



**Financial products Markup Language**

## **FpML - Messaging Component Definitions**

## ***Version: 4.3***

### **This Version:**

<http://www.fpml.org/spec/2007/lcwg-fpml-4-3-2007-10-30>

### **Latest Version:**

<http://www.fpml.org/spec/2007/lcwg-fpml-4-3-2007-10-30>

### **Previous Version:**

<http://www.fpml.org/spec/2007/wd-fpml-4-4-2007-09-10/>

### **Errata For This Version:**

<http://www.fpml.org/spec/errata/lcwg-fpml-4-3-2007-10-30-errata.html>

### **Document built**

Copyright (c) 1999 - 2007 by International Swaps and Derivatives Association, Inc.

Financial Products Markup Language is subject to the FpML® Public License.

FpML® is a registered trademark of the International Swaps and Derivatives Association, Inc.

A copy of this license is available at <http://www.fpml.org/license/license.html>

The FpML specifications provided are without warranty of any kind, either expressed or implied, including, without limitation, warranties that FpML, or the FpML specifications are free of defects, merchantable, fit for a particular purpose or non-infringing. The entire risk as to the quality and performance of the specifications is with you. Should any of the FpML specifications prove defective in any respect, you assume the cost of any necessary servicing or repair. Under no circumstances and under no legal theory, whether tort (including negligence), contract, or otherwise, shall ISDA, any of its members, or any distributor of documents or software containing any of the FpML specifications, or any supplier of any of such parties, be liable to you or any other person for any indirect, special, incidental, or consequential damages of any character including, without limitation, damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses, even if such party shall have been informed of the possibility of such damages.

# Table Of Contents

1	Global Complex Types	8
1.1	AdditionalData	9
1.1.1	Description:	9
1.1.2	Contents:	9
1.1.3	Used by:	9
1.1.4	Derived Types:	9
1.1.5	Figure:	9
1.1.6	Schema Fragment:	9
1.2	ConversationId	10
1.2.1	Description:	10
1.2.2	Contents:	10
1.2.3	Used by:	10
1.2.4	Derived Types:	10
1.2.5	Figure:	10
1.2.6	Schema Fragment:	10
1.3	Message	11
1.3.1	Description:	11
1.3.2	Contents:	11
1.3.3	Used by:	11
1.3.4	Derived Types:	11
1.3.5	Figure:	11
1.3.6	Schema Fragment:	11
1.4	MessageAddress	12
1.4.1	Description:	12
1.4.2	Contents:	12
1.4.3	Used by:	12
1.4.4	Derived Types:	12
1.4.5	Figure:	12
1.4.6	Schema Fragment:	12
1.5	MessageHeader	13
1.5.1	Description:	13
1.5.2	Contents:	13
1.5.3	Used by:	13
1.5.4	Derived Types:	13
1.5.5	Figure:	13
1.5.6	Schema Fragment:	13
1.6	MessageId	15
1.6.1	Description:	15
1.6.2	Contents:	15
1.6.3	Used by:	15
1.6.4	Derived Types:	15
1.6.5	Figure:	15
1.6.6	Schema Fragment:	15
1.7	MessageRejected	16
1.7.1	Description:	16
1.7.2	Contents:	16
1.7.3	Used by:	16
1.7.4	Derived Types:	16
1.7.5	Figure:	16
1.7.6	Schema Fragment:	16
1.8	NotificationMessage	17
1.8.1	Description:	17
1.8.2	Contents:	17
1.8.3	Used by:	17
1.8.4	Derived Types:	17
1.8.5	Figure:	18
1.8.6	Schema Fragment:	18
1.9	NotificationMessageHeader	19
1.9.1	Description:	19
1.9.2	Contents:	19

1.9.3	Used by:	19
1.9.4	Derived Types:	19
1.9.5	Figure:	19
1.9.6	Schema Fragment:	20
1.10	<b>PartyMessageInformation</b>	21
1.10.1	Description:	21
1.10.2	Contents:	21
1.10.3	Used by:	21
1.10.4	Derived Types:	21
1.10.5	Figure:	21
1.10.6	Schema Fragment:	21
1.11	<b>ProblemLocation</b>	22
1.11.1	Description:	22
1.11.2	Contents:	22
1.11.3	Used by:	22
1.11.4	Derived Types:	22
1.11.5	Figure:	22
1.11.6	Schema Fragment:	22
1.12	<b>Reason</b>	23
1.12.1	Description:	23
1.12.2	Contents:	23
1.12.3	Used by:	23
1.12.4	Derived Types:	23
1.12.5	Figure:	23
1.12.6	Schema Fragment:	23
1.13	<b>ReasonCode</b>	25
1.13.1	Description:	25
1.13.2	Contents:	25
1.13.3	Used by:	25
1.13.4	Derived Types:	25
1.13.5	Figure:	25
1.13.6	Schema Fragment:	25
1.14	<b>RequestMessage</b>	26
1.14.1	Description:	26
1.14.2	Contents:	26
1.14.3	Used by:	26
1.14.4	Derived Types:	26
1.14.5	Figure:	27
1.14.6	Schema Fragment:	27
1.15	<b>RequestMessageHeader</b>	28
1.15.1	Description:	28
1.15.2	Contents:	28
1.15.3	Used by:	28
1.15.4	Derived Types:	28
1.15.5	Figure:	28
1.15.6	Schema Fragment:	29
1.16	<b>RequestTradeStatus</b>	30
1.16.1	Description:	30
1.16.2	Contents:	30
1.16.3	Used by:	30
1.16.4	Derived Types:	30
1.16.5	Figure:	30
1.16.6	Schema Fragment:	30
1.17	<b>ResponseMessage</b>	32
1.17.1	Description:	32
1.17.2	Contents:	32
1.17.3	Used by:	32
1.17.4	Derived Types:	32
1.17.5	Figure:	33
1.17.6	Schema Fragment:	33
1.18	<b>ResponseMessageHeader</b>	34
1.18.1	Description:	34
1.18.2	Contents:	34
1.18.3	Used by:	34

1.18.4	Derived Types:	34
1.18.5	Figure:	34
1.18.6	Schema Fragment:	35
1.19	<b>TradeAlreadyCancelled</b>	36
1.19.1	Description:	36
1.19.2	Contents:	36
1.19.3	Used by:	36
1.19.4	Derived Types:	36
1.19.5	Figure:	36
1.19.6	Schema Fragment:	36
1.20	<b>TradeAlreadySubmitted</b>	37
1.20.1	Description:	37
1.20.2	Contents:	37
1.20.3	Used by:	37
1.20.4	Derived Types:	37
1.20.5	Figure:	37
1.20.6	Schema Fragment:	37
1.21	<b>TradeAlreadyTerminated</b>	39
1.21.1	Description:	39
1.21.2	Contents:	39
1.21.3	Used by:	39
1.21.4	Derived Types:	39
1.21.5	Figure:	39
1.21.6	Schema Fragment:	39
1.22	<b>TradeErrorResponse</b>	40
1.22.1	Description:	40
1.22.2	Contents:	40
1.22.3	Used by:	40
1.22.4	Derived Types:	40
1.22.5	Figure:	40
1.22.6	Schema Fragment:	40
1.23	<b>TradeNotFound</b>	42
1.23.1	Description:	42
1.23.2	Contents:	42
1.23.3	Used by:	42
1.23.4	Derived Types:	42
1.23.5	Figure:	42
1.23.6	Schema Fragment:	42
1.24	<b>TradeStatus</b>	44
1.24.1	Description:	44
1.24.2	Contents:	44
1.24.3	Used by:	44
1.24.4	Derived Types:	44
1.24.5	Figure:	44
1.24.6	Schema Fragment:	44
1.25	<b>TradeStatusItem</b>	46
1.25.1	Description:	46
1.25.2	Contents:	46
1.25.3	Used by:	46
1.25.4	Derived Types:	46
1.25.5	Figure:	46
1.25.6	Schema Fragment:	46
1.26	<b>TradeStatusValue</b>	47
1.26.1	Description:	47
1.26.2	Contents:	47
1.26.3	Used by:	47
1.26.4	Derived Types:	47
1.26.5	Figure:	47
1.26.6	Schema Fragment:	47
2	<b>Groups</b>	48
2.1	<b>Exception.model</b>	49
2.1.1	Description:	49
2.1.2	Contents:	49
2.1.3	Used by:	49

2.1.4	Figure:	49
2.1.5	Schema Fragment:	49
2.2	MessageHeader.model	50
2.2.1	Description:	50
2.2.2	Contents:	50
2.2.3	Used by:	50
2.2.4	Figure:	50
2.2.5	Schema Fragment:	50
3	Schema listing	52

## ***1 Global Complex Types***



## 1.1 AdditionalData

### 1.1.1 Description:

Provides extra information not represented in the model that may be useful in processing the message i.e. diagnosing the reason for failure. In case the extra information is in XML format, a CDATA section must be placed around the source message to prevent its interpretation as XML content.

### 1.1.2 Contents:

Inherited element(s): (This definition inherits the content defined by the type xsd:string)

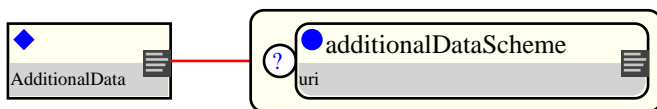
•

### 1.1.3 Used by:

- Complex type: Reason

### 1.1.4 Derived Types:

### 1.1.5 Figure:



### 1.1.6 Schema Fragment:

```
<xsd:complexType name="AdditionalData">
  <xsd:annotation>
    <xsd:documentation xml:lang="en">
      Provides extra information not represented in the model that may
      be useful in processing the message i.e. diagnosing the reason
      for failure. In case the extra information is in XML format, a
      CDATA section must be placed around the source message to prevent
      its interpretation as XML content.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:simpleContent>
    <xsd:extension base="xsd:string">
      <xsd:attribute name="additionalDataScheme" type="xsd:anyURI" use="optional"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
```

## 1.2 ConversationId

### 1.2.1 Description:

The unique identifier (name) for the conversation (session), this message is within. A conversation identifier is usually assigned by the initiator of a conversation. Conversations may only be initiated and terminated. Joining conversations has the effect of initiating new conversations. Conversations cannot be split; this instead has the effect of parallel activities on the same conversation or the initiation of a new conversation. Each message belongs to only one conversation. Conversation scopes are defined in the business process definition.

### 1.2.2 Contents:

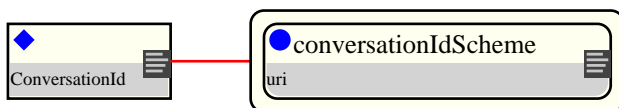
Inherited element(s): (This definition inherits the content defined by the type `xsd:normalizedString`)

### 1.2.3 Used by:

- Complex type: `MessageHeader`

### 1.2.4 Derived Types:

### 1.2.5 Figure:



### 1.2.6 Schema Fragment:

```
<xsd:complexType name="ConversationId">
  <xsd:annotation>
    <xsd:documentation xml:lang="en">
      The unique identifier (name) for the conversation (session), this
      message is within. A conversation identifier is usually assigned
      by the initiator of a conversation. Conversations may only be
      initiated and terminated. Joining conversations has the effect of
      initiating new conversations. Conversations cannot be split; this
      instead has the effect of parallel activities on the same
      conversation or the initiation of a new conversation. Each
      message belongs to only one conversation. Conversation scopes are
      defined in the business process definition.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:simpleContent>
    <xsd:extension base="xsd:normalizedString">
      <xsd:attribute name="conversationIdScheme" type="xsd:anyURI" use="required"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
```

## 1.3 Message

### 1.3.1 Description:

A type defining the basic structure of all FpML messages which is refined by its derived types.

### 1.3.2 Contents:

Inherited element(s): (This definition inherits the content defined by the type Document)

- The abstract base type from which all FpML compliant messages and documents must be derived.

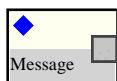
### 1.3.3 Used by:

- Complex type: NotificationMessage
- Complex type: RequestMessage
- Complex type: ResponseMessage

### 1.3.4 Derived Types:

- Complex type: NotificationMessage
- Complex type: RequestMessage
- Complex type: ResponseMessage

### 1.3.5 Figure:



### 1.3.6 Schema Fragment:

```
<xsd:complexType name="Message" abstract="true">
  <xsd:annotation>
    <xsd:documentation xml:lang="en">
      A type defining the basic structure of all FpML messages which is
      refined by its derived types.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:complexContent>
    <xsd:extension base="Document" />
  </xsd:complexContent>
</xsd:complexType>
```

## 1.4 MessageAddress

### 1.4.1 Description:

The data type used for identifying a message address. It includes a partyIdScheme for keeping the type backward compatible with the PartyId complex type. In the next major version, partyIdScheme attribute will be replaced by a messageAddressScheme attribute.

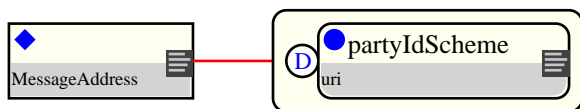
### 1.4.2 Contents:

Inherited element(s): (This definition inherits the content defined by the type xsd:normalizedString)

### 1.4.3 Used by:

### 1.4.4 Derived Types:

### 1.4.5 Figure:



### 1.4.6 Schema Fragment:

```
<xsd:complexType name="MessageAddress">
  <xsd:annotation>
    <xsd:documentation xml:lang="en">
      The data type used for identifying a message address. It includes
      a partyIdScheme for keeping the type backward compatible with the
      PartyId complex type. In the next major version, partyIdScheme
      attribute will be replaced by a messageAddressScheme attribute.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:simpleContent>
    <xsd:extension base="xsd:normalizedString">
      <xsd:attribute name="partyIdScheme" type="xsd:anyURI" default="http://www.fpml.org/ext/is
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
```

## 1.5 MessageHeader

### 1.5.1 Description:

A type defining the content model for a generic message header that is refined by its derived classes.

### 1.5.2 Contents:

**conversationId** (zero or one occurrence; of the type ConversationId) The unique identifier (name) for the conversation (session), this message is within. A conversation identifier is usually assigned by the initiator of a conversation. Conversations may only be initiated and terminated. Joining conversations has the effect of initiating new conversations. Conversations cannot be split; this instead has the effect of parallel activities on the same conversation or the initiation of a new conversation. Each message belongs to only one conversation. Conversation scopes are defined in the business process definition.

**messageId** (exactly one occurrence; of the type MessageId) A unique identifier (within its coding scheme) assigned to the message by its creating party.

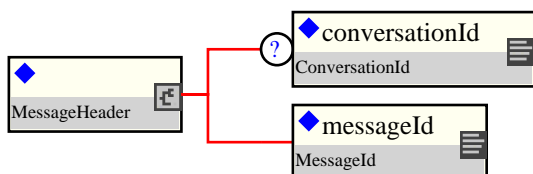
### 1.5.3 Used by:

- Complex type: NotificationMessageHeader
- Complex type: RequestMessageHeader
- Complex type: ResponseMessageHeader

### 1.5.4 Derived Types:

- Complex type: NotificationMessageHeader
- Complex type: RequestMessageHeader
- Complex type: ResponseMessageHeader

### 1.5.5 Figure:



### 1.5.6 Schema Fragment:

```
<xsd:complexType name="MessageHeader" abstract="true">
  <xsd:annotation>
    <xsd:documentation xml:lang="en">
      A type defining the content model for a generic message header
      that is refined by its derived classes.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:sequence>
    <xsd:element name="conversationId" type="ConversationId" minOccurs="0">
      <xsd:annotation>
        <xsd:documentation xml:lang="en">
          The unique identifier (name) for the conversation (session),
          this message is within. A conversation identifier is usually
          assigned by the initiator of a conversation. Conversations
          may only be initiated and terminated. Joining conversations
          has the effect of initiating new conversations. Conversations
          cannot be split; this instead has the effect of parallel
          activities on the same conversation or the initiation of a
          new conversation. Each message belongs to only one
          conversation. Conversation scopes are defined in the business
          process definition.
        </xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>
```

```
</xsd:element>
<xsd:element name="messageId" type="MessageId">
  <xsd:annotation>
    <xsd:documentation xml:lang="en">
      A unique identifier (within its coding scheme) assigned to
      the message by its creating party.
    </xsd:documentation>
  </xsd:annotation>
</xsd:element>
</xsd:sequence>
</xsd:complexType>
```

## 1.6 MessageId

### 1.6.1 Description:

The data type use for message identifiers.

### 1.6.2 Contents:

Inherited element(s): (This definition inherits the content defined by the type `xsd:normalizedString`)

•

### 1.6.3 Used by:

- Complex type: MessageHeader
- Complex type: NotificationMessageHeader
- Complex type: ResponseMessageHeader

### 1.6.4 Derived Types:

### 1.6.5 Figure:



### 1.6.6 Schema Fragment:

```
<xsd:complexType name="MessageId">
  <xsd:annotation>
    <xsd:documentation xml:lang="en">
      The data type use for message identifiers.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:simpleContent>
    <xsd:extension base="xsd:normalizedString">
      <xsd:attribute name="messageIdScheme" type="xsd:anyURI" use="required"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
```

## 1.7 MessageRejected

### 1.7.1 Description:

A type defining the content for a standard message sent when a recipient cannot interpret or process an earlier message.

### 1.7.2 Contents:

Inherited element(s): (This definition inherits the content defined by the type NotificationMessage)

- A type defining the basic content for a message sent to inform another system that some 'business event' has occurred. Notifications are not expected to be replied to.

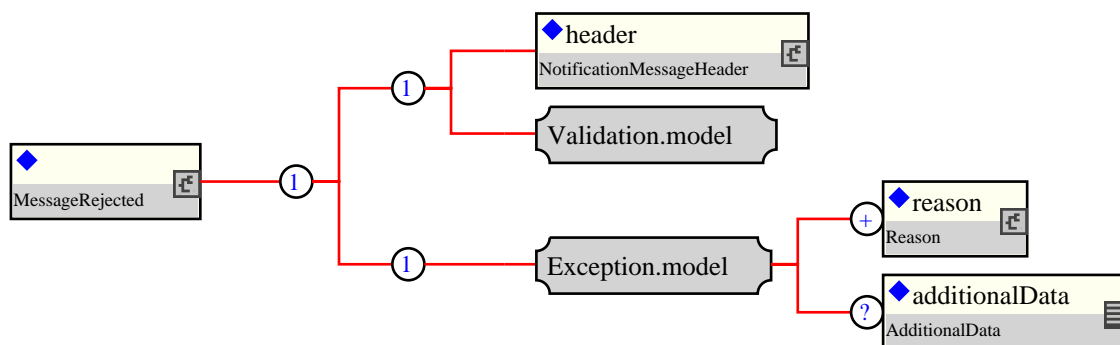
**reason** (one or more occurrences; of the type Reason) An instance of the Reason type used to record the nature of any errors associated with a message.

**additionalData** (zero or one occurrence; of the type AdditionalData) Any string of additional data that may help the message processor, for example in a rejection message this might contain a code value or the text of the original request (within a CDATA section).

### 1.7.3 Used by:

### 1.7.4 Derived Types:

### 1.7.5 Figure:



### 1.7.6 Schema Fragment:

```
<xsd:complexType name="MessageRejected">
  <xsd:annotation>
    <xsd:documentation xml:lang="en">
      A type defining the content for a standard message sent when a
      recipient cannot interpret or process an earlier message.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:complexContent>
    <xsd:extension base="NotificationMessage">
      <xsd:sequence>
        <xsd:group ref="Exception.model" />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```



## 1.8 NotificationMessage

### 1.8.1 Description:

A type defining the basic content for a message sent to inform another system that some 'business event' has occurred. Notifications are not expected to be replied to.

### 1.8.2 Contents:

Inherited element(s): (This definition inherits the content defined by the type Message)

- A type defining the basic structure of all FpML messages which is refined by its derived types.

**header** (exactly one occurrence; of the type NotificationMessageHeader)

**validation** (zero or more occurrences; of the type Validation)

### 1.8.3 Used by:

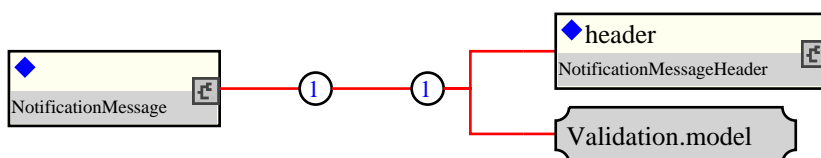
- Complex type: AllocationAmended
- Complex type: AllocationCancelled
- Complex type: AllocationCreated
- Complex type: AmendmentConfirmed
- Complex type: CancelTradeCashflows
- Complex type: ContractCreated
- Complex type: ContractFullTermination
- Complex type: ContractIncreased
- Complex type: ContractNovated
- Complex type: ContractPartialTermination
- Complex type: ContractReferenceMessage
- Complex type: CreditEventNotification
- Complex type: IncreaseConfirmed
- Complex type: MessageRejected
- Complex type: NovationNotificationMessage
- Complex type: PositionReport
- Complex type: TerminationConfirmed
- Complex type: TradeAffirmation
- Complex type: TradeAlleged
- Complex type: TradeAmended
- Complex type: TradeCancelled
- Complex type: TradeCashflowsAsserted
- Complex type: TradeConfirmed
- Complex type: TradeCreated
- Complex type: TradeMatched
- Complex type: TradeMismatched
- Complex type: TradeUnmatched
- Complex type: ValuationReport

### 1.8.4 Derived Types:

- Complex type: AllocationAmended
- Complex type: AllocationCancelled
- Complex type: AllocationCreated
- Complex type: AmendmentConfirmed
- Complex type: CancelTradeCashflows

- Complex type: ContractCreated
- Complex type: ContractFullTermination
- Complex type: ContractIncreased
- Complex type: ContractNovated
- Complex type: ContractPartialTermination
- Complex type: ContractReferenceMessage
- Complex type: CreditEventNotification
- Complex type: IncreaseConfirmed
- Complex type: MessageRejected
- Complex type: NovationNotificationMessage
- Complex type: PositionReport
- Complex type: TerminationConfirmed
- Complex type: TradeAffirmation
- Complex type: TradeAlleged
- Complex type: TradeAmended
- Complex type: TradeCancelled
- Complex type: TradeCashflowsAsserted
- Complex type: TradeConfirmed
- Complex type: TradeCreated
- Complex type: TradeMatched
- Complex type: TradeMismatched
- Complex type: TradeUnmatched
- Complex type: ValuationReport

### 1.8.5 Figure:



### 1.8.6 Schema Fragment:

```

<xsd:complexType name="NotificationMessage" abstract="true">
  <xsd:annotation>
    <xsd:documentation xml:lang="en">
      A type defining the basic content for a message sent to inform
      another system that some 'business event' has occurred.
      Notifications are not expected to be replied to.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:complexContent>
    <xsd:extension base="Message">
      <xsd:sequence>
        <xsd:element name="header" type="NotificationMessageHeader"/>
        <xsd:group ref="Validation.model"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
  
```

## 1.9 NotificationMessageHeader

### 1.9.1 Description:

A type that refines the generic message header to match the requirements of a NotificationMessage.

### 1.9.2 Contents:

Inherited element(s): (This definition inherits the content defined by the type MessageHeader)

- A type defining the content model for a generic message header that is refined by its derived classes.

**inReplyTo** (zero or one occurrence; of the type MessageId) A copy of the unique message identifier (within its own coding scheme) to which this message is responding.

**sentBy** (exactly one occurrence; of the type MessageAddress) The unique identifier (within its coding scheme) for the originator of a message instance.

**sendTo** (zero or more occurrences; of the type MessageAddress) A unique identifier (within its coding scheme) indicating an intended recipient of a message.

**copyTo** (zero or more occurrences; of the type MessageAddress) A unique identifier (within the specified coding scheme) giving the details of some party to whom a copy of this message will be sent for reference.

**creationTimestamp** (exactly one occurrence; of the type xsd:dateTime) The date and time (on the source system) when this message instance was created.

**expiryTimestamp** (zero or one occurrence; of the type xsd:dateTime) The date and time (on the source system) when this message instance will be considered expired.

**partyMessageInformation** (zero or more occurrences; of the type PartyMessageInformation) Additional message information that may be provided by each involved party.

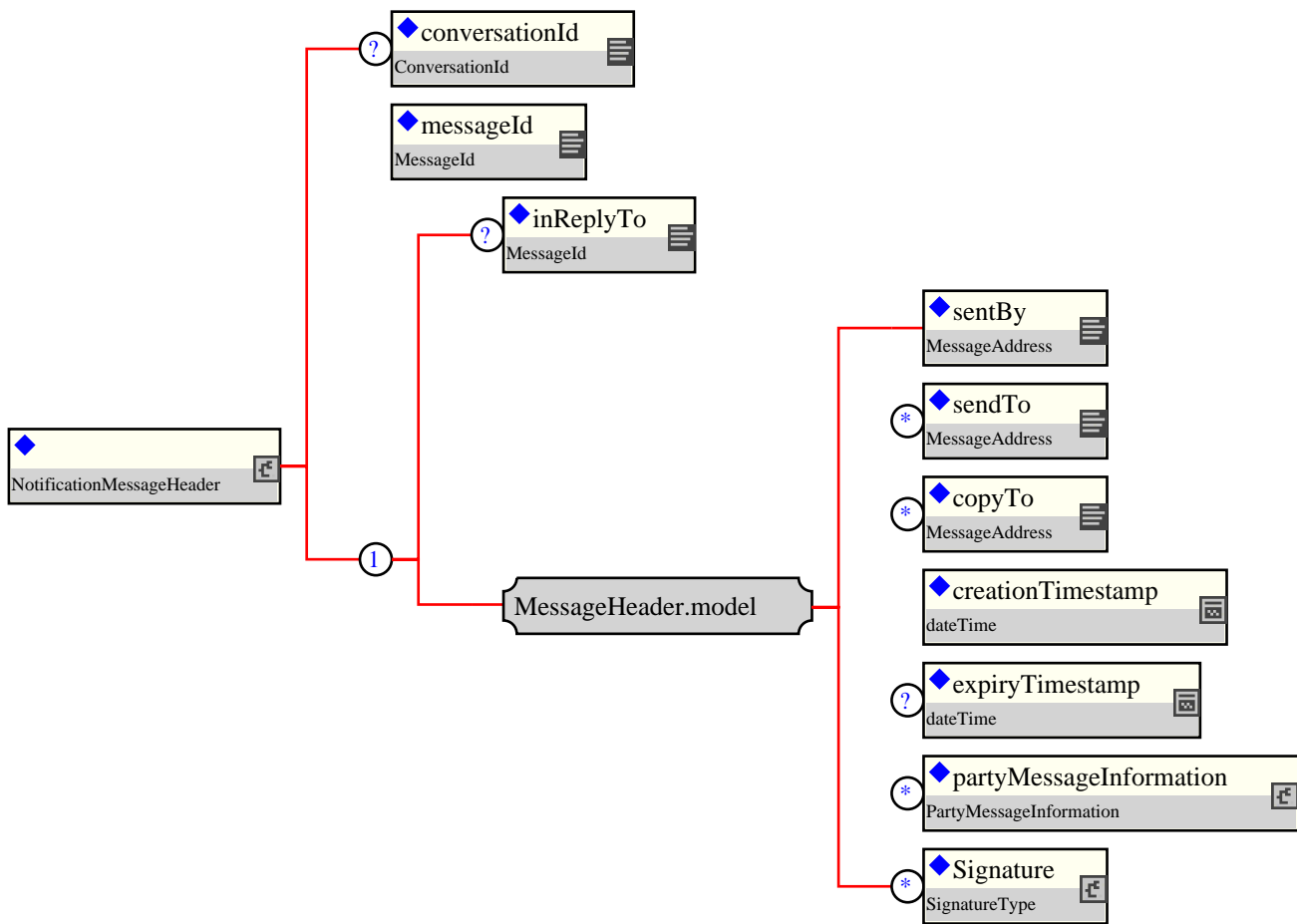
**dsig:Signature** (zero or more occurrences;

### 1.9.3 Used by:

- Complex type: NotificationMessage

### 1.9.4 Derived Types:

### 1.9.5 Figure:



### 1.9.6 Schema Fragment:

```

<xsd:complexType name="NotificationMessageHeader">
  <xsd:annotation>
    <xsd:documentation xml:lang="en">
      A type that refines the generic message header to match the
      requirements of a NotificationMessage.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:complexContent>
    <xsd:extension base="MessageHeader">
      <xsd:sequence>
        <xsd:element name="inReplyTo" type="MessageId" minOccurs="0">
          <xsd:annotation>
            <xsd:documentation xml:lang="en">
              A copy of the unique message identifier (within its own
              coding scheme) to which this message is responding.
            </xsd:documentation>
          </xsd:annotation>
        </xsd:element>
        <xsd:group ref="MessageHeader.model"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

## 1.10 PartyMessageInformation

### 1.10.1 Description:

A type defining additional information that may be recorded against a message.

### 1.10.2 Contents:

**partyReference** (exactly one occurrence; of the type PartyReference) Identifies that party that has ownership of this information.

### 1.10.3 Used by:

### 1.10.4 Derived Types:

### 1.10.5 Figure:



### 1.10.6 Schema Fragment:

```
<xsd:complexType name="PartyMessageInformation">
  <xsd:annotation>
    <xsd:documentation xml:lang="en">
      A type defining additional information that may be recorded
      against a message.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:sequence>
    <xsd:element name="partyReference" type="PartyReference">
      <xsd:annotation>
        <xsd:documentation xml:lang="en">
          Identifies that party that has ownership of this information.
        </xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>
```

## 1.11 ProblemLocation

### 1.11.1 Description:

Provides a lexical location (i.e. a line number and character for bad XML) or an XPath location (i.e. place to identify the bad location for valid XML).

### 1.11.2 Contents:

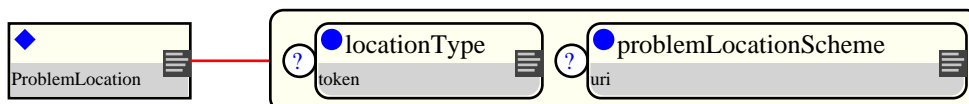
Inherited element(s): (This definition inherits the content defined by the type `xsd:normalizedString`)

### 1.11.3 Used by:

- Complex type: Reason

### 1.11.4 Derived Types:

### 1.11.5 Figure:



### 1.11.6 Schema Fragment:

```
<xsd:complexType name="ProblemLocation">
  <xsd:annotation>
    <xsd:documentation xml:lang="en">
      Provides a lexical location (i.e. a line number and character for
      bad XML) or an XPath location (i.e. place to identify the bad
      location for valid XML).
    </xsd:documentation>
  </xsd:annotation>
  <xsd:simpleContent>
    <xsd:extension base="xsd:normalizedString">
      <xsd:attribute name="locationType" type="xsd:token">
        <xsd:annotation>
          <xsd:documentation xml:lang="en">
            The value of the locationType attribute defines which type
            of location has been given. It may take the values
            'lexical' or 'xpath'.
          </xsd:documentation>
        </xsd:annotation>
      </xsd:attribute>
      <xsd:attribute name="problemLocationScheme" type="xsd:anyURI" fpml-annotation:deprecated="true">
        <xsd:annotation>
          <xsd:documentation xml:lang="en">
            DEPRECATED. It will be removed in FpML 5.0. New
            implementations are encouraged to use the locationType
            attribute.
          </xsd:documentation>
        </xsd:annotation>
      </xsd:attribute>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
```

## 1.12 Reason

### 1.12.1 Description:

A type defining a content model for describing the nature and possible location of a error within a previous message.

### 1.12.2 Contents:

**reasonCode** (exactly one occurrence; of the type ReasonCode) A machine interpretable error code.

**location** (zero or one occurrence; of the type ProblemLocation) A value indicating the location of the problem within the subject message.

**description** (zero or one occurrence; of the type xsd:string) Plain English text describing the associated error condition

**validationRuleId** (zero or one occurrence; of the type Validation) A reference identifying a rule within a validation scheme

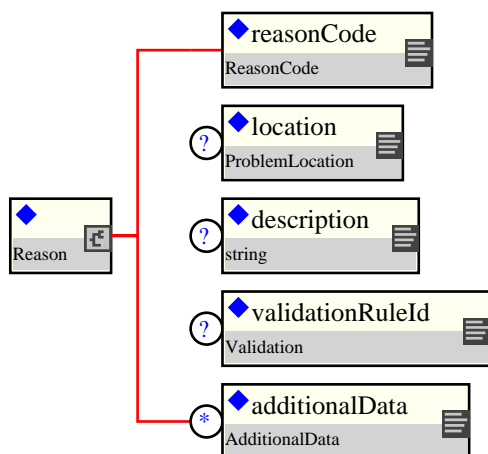
**additionalData** (zero or more occurrences; of the type AdditionalData) Any string of additional data that may help the message processor, for example in a rejection message this might contain a code value or the text of any one of the messages (within a CDATA section).

### 1.12.3 Used by:

- Complex type: NovationConsentRefused
- Complex type: UnprocessedPosition

### 1.12.4 Derived Types:

### 1.12.5 Figure:



### 1.12.6 Schema Fragment:

```
<xsd:complexType name="Reason">
  <xsd:annotation>
    <xsd:documentation xml:lang="en">
      A type defining a content model for describing the nature and
      possible location of a error within a previous message.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:sequence>
    <xsd:element name="reasonCode" type="ReasonCode">
      <xsd:annotation>
        <xsd:documentation xml:lang="en">
```

```

        A machine interpretable error code.
    </xsd:documentation>
</xsd:annotation>
</xsd:element>
<xsd:element name="location" type="ProblemLocation" minOccurs="0">
    <xsd:annotation>
        <xsd:documentation xml:lang="en">
            A value indicating the location of the problem within the
            subject message.
        </xsd:documentation>
    </xsd:annotation>
</xsd:element>
<xsd:element name="description" type="xsd:string" minOccurs="0">
    <xsd:annotation>
        <xsd:documentation xml:lang="en">
            Plain English text describing the associated error condition
        </xsd:documentation>
    </xsd:annotation>
</xsd:element>
<xsd:element name="validationRuleId" type="Validation" minOccurs="0">
    <xsd:annotation>
        <xsd:documentation xml:lang="en">
            A reference identifying a rule within a validation scheme
        </xsd:documentation>
    </xsd:annotation>
</xsd:element>
<xsd:element name="additionalData" type="AdditionalData" minOccurs="0" maxOccurs="unbounded">
    <xsd:annotation>
        <xsd:documentation xml:lang="en">
            Any string of additional data that may help the message
            processor, for example in a rejection message this might
            contain a code value or the text of any one of the messages
            (within a CDATA section).
        </xsd:documentation>
    </xsd:annotation>
</xsd:element>
</xsd:sequence>
</xsd:complexType>

```



## 1.13 ReasonCode

### 1.13.1 Description:

Defines a list of machine interpretable error codes.

### 1.13.2 Contents:

Inherited element(s): (This definition inherits the content defined by the type xsd:normalizedString)

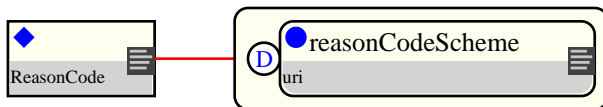
•

### 1.13.3 Used by:

- Complex type: Reason

### 1.13.4 Derived Types:

### 1.13.5 Figure:



### 1.13.6 Schema Fragment:

```
<xsd:complexType name="ReasonCode">
  <xsd:annotation>
    <xsd:documentation xml:lang="en">
      Defines a list of machine interpretable error codes.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:simpleContent>
    <xsd:extension base="xsd:normalizedString">
      <xsd:attribute name="reasonCodeScheme" type="xsd:anyURI" default="http://www.fpml.org/cod" />
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
```

## 1.14 RequestMessage

### 1.14.1 Description:

A type defining the basic content of a message that requests the receiver to perform some business operation determined by the message type and its content.

### 1.14.2 Contents:

Inherited element(s): (This definition inherits the content defined by the type Message)

- A type defining the basic structure of all FpML messages which is refined by its derived types.

**header** (exactly one occurrence; of the type RequestMessageHeader)

**validation** (zero or more occurrences; of the type Validation)

### 1.14.3 Used by:

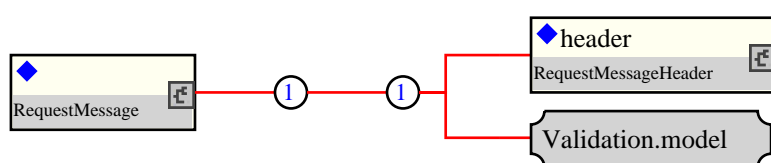
- Complex type: CancelTradeConfirmation
- Complex type: CancelTradeMatch
- Complex type: ConfirmTrade
- Complex type: ModifyTradeConfirmation
- Complex type: ModifyTradeMatch
- Complex type: NovationRequestMessage
- Complex type: PositionsAsserted
- Complex type: RequestAllocation
- Complex type: RequestAmendmentConfirmation
- Complex type: RequestIncreaseConfirmation
- Complex type: RequestPortfolio
- Complex type: RequestPositionReport
- Complex type: RequestQuote
- Complex type: RequestTerminationConfirmation
- Complex type: RequestTradeConfirmation
- Complex type: RequestTradeMatch
- Complex type: RequestTradeStatus
- Complex type: RequestValuationReport
- Complex type: TradeAmendmentRequest
- Complex type: TradeIncreaseRequest
- Complex type: TradeTerminationRequest

### 1.14.4 Derived Types:

- Complex type: CancelTradeConfirmation
- Complex type: CancelTradeMatch
- Complex type: ConfirmTrade
- Complex type: ModifyTradeConfirmation
- Complex type: ModifyTradeMatch
- Complex type: NovationRequestMessage
- Complex type: PositionsAsserted
- Complex type: RequestAllocation
- Complex type: RequestAmendmentConfirmation
- Complex type: RequestIncreaseConfirmation
- Complex type: RequestPortfolio
- Complex type: RequestPositionReport

- Complex type: RequestQuote
- Complex type: RequestTerminationConfirmation
- Complex type: RequestTradeConfirmation
- Complex type: RequestTradeMatch
- Complex type: RequestTradeStatus
- Complex type: RequestValuationReport
- Complex type: TradeAmendmentRequest
- Complex type: TradeIncreaseRequest
- Complex type: TradeTerminationRequest

#### 1.14.5 Figure:



#### 1.14.6 Schema Fragment:

```

<xsd:complexType name="RequestMessage" abstract="true">
  <xsd:annotation>
    <xsd:documentation xml:lang="en">
      A type defining the basic content of a message that requests the
      receiver to perform some business operation determined by the
      message type and its content.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:complexContent>
    <xsd:extension base="Message">
      <xsd:sequence>
        <xsd:element name="header" type="RequestMessageHeader"/>
        <xsd:group ref="Validation.model"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
  
```

## 1.15 RequestMessageHeader

### 1.15.1 Description:

A type refining the generic message header content to make it specific to request messages.

### 1.15.2 Contents:

Inherited element(s): (This definition inherits the content defined by the type MessageHeader)

- A type defining the content model for a generic message header that is refined by its derived classes.

**sentBy** (exactly one occurrence; of the type MessageAddress) The unique identifier (within its coding scheme) for the originator of a message instance.

**sendTo** (zero or more occurrences; of the type MessageAddress) A unique identifier (within its coding scheme) indicating an intended recipient of a message.

**copyTo** (zero or more occurrences; of the type MessageAddress) A unique identifier (within the specified coding scheme) giving the details of some party to whom a copy of this message will be sent for reference.

**creationTimestamp** (exactly one occurrence; of the type xsd:dateTime) The date and time (on the source system) when this message instance was created.

**expiryTimestamp** (zero or one occurrence; of the type xsd:dateTime) The date and time (on the source system) when this message instance will be considered expired.

**partyMessageInformation** (zero or more occurrences; of the type PartyMessageInformation) Additional message information that may be provided by each involved party.

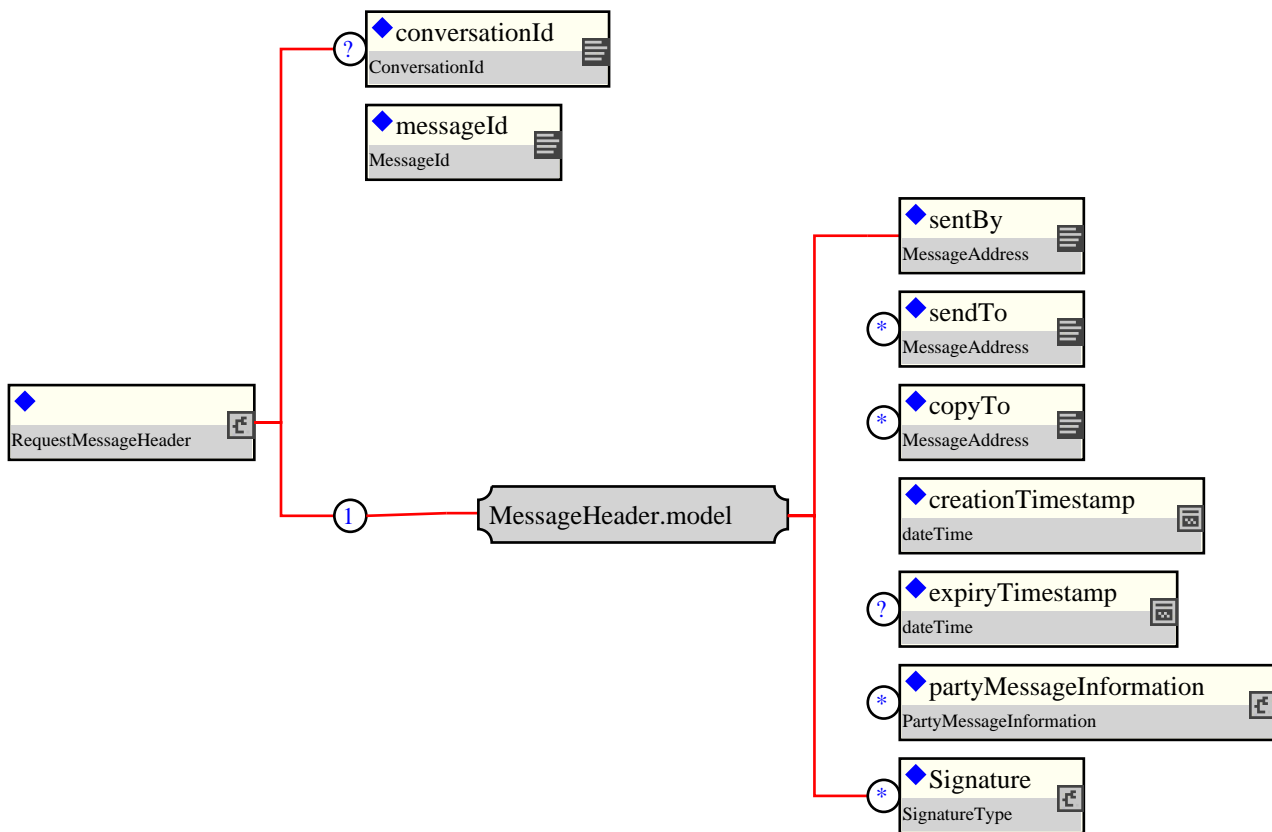
**dsig:Signature** (zero or more occurrences;

### 1.15.3 Used by:

- Complex type: RequestMessage

### 1.15.4 Derived Types:

### 1.15.5 Figure:



### 1.15.6 Schema Fragment:

```

<xsd:complexType name="RequestMessageHeader">
  <xsd:annotation>
    <xsd:documentation xml:lang="en">
      A type refining the generic message header content to make it
      specific to request messages.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:complexContent>
    <xsd:extension base="MessageHeader">
      <xsd:sequence>
        <xsd:group ref="MessageHeader.model" />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

## 1.16 RequestTradeStatus

### 1.16.1 Description:

A type defining the content model for a message allowing one party to query the status of one or many trades previously sent to another party.

### 1.16.2 Contents:

Inherited element(s): (This definition inherits the content defined by the type RequestMessage)

- A type defining the basic content of a message that requests the receiver to perform some business operation determined by the message type and its content.

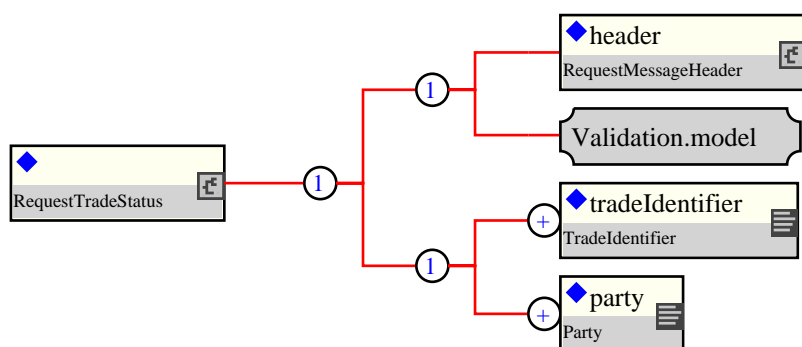
**tradeIdentifier** (one or more occurrences; of the type TradeIdentifier) An instance of a unique trade identifier.

**party** (one or more occurrences; of the type Party) A legal entity or a subdivision of a legal entity.

### 1.16.3 Used by:

### 1.16.4 Derived Types:

### 1.16.5 Figure:



### 1.16.6 Schema Fragment:

```
<xsd:complexType name="RequestTradeStatus">
  <xsd:annotation>
    <xsd:documentation xml:lang="en">
      A type defining the content model for a message allowing one
      party to query the status of one or many trades previously sent
      to another party.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:complexContent>
    <xsd:extension base="RequestMessage">
      <xsd:sequence>
        <xsd:element name="tradeIdentifier" type="TradeIdentifier" maxOccurs="unbounded">
          <xsd:annotation>
            <xsd:documentation xml:lang="en">
              An instance of a unique trade identifier.
            </xsd:documentation>
          </xsd:annotation>
        </xsd:element>
        <xsd:element name="party" type="Party" maxOccurs="unbounded">
          <xsd:annotation>
            <xsd:documentation xml:lang="en">
              A legal entity or a subdivision of a legal entity.
            </xsd:documentation>
            <xsd:documentation xml:lang="en">
              Parties can perform multiple roles in a trade lifecycle.
            </xsd:documentation>
          </xsd:annotation>
        </xsd:element>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

For example, the principal parties obligated to make payments from time to time during the term of the trade, but may include other parties involved in, or incidental to, the trade, such as parties acting in the role of novation transferor/transferee, broker, calculation agent, etc. In FpML roles are defined in multiple places within a document.

```
</xsd:documentation>
</xsd:annotation>
</xsd:element>
</xsd:sequence>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>
```

## 1.17 ResponseMessage

### 1.17.1 Description:

A type refining the generic message content model to make it specific to response messages.

### 1.17.2 Contents:

Inherited element(s): (This definition inherits the content defined by the type Message)

- A type defining the basic structure of all FpML messages which is refined by its derived types.

**header** (exactly one occurrence; of the type ResponseMessageHeader)

**validation** (zero or more occurrences; of the type Validation)

### 1.17.3 Used by:

- Complex type: AcceptQuote
- Complex type: ConfirmationCancelled
- Complex type: NovationResponseMessage
- Complex type: PositionsAcknowledged
- Complex type: PositionsMatchResults
- Complex type: QuoteAcceptanceConfirmed
- Complex type: QuoteAlreadyExpired
- Complex type: QuoteUpdated
- Complex type: RequestQuoteResponse
- Complex type: TradeAffirmed
- Complex type: TradeAlreadyMatched
- Complex type: TradeAlreadySubmitted
- Complex type: TradeAmendmentResponse
- Complex type: TradeCashflowsMatchResult
- Complex type: TradeErrorResponse
- Complex type: TradeIncreaseResponse
- Complex type: TradeNotFound
- Complex type: TradeStatus
- Complex type: TradeTerminationResponse

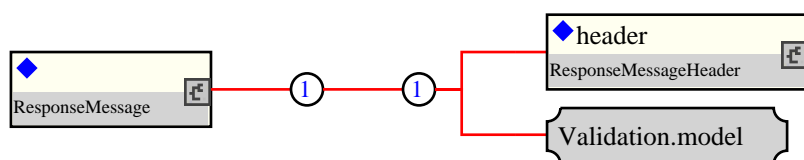
### 1.17.4 Derived Types:

- Complex type: AcceptQuote
- Complex type: ConfirmationCancelled
- Complex type: NovationResponseMessage
- Complex type: PositionsAcknowledged
- Complex type: PositionsMatchResults
- Complex type: QuoteAcceptanceConfirmed
- Complex type: QuoteAlreadyExpired
- Complex type: QuoteUpdated
- Complex type: RequestQuoteResponse
- Complex type: TradeAffirmed
- Complex type: TradeAlreadyMatched
- Complex type: TradeAlreadySubmitted
- Complex type: TradeAmendmentResponse
- Complex type: TradeCashflowsMatchResult
- Complex type: TradeErrorResponse



- Complex type: TradeIncreaseResponse
- Complex type: TradeNotFound
- Complex type: TradeStatus
- Complex type: TradeTerminationResponse

### 1.17.5 Figure:



### 1.17.6 Schema Fragment:

```

<xsd:complexType name="ResponseMessage" abstract="true">
  <xsd:annotation>
    <xsd:documentation xml:lang="en">
      A type refining the generic message content model to make it
      specific to response messages.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:complexContent>
    <xsd:extension base="Message">
      <xsd:sequence>
        <xsd:element name="header" type="ResponseMessageHeader"/>
        <xsd:group ref="Validation.model"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
  
```

## 1.18 ResponseMessageHeader

### 1.18.1 Description:

A type refining the generic message header to make it specific to response messages.

### 1.18.2 Contents:

Inherited element(s): (This definition inherits the content defined by the type MessageHeader)

- A type defining the content model for a generic message header that is refined by its derived classes.

**inReplyTo** (exactly one occurrence; of the type MessageId) A copy of the unique message identifier (within its own coding scheme) to which this message is responding.

**sentBy** (exactly one occurrence; of the type MessageAddress) The unique identifier (within its coding scheme) for the originator of a message instance.

**sendTo** (zero or more occurrences; of the type MessageAddress) A unique identifier (within its coding scheme) indicating an intended recipient of a message.

**copyTo** (zero or more occurrences; of the type MessageAddress) A unique identifier (within the specified coding scheme) giving the details of some party to whom a copy of this message will be sent for reference.

**creationTimestamp** (exactly one occurrence; of the type xsd:dateTime) The date and time (on the source system) when this message instance was created.

**expiryTimestamp** (zero or one occurrence; of the type xsd:dateTime) The date and time (on the source system) when this message instance will be considered expired.

**partyMessageInformation** (zero or more occurrences; of the type PartyMessageInformation) Additional message information that may be provided by each involved party.

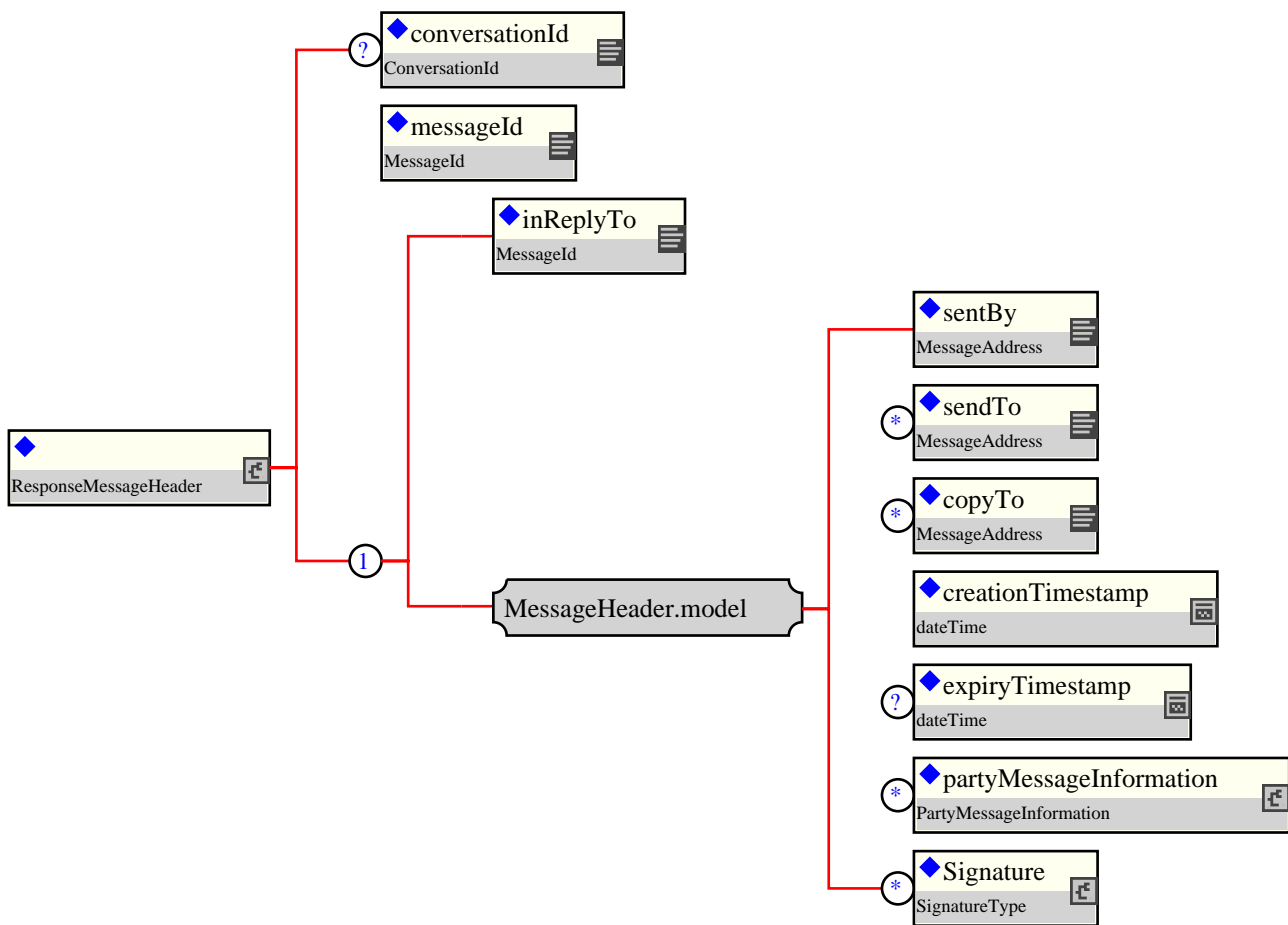
**dsig:Signature** (zero or more occurrences;

### 1.18.3 Used by:

- Complex type: ResponseMessage

### 1.18.4 Derived Types:

### 1.18.5 Figure:



### 1.18.6 Schema Fragment:

```

<xsd:complexType name="ResponseMessageHeader">
  <xsd:annotation>
    <xsd:documentation xml:lang="en">
      A type refining the generic message header to make it specific to
      response messages.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:complexContent>
    <xsd:extension base="MessageHeader">
      <xsd:sequence>
        <xsd:element name="inReplyTo" type="MessageId">
          <xsd:annotation>
            <xsd:documentation xml:lang="en">
              A copy of the unique message identifier (within its own
              coding scheme) to which this message is responding.
            </xsd:documentation>
          </xsd:annotation>
        </xsd:element>
        <xsd:group ref="MessageHeader.model"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

## 1.19 TradeAlreadyCancelled

### 1.19.1 Description:

An error response message indicating that a trade has already been cancelled.

### 1.19.2 Contents:

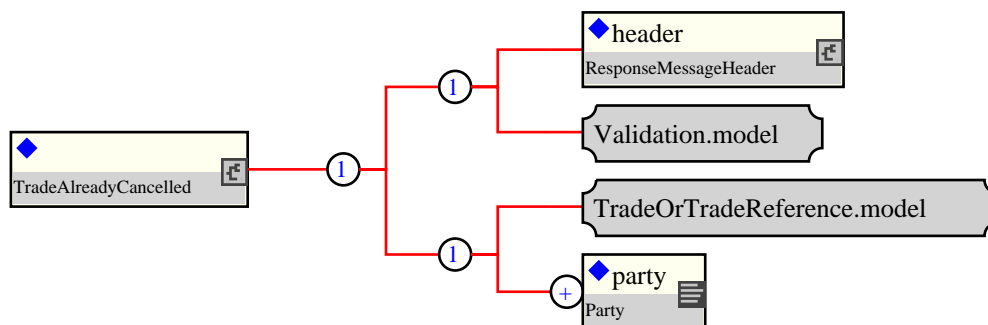
Inherited element(s): (This definition inherits the content defined by the type TradeErrorResponse)

- An abstract trade error response message containing a single trade or trade reference.

### 1.19.3 Used by:

### 1.19.4 Derived Types:

### 1.19.5 Figure:



### 1.19.6 Schema Fragment:

```
<xsd:complexType name="TradeAlreadyCancelled">
  <xsd:annotation>
    <xsd:documentation xml:lang="en">
      An error response message indicating that a trade has already
      been cancelled.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:complexContent>
    <xsd:extension base="TradeErrorResponse" />
  </xsd:complexContent>
</xsd:complexType>
```

## 1.20 TradeAlreadySubmitted

### 1.20.1 Description:

A type defining the content model for a message sent by a confirmation provider when it believes that one party has repeated a request to confirm a trade.

### 1.20.2 Contents:

Inherited element(s): (This definition inherits the content defined by the type ResponseMessage)

- A type refining the generic message content model to make it specific to response messages.

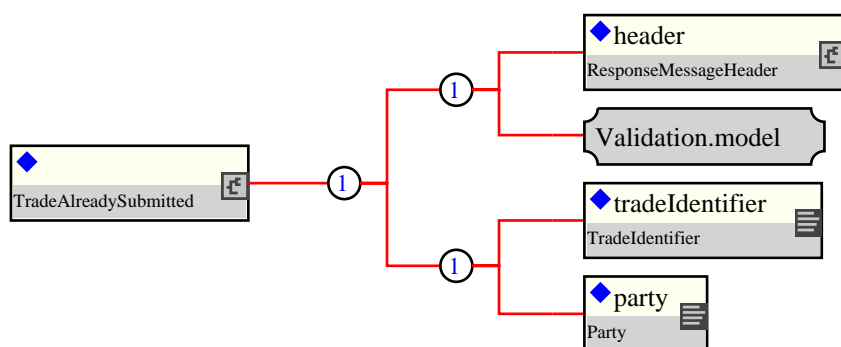
**tradeIdentifier** (exactly one occurrence; of the type TradeIdentifier) An instance of a unique trade identifier.

**party** (exactly one occurrence; of the type Party) A legal entity or a subdivision of a legal entity.

### 1.20.3 Used by:

### 1.20.4 Derived Types:

### 1.20.5 Figure:



### 1.20.6 Schema Fragment:

```
<xsd:complexType name="TradeAlreadySubmitted">
  <xsd:annotation>
    <xsd:documentation xml:lang="en">
      A type defining the content model for a message sent by a
      confirmation provider when it believes that one party has
      repeated a request to confirm a trade.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:complexContent>
    <xsd:extension base="ResponseMessage">
      <xsd:sequence>
        <xsd:element name="tradeIdentifier" type="TradeIdentifier">
          <xsd:annotation>
            <xsd:documentation xml:lang="en">
              An instance of a unique trade identifier.
            </xsd:documentation>
          </xsd:annotation>
        </xsd:element>
        <xsd:element name="party" type="Party">
          <xsd:annotation>
            <xsd:documentation xml:lang="en">
              A legal entity or a subdivision of a legal entity.
            </xsd:documentation>
            <xsd:documentation xml:lang="en">
              Parties can perform multiple roles in a trade lifecycle.
              For example, the principal parties obligated to make
              payments from time to time during the term of the trade,
            </xsd:documentation>
          </xsd:annotation>
        </xsd:element>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

but may include other parties involved in, or incidental to, the trade, such as parties acting in the role of novation transferor/transferee, broker, calculation agent, etc. In FpML roles are defined in multiple places within a document.

```
</xsd:documentation>
</xsd:annotation>
</xsd:element>
</xsd:sequence>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>
```

## 1.21 TradeAlreadyTerminated

### 1.21.1 Description:

An error response message indicating that a trade has already been terminated.

### 1.21.2 Contents:

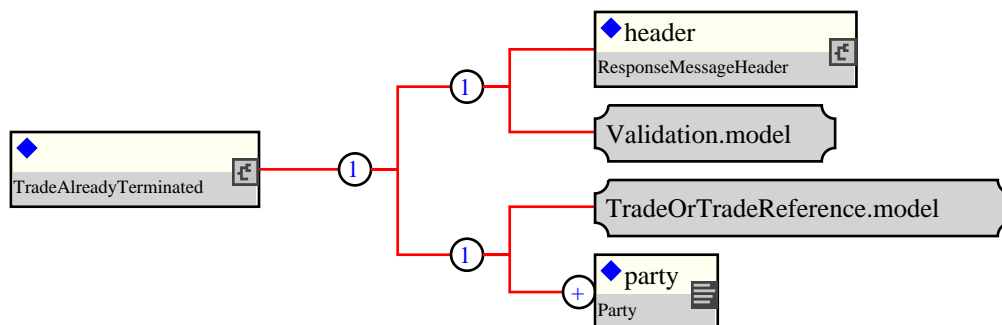
Inherited element(s): (This definition inherits the content defined by the type TradeErrorResponse)

- An abstract trade error response message containing a single trade or trade reference.

### 1.21.3 Used by:

### 1.21.4 Derived Types:

### 1.21.5 Figure:



### 1.21.6 Schema Fragment:

```
<xsd:complexType name="TradeAlreadyTerminated">
  <xsd:annotation>
    <xsd:documentation xml:lang="en">
      An error response message indicating that a trade has already
      been terminated.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:complexContent>
    <xsd:extension base="TradeErrorResponse" />
  </xsd:complexContent>
</xsd:complexType>
```

## 1.22 TradeErrorResponse

### 1.22.1 Description:

An abstract trade error response message containing a single trade or trade reference.

### 1.22.2 Contents:

Inherited element(s): (This definition inherits the content defined by the type ResponseMessage)

- A type refining the generic message content model to make it specific to response messages.

Either

**trade** (exactly one occurrence; of the type Trade) 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

Or

**tradeReference** (exactly one occurrence; of the type PartyTradeIdentifiers) A container since an individual trade can be referenced by two or more different partyTradeIdentifier elements - each allocated by a different party.

**party** (one or more occurrences; of the type Party) A legal entity or a subdivision of a legal entity.

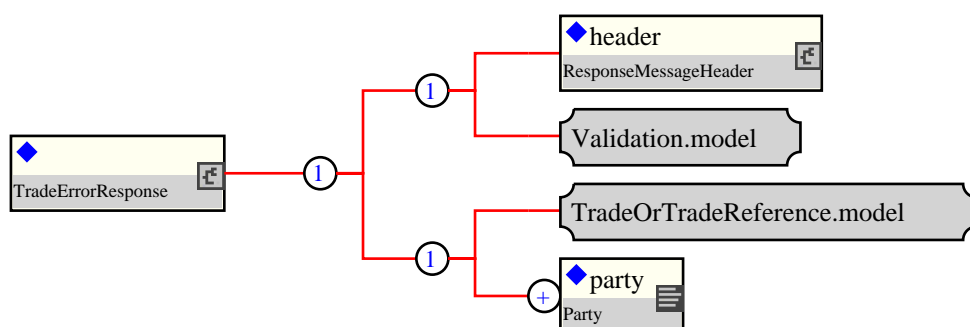
### 1.22.3 Used by:

- Complex type: TradeAlreadyAffirmed
- Complex type: TradeAlreadyCancelled
- Complex type: TradeAlreadyConfirmed
- Complex type: TradeAlreadyTerminated

### 1.22.4 Derived Types:

- Complex type: TradeAlreadyAffirmed
- Complex type: TradeAlreadyCancelled
- Complex type: TradeAlreadyConfirmed
- Complex type: TradeAlreadyTerminated

### 1.22.5 Figure:



### 1.22.6 Schema Fragment:

```
<xsd:complexType name="TradeErrorResponse" abstract="true">
  <xsd:annotation>
    <xsd:documentation xml:lang="en">
      An abstract trade error response message containing a single trade
      or trade reference.
    </xsd:documentation>
  </xsd:annotation>
</xsd:complexType>
```



```
</xsd:annotation>
<xsd:complexContent>
  <xsd:extension base="ResponseMessage">
    <xsd:sequence>
      <xsd:group ref="TradeOrTradeReference.model"/>
      <xsd:element name="party" type="Party" maxOccurs="unbounded">
        <xsd:annotation>
          <xsd:documentation xml:lang="en">
            A legal entity or a subdivision of a legal entity.
          </xsd:documentation>
          <xsd:documentation xml:lang="en">
            Parties can perform multiple roles in a trade lifecycle.
            For example, the principal parties obligated to make
            payments from time to time during the term of the trade,
            but may include other parties involved in, or incidental
            to, the trade, such as parties acting in the role of
            novation transferor/transferee, broker, calculation
            agent, etc. In FpML roles are defined in multiple places
            within a document.
          </xsd:documentation>
        </xsd:annotation>
      </xsd:element>
    </xsd:sequence>
  </xsd:extension>
</xsd:complexContent>
</xsd:complexType>
```

## 1.23 TradeNotFound

### 1.23.1 Description:

A type defining the content model of a response message generated when an operation as requested on a trade unknown to the service.

### 1.23.2 Contents:

Inherited element(s): (This definition inherits the content defined by the type ResponseMessage)

- A type refining the generic message content model to make it specific to response messages.

Either

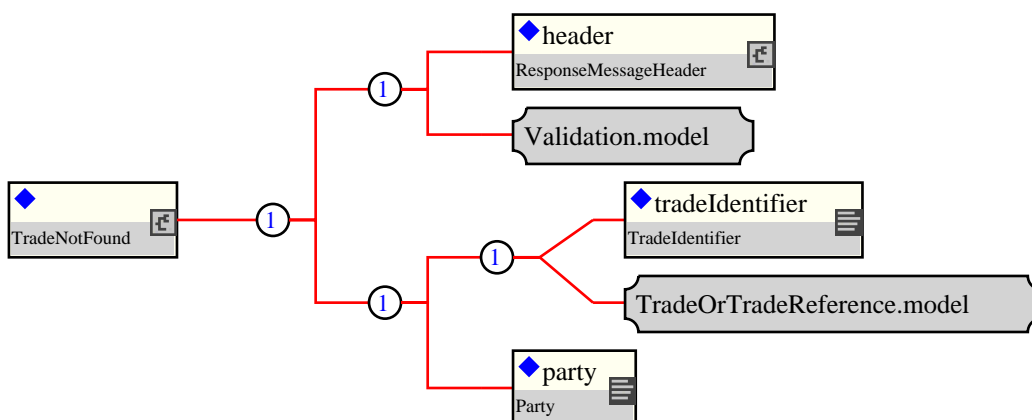
**tradeIdentifier** (exactly one occurrence; of the type TradeIdentifier) An instance of a unique trade identifier.

**party** (exactly one occurrence; of the type Party) A legal entity or a subdivision of a legal entity.

### 1.23.3 Used by:

### 1.23.4 Derived Types:

### 1.23.5 Figure:



### 1.23.6 Schema Fragment:

```
<xsd:complexType name="TradeNotFound">
  <xsd:annotation>
    <xsd:documentation xml:lang="en">
      A type defining the content model of a response message generated
      when an operation as requested on a trade unknown to the service.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:complexContent>
    <xsd:extension base="ResponseMessage">
      <xsd:sequence>
        <xsd:choice>
          <xsd:element name="tradeIdentifier" type="TradeIdentifier">
            <xsd:annotation>
              <xsd:documentation xml:lang="en">
                An instance of a unique trade identifier.
              </xsd:documentation>
            </xsd:annotation>
          </xsd:element>
          <xsd:group ref="TradeOrTradeReference.model"/>
        </xsd:choice>
        <xsd:element name="party" type="Party">

```

```
<xsd:annotation>
  <xsd:documentation xml:lang="en">
    A legal entity or a subdivision of a legal entity.
  </xsd:documentation>
  <xsd:documentation xml:lang="en">
    Parties can perform multiple roles in a trade lifecycle.
    For example, the principal parties obligated to make
    payments from time to time during the term of the trade,
    but may include other parties involved in, or incidental
    to, the trade, such as parties acting in the role of
    novation transferor/transferee, broker, calculation
    agent, etc. In FpML roles are defined in multiple places
    within a document.
  </xsd:documentation>
</xsd:annotation>
</xsd:element>
</xsd:sequence>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>
```

## 1.24 TradeStatus

### 1.24.1 Description:

A type defining the content model for a message normally generated in response to a RequestTradeStatus request.

### 1.24.2 Contents:

Inherited element(s): (This definition inherits the content defined by the type ResponseMessage)

- A type refining the generic message content model to make it specific to response messages.

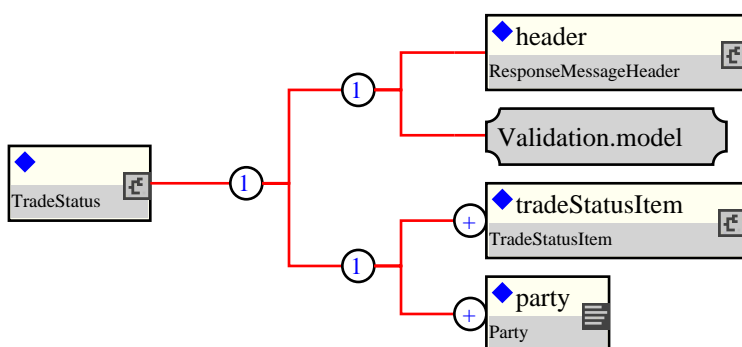
**tradeStatusItem** (one or more occurrences; of the type TradeStatusItem) A collection of data values describing the state of the given trade.

**party** (one or more occurrences; of the type Party) A legal entity or a subdivision of a legal entity.

### 1.24.3 Used by:

### 1.24.4 Derived Types:

### 1.24.5 Figure:



### 1.24.6 Schema Fragment:

```
<xsd:complexType name="TradeStatus">
  <xsd:annotation>
    <xsd:documentation xml:lang="en">
      A type defining the content model for a message normally
      generated in response to a RequestTradeStatus request.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:complexContent>
    <xsd:extension base="ResponseMessage">
      <xsd:sequence>
        <xsd:element name="tradeStatusItem" type="TradeStatusItem" maxOccurs="unbounded">
          <xsd:annotation>
            <xsd:documentation xml:lang="en">
              A collection of data values describing the state of the
              given trade.
            </xsd:documentation>
          </xsd:annotation>
        </xsd:element>
        <xsd:element name="party" type="Party" maxOccurs="unbounded">
          <xsd:annotation>
            <xsd:documentation xml:lang="en">
              A legal entity or a subdivision of a legal entity.
            </xsd:documentation>
            <xsd:documentation xml:lang="en">
              Parties can perform multiple roles in a trade lifecycle.
              For example, the principal parties obligated to make
```

payments from time to time during the term of the trade, but may include other parties involved in, or incidental to, the trade, such as parties acting in the role of novation transferor/transferee, broker, calculation agent, etc. In FpML roles are defined in multiple places within a document.

```
</xsd:documentation>
</xsd:annotation>
</xsd:element>
</xsd:sequence>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>
```

## 1.25 TradeStatusItem

### 1.25.1 Description:

A type used in trade status enquiry messages which relates a trade identifier to its current status value.

### 1.25.2 Contents:

**tradeIdentifier** (exactly one occurrence; of the type TradeIdentifier) An instance of a unique trade identifier.

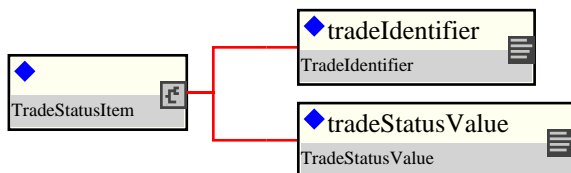
**tradeStatusValue** (exactly one occurrence; of the type TradeStatusValue) The trade status value.

### 1.25.3 Used by:

- Complex type: TradeStatus

### 1.25.4 Derived Types:

### 1.25.5 Figure:



### 1.25.6 Schema Fragment:

```
<xsd:complexType name="TradeStatusItem">
  <xsd:annotation>
    <xsd:documentation xml:lang="en">
      A type used in trade status enquiry messages which relates a
      trade identifier to its current status value.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:sequence>
    <xsd:element name="tradeIdentifier" type="TradeIdentifier">
      <xsd:annotation>
        <xsd:documentation xml:lang="en">
          An instance of a unique trade identifier.
        </xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="tradeStatusValue" type="TradeStatusValue">
      <xsd:annotation>
        <xsd:documentation xml:lang="en">
          The trade status value.
        </xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>
```

## 1.26 TradeStatusValue

### 1.26.1 Description:

The type used to hold TradeStatusScheme values.

### 1.26.2 Contents:

Inherited element(s): (This definition inherits the content defined by the type xsd:normalizedString)

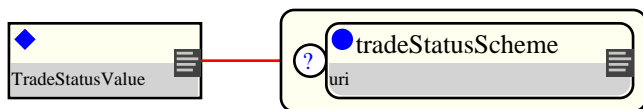
•

### 1.26.3 Used by:

- Complex type: TradeStatusItem

### 1.26.4 Derived Types:

### 1.26.5 Figure:



### 1.26.6 Schema Fragment:

```
<xsd:complexType name="TradeStatusValue">
  <xsd:annotation>
    <xsd:documentation xml:lang="en">
      The type used to hold TradeStatusScheme values.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:simpleContent>
    <xsd:extension base="xsd:normalizedString">
      <xsd:attribute name="tradeStatusScheme" type="xsd:anyURI" />
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
```

## ***2 Groups***



## 2.1 Exception.model

### 2.1.1 Description:

A model group which has exception elements.

### 2.1.2 Contents:

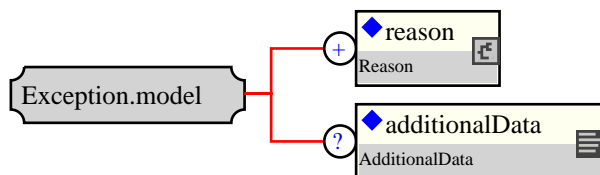
**reason** (one or more occurrences; of the type Reason) An instance of the Reason type used to record the nature of any errors associated with a message.

**additionalData** (zero or one occurrence; of the type AdditionalData) Any string of additional data that may help the message processor, for example in a rejection message this might contain a code value or the text of the original request (within a CDATA section).

### 2.1.3 Used by:

- Complex type: MessageRejected

### 2.1.4 Figure:



### 2.1.5 Schema Fragment:

```
<xsd:group name="Exception.model">
  <xsd:annotation>
    <xsd:documentation xml:lang="en">
      A model group which has exception elements.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:sequence>
    <xsd:element name="reason" type="Reason" maxOccurs="unbounded">
      <xsd:annotation>
        <xsd:documentation xml:lang="en">
          An instance of the Reason type used to record the nature of
          any errors associated with a message.
        </xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="additionalData" type="AdditionalData" minOccurs="0">
      <xsd:annotation>
        <xsd:documentation xml:lang="en">
          Any string of additional data that may help the message
          processor, for example in a rejection message this might
          contain a code value or the text of the original request
          (within a CDATA section).
        </xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
</xsd:group>
```

## 2.2 MessageHeader.model

### 2.2.1 Description:

Defines the structure that contains routing and identification information, which allows processing and transfer of the message. It contains only messaging information that is applicable to all messages. If the information is not message related or is not applicable to all messages then it is not defined in the message header.

### 2.2.2 Contents:

**sentBy** (exactly one occurrence; of the type MessageAddress) The unique identifier (within its coding scheme) for the originator of a message instance.

**sendTo** (zero or more occurrences; of the type MessageAddress) A unique identifier (within its coding scheme) indicating an intended recipient of a message.

**copyTo** (zero or more occurrences; of the type MessageAddress) A unique identifier (within the specified coding scheme) giving the details of some party to whom a copy of this message will be sent for reference.

**creationTimestamp** (exactly one occurrence; of the type xsd:dateTime) The date and time (on the source system) when this message instance was created.

**expiryTimestamp** (zero or one occurrence; of the type xsd:dateTime) The date and time (on the source system) when this message instance will be considered expired.

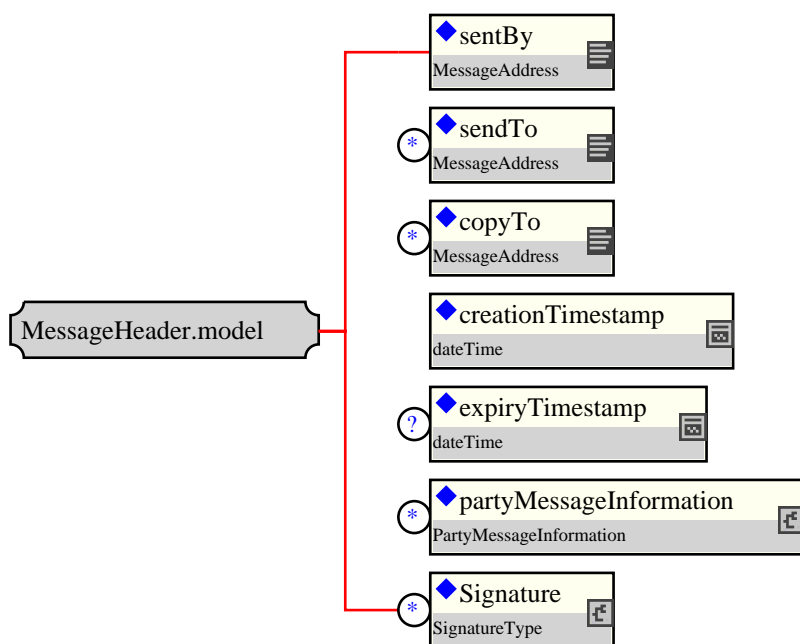
**partyMessageInformation** (zero or more occurrences; of the type PartyMessageInformation) Additional message information that may be provided by each involved party.

**dsig:Signature** (zero or more occurrences;

### 2.2.3 Used by:

- Complex type: NotificationMessageHeader
- Complex type: RequestMessageHeader
- Complex type: ResponseMessageHeader

### 2.2.4 Figure:



### 2.2.5 Schema Fragment:

```

<xsd:group name="MessageHeader.model">
  <xsd:annotation>
    <xsd:documentation xml:lang="en">
      Defines the structure that contains routing and identification
      information, which allows processing and transfer of the message.
      It contains only messaging information that is applicable to all
      messages. If the information is not message related or is not
      applicable to all messages then it is not defined in the message
      header.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:sequence>
    <xsd:element name="sentBy" type="MessageAddress">
      <xsd:annotation>
        <xsd:documentation xml:lang="en">
          The unique identifier (within its coding scheme) for the
          originator of a message instance.
        </xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="sendTo" type="MessageAddress" minOccurs="0" maxOccurs="unbounded">
      <xsd:annotation>
        <xsd:documentation xml:lang="en">
          A unique identifier (within its coding scheme) indicating an
          intended recipient of a message.
        </xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="copyTo" type="MessageAddress" minOccurs="0" maxOccurs="unbounded">
      <xsd:annotation>
        <xsd:documentation xml:lang="en">
          A unique identifier (within the specified coding scheme)
          giving the details of some party to whom a copy of this
          message will be sent for reference.
        </xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="creationTimestamp" type="xsd:dateTime">
      <xsd:annotation>
        <xsd:documentation xml:lang="en">
          The date and time (on the source system) when this message
          instance was created.
        </xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="expiryTimestamp" type="xsd:dateTime" minOccurs="0">
      <xsd:annotation>
        <xsd:documentation xml:lang="en">
          The date and time (on the source system) when this message
          instance will be considered expired.
        </xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="partyMessageInformation" type="PartyMessageInformation" minOccurs="0" maxOccurs="unbounded">
      <xsd:annotation>
        <xsd:documentation xml:lang="en">
          Additional message information that may be provided by each
          involved party.
        </xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element ref="dsig:Signature" minOccurs="0" maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:group>

```

### 3 Schema listing

```
<xsd:schema ecore:nsPrefix="fpml" ecore:package="org.fpml" ecore:documentRoot="FpML" targetNameSpace="http://www.fpml.org" >
  <xsd:import namespace="http://www.w3.org/2000/09/xmldsig#" schemaLocation="xmldsig-core-schema.xsd"/>
  <xsd:include schemaLocation="fpml-doc-4-3.xsd"/>
  <xsd:complexType name="AdditionalData">
    <xsd:annotation>
      <xsd:documentation xml:lang="en">
        Provides extra information not represented in the model that
        may be useful in processing the message i.e. diagnosing the
        reason for failure. In case the extra information is in XML
        format, a CDATA section must be placed around the source
        message to prevent its interpretation as XML content.
      </xsd:documentation>
    </xsd:annotation>
    <xsd:simpleContent>
      <xsd:extension base="xsd:string">
        <xsd:attribute name="additionalDataScheme" type="xsd:anyURI" use="optional"/>
      </xsd:extension>
    </xsd:simpleContent>
  </xsd:complexType>
  <xsd:complexType name="ConversationId">
    <xsd:annotation>
      <xsd:documentation xml:lang="en">
        The unique identifier (name) for the conversation (session),
        this message is within. A conversation identifier is usually
        assigned by the initiator of a conversation. Conversations may
        only be initiated and terminated. Joining conversations has the
        effect of initiating new conversations. Conversations cannot be
        split; this instead has the effect of parallel activities on
        the same conversation or the initiation of a new conversation.
        Each message belongs to only one conversation. Conversation
        scopes are defined in the business process definition.
      </xsd:documentation>
    </xsd:annotation>
    <xsd:simpleContent>
      <xsd:extension base="xsd:normalizedString">
        <xsd:attribute name="conversationIdScheme" type="xsd:anyURI" use="required"/>
      </xsd:extension>
    </xsd:simpleContent>
  </xsd:complexType>
  <xsd:complexType name="Message" abstract="true">
    <xsd:annotation>
      <xsd:documentation xml:lang="en">
        A type defining the basic structure of all FpML messages which
        is refined by its derived types.
      </xsd:documentation>
    </xsd:annotation>
    <xsd:complexContent>
      <xsd:extension base="Document"/>
    </xsd:complexContent>
  </xsd:complexType>
  <xsd:complexType name="MessageAddress">
    <xsd:annotation>
      <xsd:documentation xml:lang="en">
        The data type used for identifying a message address. It
        includes a partyIdScheme for keeping the type backward
        compatible with the PartyId complex type. In the next major
        version, partyIdScheme attribute will be replaced by a
        messageAddressScheme attribute.
      </xsd:documentation>
    </xsd:annotation>
    <xsd:simpleContent>
      <xsd:extension base="xsd:normalizedString">
        <xsd:attribute name="partyIdScheme" type="xsd:anyURI" default="http://www.fpml.org/extn/partyIdScheme"/>
      </xsd:extension>
    </xsd:simpleContent>
  </xsd:complexType>
  <xsd:complexType name="MessageHeader" abstract="true">
    <xsd:annotation>
      <xsd:documentation xml:lang="en">
        A type defining the content model for a generic message header
        that is refined by its derived classes.
      </xsd:documentation>
    </xsd:annotation>
    <xsd:sequence>
      <xsd:element name="conversationId" type="ConversationId" minOccurs="0">
        <xsd:annotation>
          <xsd:documentation xml:lang="en">
            The unique identifier (name) for the conversation
            (session), this message is within. A conversation

```

identifier is usually assigned by the initiator of a conversation. Conversations may only be initiated and terminated. Joining conversations has the effect of initiating new conversations. Conversations cannot be split; this instead has the effect of parallel activities on the same conversation or the initiation of a new conversation. Each message belongs to only one conversation. Conversation scopes are defined in the business process definition.

```
</xsd:documentation>
</xsd:annotation>
</xsd:element>
<xsd:element name="messageId" type="MessageId">
  <xsd:annotation>
    <xsd:documentation xml:lang="en">
      A unique identifier (within its coding scheme) assigned to
      the message by its creating party.
    </xsd:documentation>
  </xsd:annotation>
</xsd:element>
</xsd:sequence>
</xsd:complexType>
<xsd:complexType name="MessageId">
  <xsd:annotation>
    <xsd:documentation xml:lang="en">
      The data type use for message identifiers.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:simpleContent>
    <xsd:extension base="xsd:normalizedString">
      <xsd:attribute name="messageIdScheme" type="xsd:anyURI" use="required"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
<xsd:complexType name="MessageRejected">
  <xsd:annotation>
    <xsd:documentation xml:lang="en">
      A type defining the content for a standard message sent when a
      recipient cannot interpret or process an earlier message.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:complexContent>
    <xsd:extension base="NotificationMessage">
      <xsd:sequence>
        <xsd:group ref="Exception.model"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="NotificationMessage" abstract="true">
  <xsd:annotation>
    <xsd:documentation xml:lang="en">
      A type defining the basic content for a message sent to inform
      another system that some 'business event' has occurred.
      Notifications are not expected to be replied to.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:complexContent>
    <xsd:extension base="Message">
      <xsd:sequence>
        <xsd:element name="header" type="NotificationMessageHeader"/>
        <xsd:group ref="Validation.model"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="NotificationMessageHeader">
  <xsd:annotation>
    <xsd:documentation xml:lang="en">
      A type that refines the generic message header to match the
      requirements of a NotificationMessage.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:complexContent>
    <xsd:extension base="MessageHeader">
      <xsd:sequence>
        <xsd:element name="inReplyTo" type="MessageId" minOccurs="0">
          <xsd:annotation>
            <xsd:documentation xml:lang="en">
              A copy of the unique message identifier (within its own
              coding scheme) to which this message is responding.
            </xsd:documentation>
          </xsd:annotation>
        </xsd:element>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

```

        </xsd:element>
        <xsd:group ref="MessageHeader.model"/>
    </xsd:sequence>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="PartyMessageInformation">
    <xsd:annotation>
        <xsd:documentation xml:lang="en">
            A type defining additional information that may be recorded
            against a message.
        </xsd:documentation>
    </xsd:annotation>
    <xsd:sequence>
        <xsd:element name="partyReference" type="PartyReference">
            <xsd:annotation>
                <xsd:documentation xml:lang="en">
                    Identifies that party that has ownership of this
                    information.
                </xsd:documentation>
            </xsd:annotation>
        </xsd:element>
    </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="ProblemLocation">
    <xsd:annotation>
        <xsd:documentation xml:lang="en">
            Provides a lexical location (i.e. a line number and character
            for bad XML) or an XPath location (i.e. place to identify the
            bad location for valid XML).
        </xsd:documentation>
    </xsd:annotation>
    <xsd:simpleContent>
        <xsd:extension base="xsd:normalizedString">
            <xsd:attribute name="locationType" type="xsd:token">
                <xsd:annotation>
                    <xsd:documentation xml:lang="en">
                        The value of the locationType attribute defines which
                        type of location has been given. It may take the values
                        'lexical' or 'xpath'.
                    </xsd:documentation>
                </xsd:annotation>
            </xsd:attribute>
            <xsd:attribute name="problemLocationScheme" type="xsd:anyURI" fpml-annotation:deprecate
                <xsd:annotation>
                    <xsd:documentation xml:lang="en">
                        DEPRECATED. It will be removed in FpML 5.0. New
                        implementations are encouraged to use the locationType
                        attribute.
                    </xsd:documentation>
                </xsd:annotation>
            </xsd:attribute>
        </xsd:extension>
    </xsd:simpleContent>
</xsd:complexType>
<xsd:complexType name="Reason">
    <xsd:annotation>
        <xsd:documentation xml:lang="en">
            A type defining a content model for describing the nature and
            possible location of a error within a previous message.
        </xsd:documentation>
    </xsd:annotation>
    <xsd:sequence>
        <xsd:element name="reasonCode" type="ReasonCode">
            <xsd:annotation>
                <xsd:documentation xml:lang="en">
                    A machine interpretable error code.
                </xsd:documentation>
            </xsd:annotation>
        </xsd:element>
        <xsd:element name="location" type="ProblemLocation" minOccurs="0">
            <xsd:annotation>
                <xsd:documentation xml:lang="en">
                    A value indicating the location of the problem within the
                    subject message.
                </xsd:documentation>
            </xsd:annotation>
        </xsd:element>
        <xsd:element name="description" type="xsd:string" minOccurs="0">
            <xsd:annotation>
                <xsd:documentation xml:lang="en">
                    Plain English text describing the associated error
                    condition

```

```

        </xsd:documentation>
    </xsd:annotation>
</xsd:element>
<xsd:element name="validationRuleId" type="Validation" minOccurs="0">
    <xsd:annotation>
        <xsd:documentation xml:lang="en">
            A reference identifying a rule within a validation scheme
        </xsd:documentation>
    </xsd:annotation>
</xsd:element>
<xsd:element name="additionalData" type="AdditionalData" minOccurs="0" maxOccurs="unbounded">
    <xsd:annotation>
        <xsd:documentation xml:lang="en">
            Any string of additional data that may help the message
            processor, for example in a rejection message this might
            contain a code value or the text of any one of the messages
            (within a CDATA section).
        </xsd:documentation>
    </xsd:annotation>
</xsd:element>
</xsd:sequence>
</xsd:complexType>
<xsd:complexType name="ReasonCode">
    <xsd:annotation>
        <xsd:documentation xml:lang="en">
            Defines a list of machine interpretable error codes.
        </xsd:documentation>
    </xsd:annotation>
    <xsd:simpleContent>
        <xsd:extension base="xsd:normalizedString">
            <xsd:attribute name="reasonCodeScheme" type="xsd:anyURI" default="http://www.fpml.org/c
            </xsd:extension>
        </xsd:simpleContent>
    </xsd:complexType>
<xsd:complexType name="RequestMessage" abstract="true">
    <xsd:annotation>
        <xsd:documentation xml:lang="en">
            A type defining the basic content of a message that requests
            the receiver to perform some business operation determined by
            the message type and its content.
        </xsd:documentation>
    </xsd:annotation>
    <xsd:complexContent>
        <xsd:extension base="Message">
            <xsd:sequence>
                <xsd:element name="header" type="RequestMessageHeader"/>
                <xsd:group ref="Validation.model"/>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="RequestMessageHeader">
    <xsd:annotation>
        <xsd:documentation xml:lang="en">
            A type refining the generic message header content to make it
            specific to request messages.
        </xsd:documentation>
    </xsd:annotation>
    <xsd:complexContent>
        <xsd:extension base="MessageHeader">
            <xsd:sequence>
                <xsd:group ref="MessageHeader.model"/>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="RequestTradeStatus">
    <xsd:annotation>
        <xsd:documentation xml:lang="en">
            A type defining the content model for a message allowing one
            party to query the status of one or many trades previously sent
            to another party.
        </xsd:documentation>
    </xsd:annotation>
    <xsd:complexContent>
        <xsd:extension base="RequestMessage">
            <xsd:sequence>
                <xsd:element name="tradeIdentifier" type="TradeIdentifier" maxOccurs="unbounded">
                    <xsd:annotation>
                        <xsd:documentation xml:lang="en">
                            An instance of a unique trade identifier.
                        </xsd:documentation>
                    </xsd:annotation>
                </xsd:sequence>
            </xsd:extension>
        </xsd:extension>
    </xsd:complexContent>

```

```

</xsd:element>
<xsd:element name="party" type="Party" maxOccurs="unbounded">
  <xsd:annotation>
    <xsd:documentation xml:lang="en">
      A legal entity or a subdivision of a legal entity.
    </xsd:documentation>
    <xsd:documentation xml:lang="en">
      Parties can perform multiple roles in a trade
      lifecycle. For example, the principal parties obligated
      to make payments from time to time during the term of
      the trade, but may include other parties involved in,
      or incidental to, the trade, such as parties acting in
      the role of novation transferor/transferee, broker,
      calculation agent, etc. In FpML roles are defined in
      multiple places within a document.
    </xsd:documentation>
  </xsd:annotation>
</xsd:element>
</xsd:sequence>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="ResponseMessage" abstract="true">
  <xsd:annotation>
    <xsd:documentation xml:lang="en">
      A type refining the generic message content model to make it
      specific to response messages.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:complexContent>
    <xsd:extension base="Message">
      <xsd:sequence>
        <xsd:element name="header" type="ResponseMessageHeader"/>
        <xsd:group ref="Validation.model"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="ResponseMessageHeader">
  <xsd:annotation>
    <xsd:documentation xml:lang="en">
      A type refining the generic message header to make it specific
      to response messages.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:complexContent>
    <xsd:extension base="MessageHeader">
      <xsd:sequence>
        <xsd:element name="inReplyTo" type="MessageId">
          <xsd:annotation>
            <xsd:documentation xml:lang="en">
              A copy of the unique message identifier (within its own
              coding scheme) to which this message is responding.
            </xsd:documentation>
          </xsd:annotation>
        </xsd:element>
        <xsd:group ref="MessageHeader.model"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="TradeNotFound">
  <xsd:annotation>
    <xsd:documentation xml:lang="en">
      A type defining the content model of a response message
      generated when an operation as requested on a trade unknown to
      the service.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:complexContent>
    <xsd:extension base="ResponseMessage">
      <xsd:sequence>
        <xsd:choice>
          <xsd:element name="tradeIdentifier" type="TradeIdentifier">
            <xsd:annotation>
              <xsd:documentation xml:lang="en">
                An instance of a unique trade identifier.
              </xsd:documentation>
            </xsd:annotation>
          </xsd:element>
          <xsd:group ref="TradeOrTradeReference.model"/>
        </xsd:choice>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
  <xsd:element name="party" type="Party">

```



```

<xsd:annotation>
  <xsd:documentation xml:lang="en">
    A legal entity or a subdivision of a legal entity.
  </xsd:documentation>
  <xsd:documentation xml:lang="en">
    Parties can perform multiple roles in a trade
    lifecycle. For example, the principal parties obligated
    to make payments from time to time during the term of
    the trade, but may include other parties involved in,
    or incidental to, the trade, such as parties acting in
    the role of novation transferor/transferee, broker,
    calculation agent, etc. In FpML roles are defined in
    multiple places within a document.
  </xsd:documentation>
</xsd:annotation>
</xsd:element>
</xsd:sequence>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="TradeStatus">
  <xsd:annotation>
    <xsd:documentation xml:lang="en">
      A type defining the content model for a message normally
      generated in response to a RequestTradeStatus request.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:complexContent>
    <xsd:extension base="ResponseMessage">
      <xsd:sequence>
        <xsd:element name="tradeStatusItem" type="TradeStatusItem" maxOccurs="unbounded">
          <xsd:annotation>
            <xsd:documentation xml:lang="en">
              A collection of data values describing the state of the
              given trade.
            </xsd:documentation>
          </xsd:annotation>
        </xsd:element>
        <xsd:element name="party" type="Party" maxOccurs="unbounded">
          <xsd:annotation>
            <xsd:documentation xml:lang="en">
              A legal entity or a subdivision of a legal entity.
            </xsd:documentation>
            <xsd:documentation xml:lang="en">
              Parties can perform multiple roles in a trade
              lifecycle. For example, the principal parties obligated
              to make payments from time to time during the term of
              the trade, but may include other parties involved in,
              or incidental to, the trade, such as parties acting in
              the role of novation transferor/transferee, broker,
              calculation agent, etc. In FpML roles are defined in
              multiple places within a document.
            </xsd:documentation>
          </xsd:annotation>
        </xsd:element>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="TradeStatusItem">
  <xsd:annotation>
    <xsd:documentation xml:lang="en">
      A type used in trade status enquiry messages which relates a
      trade identifier to its current status value.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:sequence>
    <xsd:element name="tradeIdentifier" type="TradeIdentifier">
      <xsd:annotation>
        <xsd:documentation xml:lang="en">
          An instance of a unique trade identifier.
        </xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="tradeStatusValue" type="TradeStatusValue">
      <xsd:annotation>
        <xsd:documentation xml:lang="en">
          The trade status value.
        </xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>

```

```

<xsd:complexType name="TradeStatusValue">
  <xsd:annotation>
    <xsd:documentation xml:lang="en">
      The type used to hold TradeStatusScheme values.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:simpleContent>
    <xsd:extension base="xsd:normalizedString">
      <xsd:attribute name="tradeStatusScheme" type="xsd:anyURI"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
<xsd:group name="Exception.model">
  <xsd:annotation>
    <xsd:documentation xml:lang="en">
      A model group which has exception elements.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:sequence>
    <xsd:element name="reason" type="Reason" maxOccurs="unbounded">
      <xsd:annotation>
        <xsd:documentation xml:lang="en">
          An instance of the Reason type used to record the nature of
          any errors associated with a message.
        </xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="additionalData" type="AdditionalData" minOccurs="0">
      <xsd:annotation>
        <xsd:documentation xml:lang="en">
          Any string of additional data that may help the message
          processor, for example in a rejection message this might
          contain a code value or the text of the original request
          (within a CDATA section).
        </xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
</xsd:group>
<xsd:group name="MessageHeader.model">
  <xsd:annotation>
    <xsd:documentation xml:lang="en">
      Defines the structure that contains routing and identification
      information, which allows processing and transfer of the
      message. It contains only messaging information that is
      applicable to all messages. If the information is not message
      related or is not applicable to all messages then it is not
      defined in the message header.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:sequence>
    <xsd:element name="sentBy" type="MessageAddress">
      <xsd:annotation>
        <xsd:documentation xml:lang="en">
          The unique identifier (within its coding scheme) for the
          originator of a message instance.
        </xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="sendTo" type="MessageAddress" minOccurs="0" maxOccurs="unbounded">
      <xsd:annotation>
        <xsd:documentation xml:lang="en">
          A unique identifier (within its coding scheme) indicating
          an intended recipient of a message.
        </xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="copyTo" type="MessageAddress" minOccurs="0" maxOccurs="unbounded">
      <xsd:annotation>
        <xsd:documentation xml:lang="en">
          A unique identifier (within the specified coding scheme)
          giving the details of some party to whom a copy of this
          message will be sent for reference.
        </xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="creationTimestamp" type="xsd:dateTime">
      <xsd:annotation>
        <xsd:documentation xml:lang="en">
          The date and time (on the source system) when this message
          instance was created.
        </xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
</xsd:group>

```

```

</xsd:element>
<xsd:element name="expiryTimestamp" type="xsd:dateTime" minOccurs="0">
  <xsd:annotation>
    <xsd:documentation xml:lang="en">
      The date and time (on the source system) when this message
      instance will be considered expired.
    </xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:element name="partyMessageInformation" type="PartyMessageInformation" minOccurs="0">
  <xsd:annotation>
    <xsd:documentation xml:lang="en">
      Additional message information that may be provided by each
      involved party.
    </xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:element ref="dsig:Signature" minOccurs="0" maxOccurs="unbounded"/>
</xsd:sequence>
</xsd:group>
<xsd:complexType name="TradeErrorResponse" abstract="true">
  <xsd:annotation>
    <xsd:documentation xml:lang="en">
      An abstract trade error response message containing a single
      trade or trade reference.
    </xsd:documentation>
  </xsd:annotation>
<xsd:complexContent>
  <xsd:extension base="ResponseMessage">
    <xsd:sequence>
      <xsd:group ref="TradeOrTradeReference.model"/>
      <xsd:element name="party" type="Party" maxOccurs="unbounded">
        <xsd:annotation>
          <xsd:documentation xml:lang="en">
            A legal entity or a subdivision of a legal entity.
          </xsd:documentation>
          <xsd:documentation xml:lang="en">
            Parties can perform multiple roles in a trade
            lifecycle. For example, the principal parties obligated
            to make payments from time to time during the term of
            the trade, but may include other parties involved in,
            or incidental to, the trade, such as parties acting in
            the role of novation transferor/transferee, broker,
            calculation agent, etc. In FpML roles are defined in
            multiple places within a document.
          </xsd:documentation>
        </xsd:annotation>
      </xsd:element>
    </xsd:sequence>
  </xsd:extension>
</xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="TradeAlreadyCancelled">
  <xsd:annotation>
    <xsd:documentation xml:lang="en">
      An error response message indicating that a trade has already
      been cancelled.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:complexContent>
    <xsd:extension base="TradeErrorResponse"/>
  </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="TradeAlreadySubmitted">
  <xsd:annotation>
    <xsd:documentation xml:lang="en">
      A type defining the content model for a message sent by a
      confirmation provider when it believes that one party has
      repeated a request to confirm a trade.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:complexContent>
    <xsd:extension base="ResponseMessage">
      <xsd:sequence>
        <xsd:element name="tradeIdentifier" type="TradeIdentifier">
          <xsd:annotation>
            <xsd:documentation xml:lang="en">
              An instance of a unique trade identifier.
            </xsd:documentation>
          </xsd:annotation>
        </xsd:element>
        <xsd:element name="party" type="Party">
          <xsd:annotation>

```

```

<xsd:documentation xml:lang="en">
  A legal entity or a subdivision of a legal entity.
</xsd:documentation>
<xsd:documentation xml:lang="en">
  Parties can perform multiple roles in a trade
  lifecycle. For example, the principal parties obligated
  to make payments from time to time during the term of
  the trade, but may include other parties involved in,
  or incidental to, the trade, such as parties acting in
  the role of novation transferor/transferee, broker,
  calculation agent, etc. In FpML roles are defined in
  multiple places within a document.
</xsd:documentation>
</xsd:annotation>
</xsd:element>
</xsd:sequence>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="TradeAlreadyTerminated">
  <xsd:annotation>
    <xsd:documentation xml:lang="en">
      An error response message indicating that a trade has already
      been terminated.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:complexContent>
    <xsd:extension base="TradeErrorResponse"/>
  </xsd:complexContent>
</xsd:complexType>
</xsd:schema>

```