeriod> [0..1]

'This element corresponds to the applicability of the Full First Calculation Period as defined in the 2004 ISDA Novation Definitions, section 1.20.'

<firstPeriodStartDate> FirstPeriodStartDate **</firstPeriodStartDate>** [0..2]

'Element that is used to be able to make sense of the "new transaction" without requiring reference back to the "old transaction". In the case of interest rate products there are potentially 2 "first period start dates" to reference – one with respect to each party to the new transaction. For Credit Default Swaps there is just the one with respect to the party that is the fixed rate payer.'

<nonReliance> Empty **</nonReliance>** [0..1]

'This element corresponds to the non-Reliance section in the 2004 ISDA Novation Definitions, section 2.1 (c) (i). The element appears in the instance document when non-Reliance is applicable.'

<creditDerivativesNotices> CreditDerivativesNotices **</creditDerivativesNotices>** [0..1]

'This element should be specified if one or more of either a Credit Event Notice, Notice of Publicly Available Information, Notice of Physical Settlement or Notice of Intended Physical Settlement, as applicable, has been delivered by or to the Transferor or the Remaining Party. The type of notice or notices that have been delivered should be indicated by setting the relevant boolean element value(s) to true. The absence of the element means that no Credit Event Notice, Notice of Publicly Available Information, Notice of Physical Settlement or Notice of Intended Physical Settlement, as applicable, has been delivered by or to the Transferor or the Remaining Party.'

<contractualDefinitions> ContractualDefinitions **</contractualDefinitions>** [0..*]

'The definitions (such as those published by ISDA) that will define the terms of the novation transaction.'

Start Choice [1]

<contractualSupplement> ContractualSupplement **</contractualSupplement>** [0..*]

'DEPRECATED - This element will be removed in the next major version of FpML. The element contractualTermsSupplement should be used instead. Definition: A contractual supplement (such as those published by ISDA) that will apply to the trade.'

<contractualTermsSupplement> ContractualTermsSupplement **</contractualTermsSupplement>** [0..*]

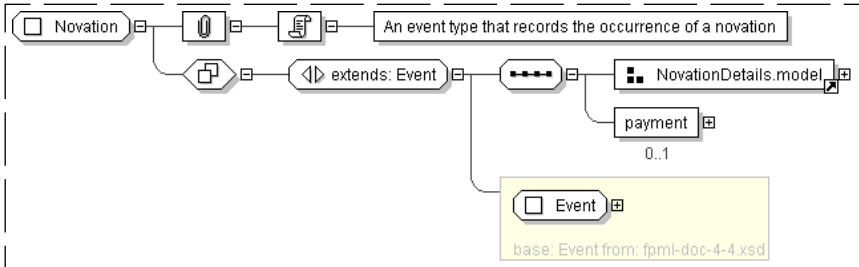
'A contractual supplement (such as those published by ISDA) that will apply to the trade.'

End Choice

<payment> Payment **</payment>** [0..1]

</...>

Diagram



Schema Component Representation

```
<xsd:complexType name="Novation">
   <xsd:complexContent>
      <xsd:extension base=" Event ">
         <xsd:sequence>
            <xsd:group ref=" NovationDetails.model "/>
            <xsd:element name="payment" type=" Payment " minOccurs="0"/>
         </xsd:sequence>
      </xsd:extension>
   </xsd:complexContent>
</xsd:complexType>
```

top

Complex Type: NovationNotificationMessage

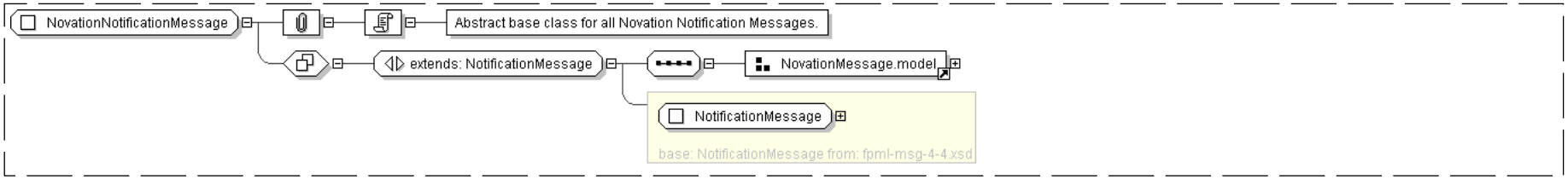
Super-types:	NotificationMessage < **NovationNotificationMessage** (by extension)
Sub-types:	None

Name	NovationNotificationMessage
Abstract	yes
Documentation	Abstract base class for all Novation Notification Messages.

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..*]
   <novation> Novation </novation> [1]
   <party> Party </party> [3..*]
</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="NovationNotificationMessage" abstract="true">
   <xsd:complexContent>
      <xsd:extension base=" NotificationMessage ">
         <xsd:sequence>
            <xsd:group ref=" NovationMessage.model "/>
         </xsd:sequence>
      </xsd:extension>
   </xsd:complexContent>
</xsd:complexType>
```

top

Complex Type: NovationRequestMessage

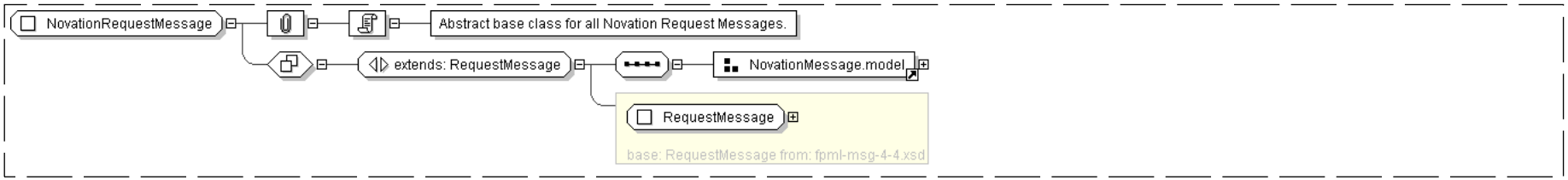
Super-types:	RequestMessage < **NovationRequestMessage** (by extension)
Sub-types:	None

Name	NovationRequestMessage
Abstract	yes
Documentation	Abstract base class for all Novation Request Messages.

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..*]
   <novation> Novation </novation> [1]
   <party> Party </party> [3..*]
</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="NovationRequestMessage" abstract="true">
   <xsd:complexContent>
      <xsd:extension base=" RequestMessage ">
         <xsd:sequence>
            <xsd:group ref=" NovationMessage.model "/>
         </xsd:sequence>
      </xsd:extension>
   </xsd:complexContent>
</xsd:complexType>
```

top

Complex Type: NovationResponseMessage

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

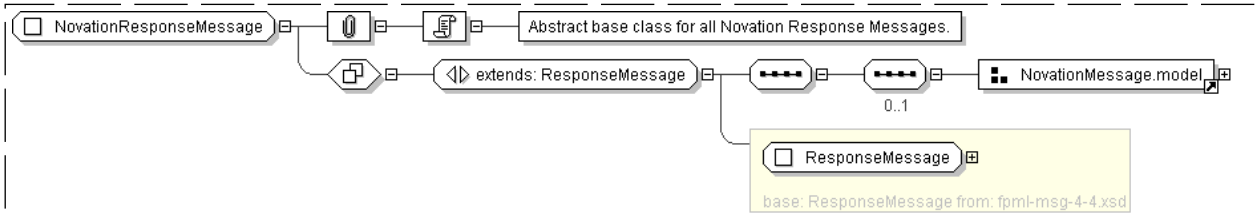
Name	NovationResponseMessage
Abstract	yes

Documentation	Abstract base class for all Novation Response Messages.

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..*]
Start Sequence [0..1]
      <novation> Novation </novation> [1]
      <party> Party </party> [3..*]
End Sequence
</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="NovationResponseMessage" abstract="true">
   <xsd:complexContent>
      <xsd:extension base=" ResponseMessage ">
         <xsd:sequence>
            <xsd:sequence minOccurs="0">
               <xsd:group ref=" NovationMessage.model "/>
            </xsd:sequence>
         </xsd:sequence>
      </xsd:extension>
   </xsd:complexContent>
</xsd:complexType>
```

top

Complex Type: PartialTerminationAmount

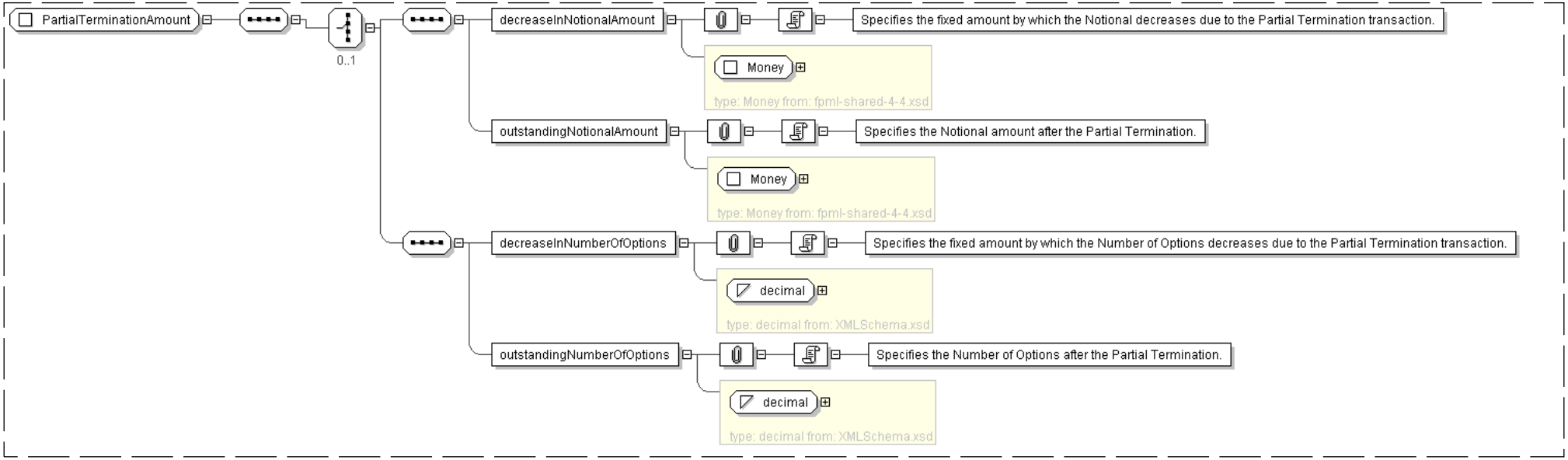
Super-types:	None
Sub-types:	None

Name	PartialTerminationAmount
Used by (from the same schema document)	Model Group **TerminationDetails.model**
Abstract	no

XML Instance Representation

```
<...>
Start Choice [0..1]
      <decreaseInNotionalAmount> Money </decreaseInNotionalAmount> [1]
      'Specifies the fixed amount by which the Notional decreases due to the Partial
      Termination transaction.'

      <outstandingNotionalAmount> Money </outstandingNotionalAmount> [1]
      'Specifies the Notional amount after the Partial Termination.'

      <decreaseInNumberOfOptions> xsd:decimal </decreaseInNumberOfOptions> [1]
      'Specifies the fixed amount by which the Number of Options decreases due to the
      Partial Termination transaction.'

      <outstandingNumberOfOptions> xsd:decimal </outstandingNumberOfOptions> [1]
      'Specifies the Number of Options after the Partial Termination.'

End Choice
</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="PartialTerminationAmount">
   <xsd:sequence>
      <xsd:choice minOccurs="0">
         <xsd:sequence>
            <xsd:element name="decreaseInNotionalAmount" type=" Money "/>
            <xsd:element name="outstandingNotionalAmount" type=" Money "/>
         </xsd:sequence>
         <xsd:sequence>
            <xsd:element name="decreaseInNumberOfOptions" type=" xsd:decimal "/>
            <xsd:element name="outstandingNumberOfOptions" type=" xsd:decimal "/>
         </xsd:sequence>
      </xsd:choice>
   </xsd:sequence>
</xsd:complexType>
```

top

Complex Type: Termination

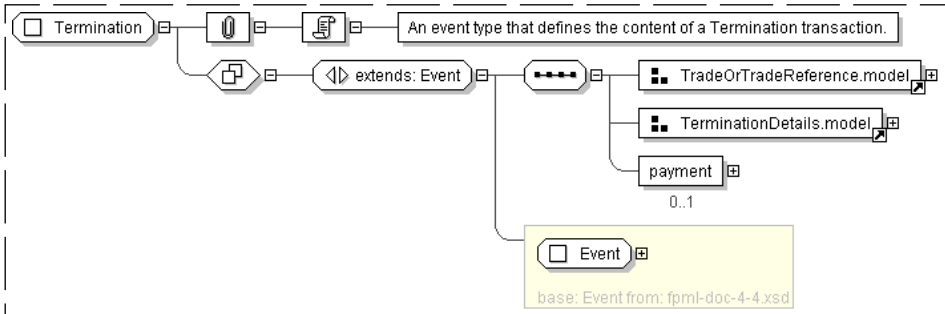
Super-types:	Event < **Termination** (by extension)
Sub-types:	None

Name	Termination
Abstract	no
Documentation	An event type that defines the content of a Termination transaction.

XML Instance Representation

```
<...>
   <eventId> EventId </eventId> [0..*]
   ''

Start Choice [1]
      <trade> Trade </trade> [1]
      'An element that allows the full details of the trade to be used as a mechanism for
      identifying the trade for which the post-trade event pertains'

      <tradeReference> PartyTradeIdentifiers </tradeReference> [1]
      'A container since an individual trade can be referenced by two or more
      different partyTradeIdentifier elements - each allocated by a different party.'

End Choice
   <terminationTradeDate> xsd:date </terminationTradeDate> [1]
   'The date on which the the parties enter into the Termination transaction.'

   <terminationEffectiveDate> xsd:date </terminationEffectiveDate> [1]
   'The date on which the Termination becomes effective.'

Start Choice [1]
      <full> Empty </full> [1]
      'The use of the Full element indicates that this is a Full Termination.'

      <partial> PartialTerminationAmount </partial> [1]
      'The use of the Partial element indicates that this is a Partial Termination.'

End Choice
   <payment> Payment </payment> [0..1]
   'A payment for the right to terminate the trade.'

</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="Termination">
   <xsd:complexContent>
      <xsd:extension base=" Event ">
         <xsd:sequence>
```

```
            <xsd:group ref=" TradeOrTradeReference.model "/>
            <xsd:group ref=" TerminationDetails.model "/>
            <xsd:element name="payment" type=" Payment " minOccurs="0"/>
         </xsd:sequence>
      </xsd:extension>
   </xsd:complexContent>
</xsd:complexType>
```

top

Complex Type: TradeAmendment

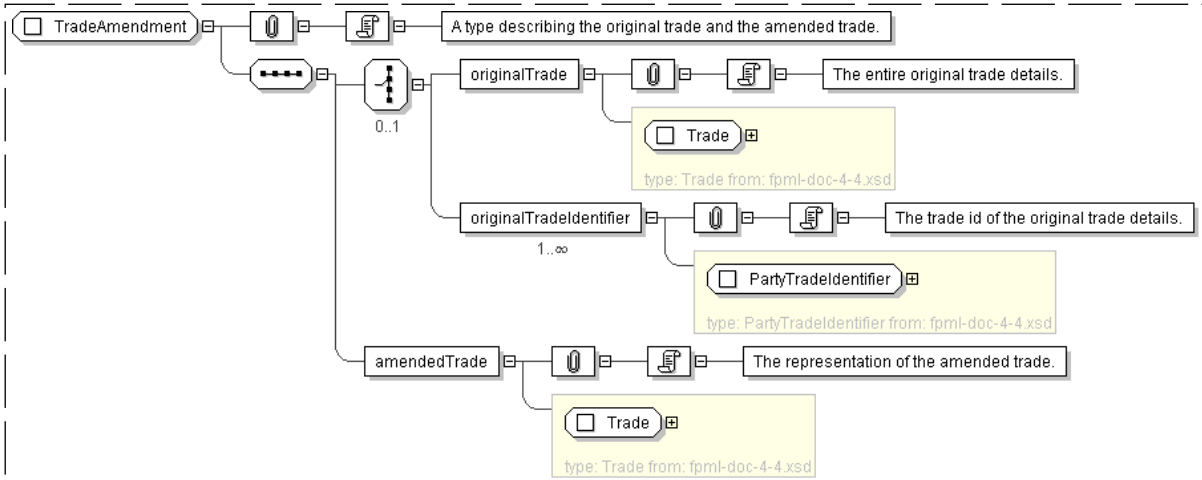
Super-types:	None
Sub-types:	None

Name	TradeAmendment
Abstract	no
Documentation	A type describing the original trade and the amended trade.

XML Instance Representation

```
<...>
Start Choice [0..1]
      <originalTrade> Trade </originalTrade> [1]
      'The entire original trade details.'

      <originalTradeIdentifier> PartyTradeIdentifier </originalTradeIdentifier> [1..*]
      'The trade id of the original trade details.'

End Choice
   <amendedTrade> Trade </amendedTrade> [1]
   'The representation of the amended trade.'

</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="TradeAmendment">
   <xsd:sequence>
      <xsd:choice minOccurs="0">
         <xsd:element name="originalTrade" type=" Trade "/>
         <xsd:element name="originalTradeIdentifier" type=" PartyTradeIdentifier "
         maxOccurs="unbounded"/>
```

```
      </xsd:choice>
      <xsd:element name="amendedTrade" type=" Trade "/>
   </xsd:sequence>
</xsd:complexType>
```

top

Model Group: NovationDetails.model

Name	NovationDetails.model
Used by (from the same schema document)	Complex Type **Novation**

XML Instance Representation

```
Start Choice [1]
Start Choice [1]
      <newTransactionReference> PartyTradeIdentifiers </newTransactionReference> [1]
      'Indicates a reference to the new transaction between the transferee and the remaining party.'

      <newTransaction> Trade </newTransaction> [1]
      'Indicates the new transaction between the transferee and the remaining party.'

End Choice
Start Choice [1]
      <oldTransactionReference> PartyTradeIdentifiers </oldTransactionReference> [1]
      'Indicates a reference to the original trade between the transferor and the remaining party.'

      <oldTransaction> Trade </oldTransaction> [1]
      'Indicates the original trade between the transferor and the remaining party.'

End Choice
Start Choice [0..1]
      <newTransactionReference> PartyTradeIdentifiers </newTransactionReference> [1]
      'Indicates a reference to the new transaction between the transferee and the remaining party.'

      <newTransaction> Trade </newTransaction> [1]
      'Indicates the new transaction between the transferee and the remaining party.'

End Choice
End Choice
<transferor> PartyReference </transferor> [1]
'A pointer style reference to a party identifier defined elsewhere in the document. In a
three-way novation the party referenced is the Transferor (outgoing party) in the novation.
The Transferor means a party which transfers by novation to a Transferee all of its
rights, liabilities, duties and obligations with respect to a Remaining Party. In a four-
way novation the party referenced is Transferor 1 which transfers by novation to Transferee
1 all of its rights, liabilities, duties and obligations with respect to Transferor 2.
ISDA 2004 Novation Term: Transferor (three-way novation) or Transferor 1 (four-way novation).'

<transferee> PartyReference </transferee> [1]
'A pointer style reference to a party identifier defined elsewhere in the document. In a
three-way novation the party referenced is the Transferee (incoming party) in the
novation. Transferee means a party which accepts by way of novation all rights,
liabilities, duties and obligations of a Transferor with respect to a Remaining Party. In
a four-way novation the party referenced is Transferee 1 which accepts by way of novation
the rights, liabilities, duties and obligations of Transferor 1. ISDA 2004 Novation
Term: Transferee (three-way novation) or Transferee 1 (four-way novation).'

<remainingParty> PartyReference </remainingParty> [1]
'A pointer style reference to a party identifier defined elsewhere in the document. In a
three-way novation the party referenced is the Remaining Party in the novation. Remaining
Party means a party which consents to a Transferor\'s transfer by novation and the
acceptance thereof by the Transferee of all of the Transferor\'s rights, liabilities,
duties and obligations with respect to such Remaining Party under and with respect of
the Novated Amount of a transaction. In a four-way novation the party referenced is
Transferor 2 per the ISDA definition and acts in the role of a Transferor. Transferor
2 transfers by novation to Transferee 2 all of its rights, liabilities, duties and
```

obligations with respect to Transferor 1. ISDA 2004 Novation Term: Remaining Party (three-way novation) or Transferor 2 (four-way novation).'

<otherRemainingParty> PartyReference </otherRemainingParty> [0..1]

'A pointer style reference to a party identifier defined elsewhere in the document. This element is not applicable in a three-way novation and should be omitted. In a four-way novation the party referenced is Transferee 2. Transferee 2 means a party which accepts by way of novation the rights, liabilities, duties and obligations of Transferor 2. ISDA 2004 Novation Term: Transferee 2 (four-way novation).'

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

'Specifies the date that one party\'s legal obligations with regard to a trade are transferred to another party. It corresponds to the Novation Date section of the 2004 ISDA Novation Definitions, section 1.16.'

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

'Specifies the date the parties agree to assign or novate a trade. If this element is not specified, the novationTradeDate will be deemed to be the novationDate. It corresponds to the Novation Trade Date section of the 2004 ISDA Novation Definitions, section 1.17.'

Start Choice [0..1]

<novatedAmount> Money </novatedAmount> [1]

'The amount which represents the portion of the Old Transaction being novated.'

<novatedNumberOfOptions> xsd:decimal </novatedNumberOfOptions> [1]

'The number of options which represent the portion of the Old Transaction being novated.'

End Choice

<remainingTrade> Trade </remainingTrade> [0..1]

'This element contains a description of the remaining portion of a partially novated trade.'

<fullFirstCalculationPeriod> xsd:boolean </fullFirstCalculationPeriod> [0..1]

'This element corresponds to the applicability of the Full First Calculation Period as defined in the 2004 ISDA Novation Definitions, section 1.20.'

<firstPeriodStartDate> FirstPeriodStartDate </firstPeriodStartDate> [0..2]

'Element that is used to be able to make sense of the "new transaction" without requiring reference back to the "old transaction". In the case of interest rate products there are potentially 2 "first period start dates" to reference – one with respect to each party to the new transaction. For Credit Default Swaps there is just the one with respect to the party that is the fixed rate payer.'

<nonReliance> Empty </nonReliance> [0..1]

'This element corresponds to the non-Reliance section in the 2004 ISDA Novation Definitions, section 2.1 (c) (i). The element appears in the instance document when non-Reliance is applicable.'

<creditDerivativesNotices> CreditDerivativesNotices </creditDerivativesNotices> [0..1]

'This element should be specified if one or more of either a Credit Event Notice, Notice of Publicly Available Information, Notice of Physical Settlement or Notice of Intended Physical Settlement, as applicable, has been delivered by or to the Transferor or the Remaining Party. The type of notice or notices that have been delivered should be indicated by setting the relevant boolean element value(s) to true. The absence of the element means that no Credit Event Notice, Notice of Publicly Available Information, Notice of Physical Settlement or Notice of Intended Physical Settlement, as applicable, has been delivered by or to the Transferor or the Remaining Party.'

<contractualDefinitions> ContractualDefinitions </contractualDefinitions> [0..*]

'The definitions (such as those published by ISDA) that will define the terms of the novation transaction.'

Start Choice [1]

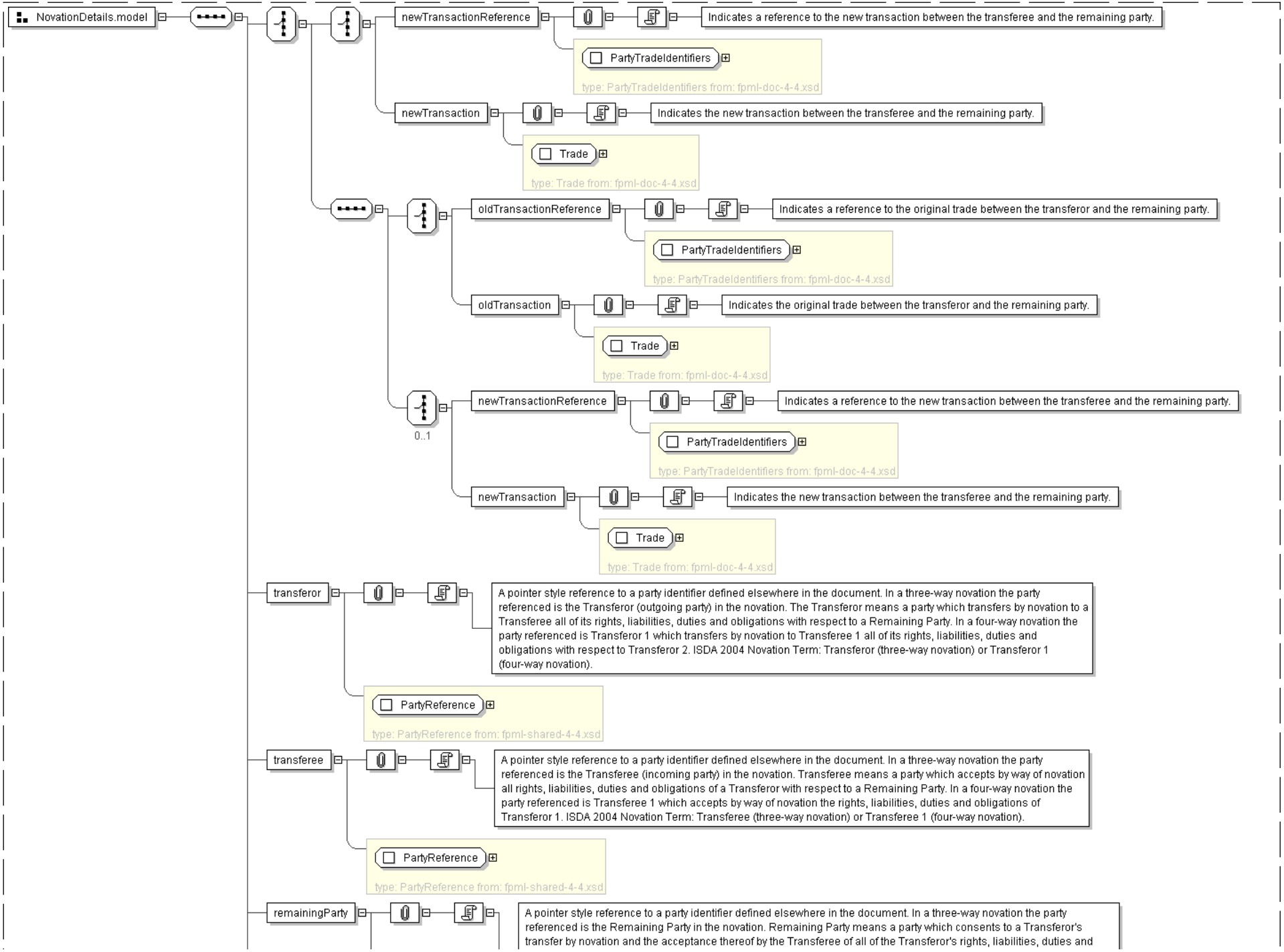
<contractualSupplement> ContractualSupplement </contractualSupplement> [0..*]

'DEPRECATED - This element will be removed in the next major version of FpML. The element contractualTermsSupplement should be used instead. Definition: A contractual supplement (such as those published by ISDA) that will apply to the trade.'

<contractualTermsSupplement> ContractualTermsSupplement </contractualTermsSupplement> [0..*]

```
'A contractual supplement (such as those published by ISDA) that will apply to the trade.'
End Choice
```

Diagram



obligations with respect to such Remaining Party under and with respect of the Novated Amount of a transaction. In a four-way novation the party referenced is Transferor 2 per the ISDA definition and acts in the role of a Transferor. Transferor 2 transfers by novation to Transferee 2 all of its rights, liabilities, duties and obligations with respect to Transferor 1. ISDA 2004 Novation Term: Remaining Party (three-way novation) or Transferor 2 (four-way novation).

PartyReference
type: PartyReference from: fpml-shared-4-4.xsd

otherRemainingParty
0..1

A pointer style reference to a party identifier defined elsewhere in the document. This element is not applicable in a three-way novation and should be omitted. In a four-way novation the party referenced is Transferee 2. Transferee 2 means a party which accepts by way of novation the rights, liabilities, duties and obligations of Transferor 2. ISDA 2004 Novation Term: Transferee 2 (four-way novation).

PartyReference
type: PartyReference from: fpml-shared-4-4.xsd

novationDate

Specifies the date that one party's legal obligations with regard to a trade are transferred to another party. It corresponds to the Novation Date section of the 2004 ISDA Novation Definitions, section 1.16.

date
type: date from: XMLSchema.xsd

novationTradeDate
0..1

Specifies the date the parties agree to assign or novate a trade. If this element is not specified, the novationTradeDate will be deemed to be the novationDate. It corresponds to the Novation Trade Date section of the 2004 ISDA Novation Definitions, section 1.17.

date
type: date from: XMLSchema.xsd

0..1

novatedAmount

The amount which represents the portion of the Old Transaction being novated.

Money
type: Money from: fpml-shared-4-4.xsd

novatedNumberOfOptions

The number of options which represent the portion of the Old Transaction being novated.

decimal
type: decimal from: XMLSchema.xsd

remainingTrade
0..1

This element contains a description of the remaining portion of a partially novated trade.

Trade
type: Trade from: fpml-doc-4-4.xsd

fullFirstCalculationPeriod
0..1

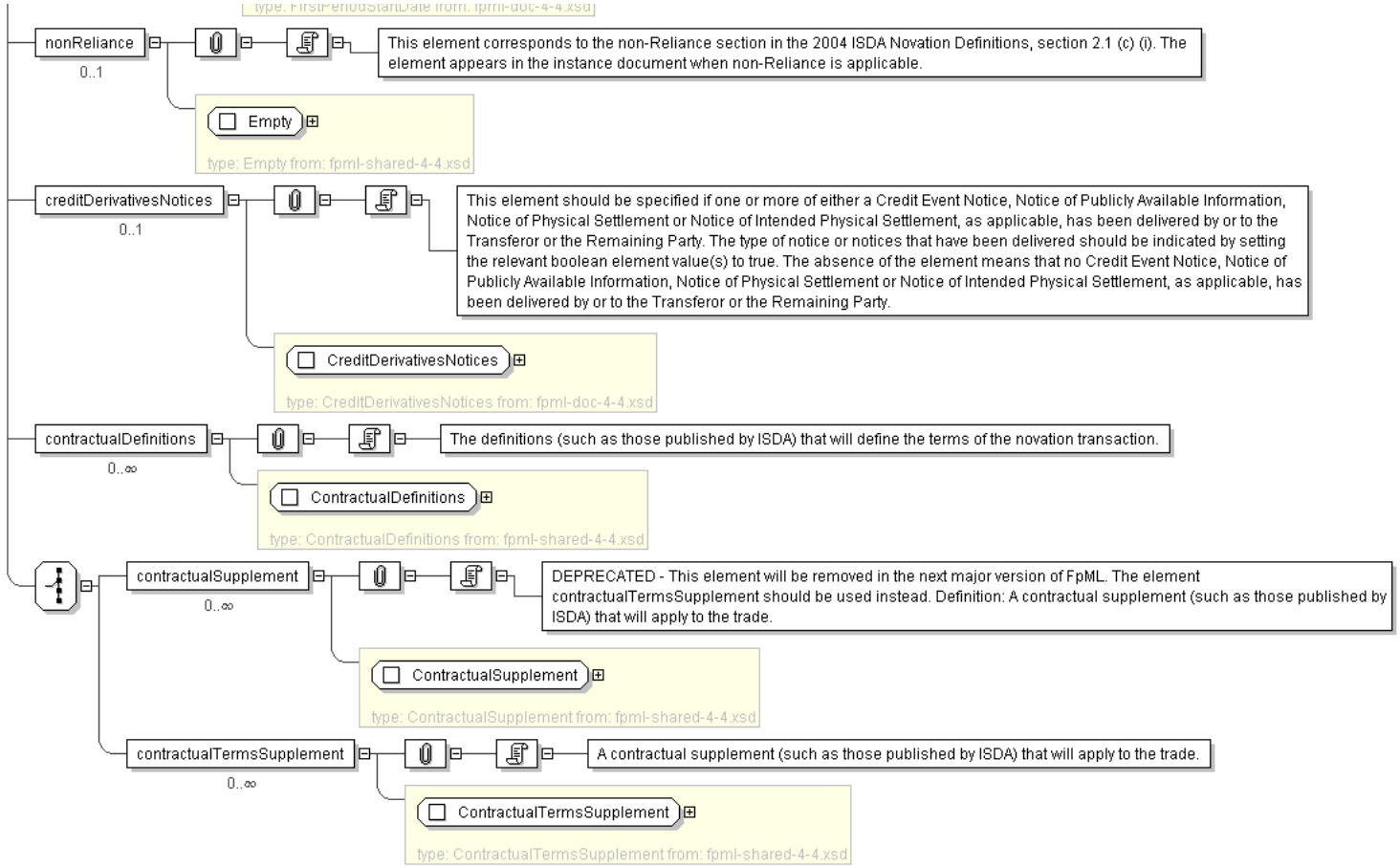
This element corresponds to the applicability of the Full First Calculation Period as defined in the 2004 ISDA Novation Definitions, section 1.20.

boolean
type: boolean from: XMLSchema.xsd

firstPeriodStartDate
0..2

Element that is used to be able to make sense of the "new transaction" without requiring reference back to the "old transaction". In the case of interest rate products there are potentially 2 "first period start dates" to reference - one with respect to each party to the new transaction. For Credit Default Swaps there is just the one with respect to the party that is the fixed rate payer.

FirstPeriodStartDate



Schema Component Representation

```
<xsd:group name="NovationDetails.model">
   <xsd:sequence>
      <xsd:choice>
         <xsd:choice>
            <xsd:element name="newTransactionReference" type=" PartyTradeIdentifiers "/>
            <xsd:element name="newTransaction" type=" Trade "/>
         </xsd:choice>
         <xsd:sequence>
            <xsd:choice>
               <xsd:element name="oldTransactionReference" type=" PartyTradeIdentifiers "/>
               <xsd:element name="oldTransaction" type=" Trade "/>
            </xsd:choice>
            <xsd:choice minOccurs="0">
               <xsd:element name="newTransactionReference" type=" PartyTradeIdentifiers "/>
               <xsd:element name="newTransaction" type=" Trade "/>
            </xsd:choice>
         </xsd:sequence>
      </xsd:choice>
      <xsd:element name="transferor" type=" PartyReference "/>
      <xsd:element name="transferee" type=" PartyReference "/>
      <xsd:element name="remainingParty" type=" PartyReference "/>
      <xsd:element name="otherRemainingParty" type=" PartyReference " minOccurs="0"/>
      <xsd:element name="novationDate" type=" xsd:date "/>
      <xsd:element name="novationTradeDate" type=" xsd:date " minOccurs="0"/>
      <xsd:choice minOccurs="0">
         <xsd:element name="novatedAmount" type=" Money "/>
```

```
      <xsd:element name="novatedNumberOfOptions" type=" xsd:decimal "/>
   </xsd:choice>
   <xsd:element name="remainingTrade" type=" Trade " minOccurs="0"/>
   <xsd:element name="fullFirstCalculationPeriod" type=" xsd:boolean " minOccurs="0"/>
   <xsd:element name="firstPeriodStartDate" type=" FirstPeriodStartDate "
   minOccurs="0" maxOccurs="2"/>
   <xsd:element name="nonReliance" type=" Empty " minOccurs="0"/>
   <xsd:element name="creditDerivativesNotices" type=" CreditDerivativesNotices " minOccurs="0"/>
   <xsd:element name="contractualDefinitions" type=" ContractualDefinitions "
   minOccurs="0" maxOccurs="unbounded"/>
   <xsd:choice>
      <xsd:element name="contractualSupplement" type=" ContractualSupplement "
      minOccurs="0" maxOccurs="unbounded" deprecated="true"
      deprecatedReason="The contractualTermsSupplement includes the publication date, which was
      not present in the contractualSupplement"/>
      <xsd:element name="contractualTermsSupplement" type=" ContractualTermsSupplement
      " minOccurs="0" maxOccurs="unbounded"/>
   </xsd:choice>
  </xsd:sequence>
</xsd:group>
```

top

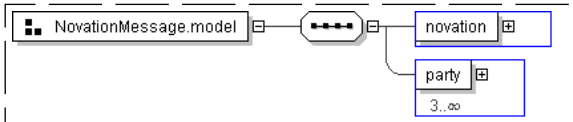
Model Group: NovationMessage.model

Name	NovationMessage.model
Used by (from the same schema document)	Complex Type **NovationNotificationMessage** , Complex Type **NovationRequestMessage** , Complex Type **NovationResponseMessage**

XML Instance Representation

```
<novation> Novation </novation> [1]
<party> Party </party> [3..*]
```

Diagram



Schema Component Representation

```
<xsd:group name="NovationMessage.model">
  <xsd:sequence>
    <xsd:element name="novation" type=" Novation "/>
    <xsd:element name="party" type=" Party " minOccurs="3" maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:group>
```

top

Model Group: TerminationDetails.model

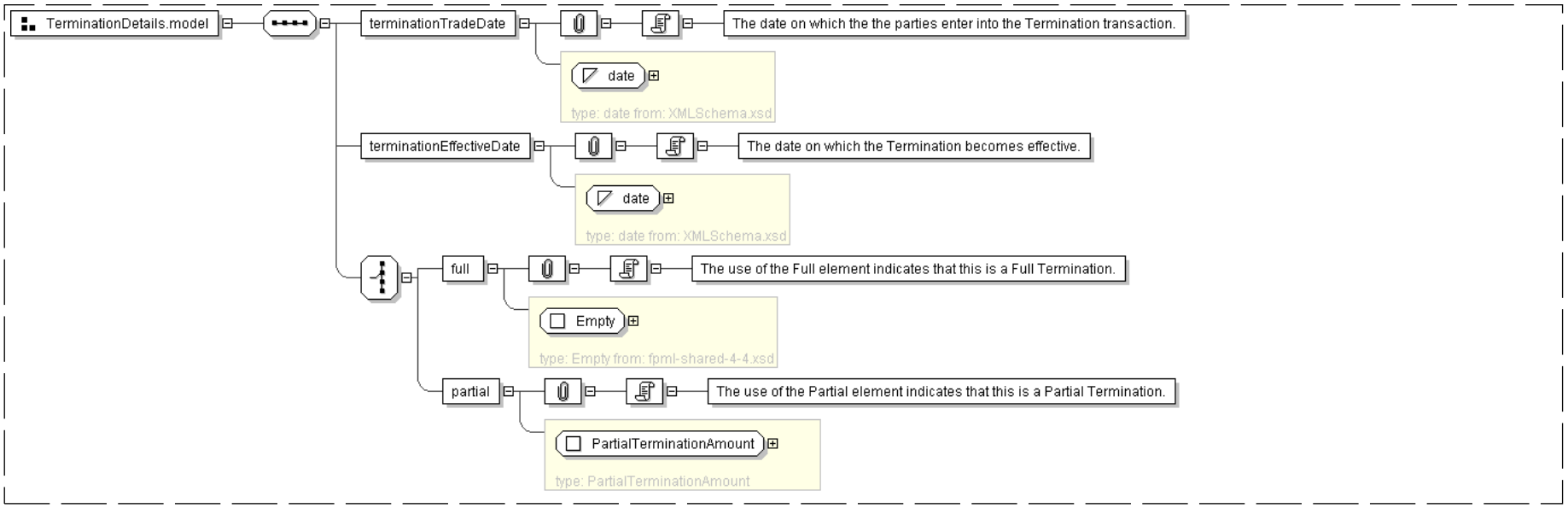
Name	TerminationDetails.model
Used by (from the same schema document)	Complex Type **Termination**

XML Instance Representation

```
<terminationTradeDate> xsd:date </terminationTradeDate> [1]
'The date on which the the parties enter into the Termination transaction.'

<terminationEffectiveDate> xsd:date </terminationEffectiveDate> [1]
'The date on which the Termination becomes effective.'
```

```
Start Choice [1]
   <full> Empty </full> [1]
   'The use of the Full element indicates that this is a Full Termination.'

   <partial> PartialTerminationAmount </partial> [1]
   'The use of the Partial element indicates that this is a Partial Termination.'

End Choice
```

Diagram



Schema Component Representation

```xsd:group name="TerminationDetails.model">
   <xsd:sequence>
      <xsd:element name="terminationTradeDate" type=" xsd:date "/>
      <xsd:element name="terminationEffectiveDate" type=" xsd:date "/>
      <xsd:choice>
         <xsd:element name="full" type=" Empty "/>
         <xsd:element name="partial" type=" PartialTerminationAmount "/>
      </xsd:choice>
   </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.