



**Financial products Markup Language**

## **FpML 4.2 - Messaging Component Definitions**

## ***Version: 4.2***

### **This Version:**

<http://www.fpml.org/spec/fpml-4-2-11-tr-5>

### **Latest Version:**

<http://www.fpml.org/spec/fpml-4-2-11-tr-5>

### **Previous Version:**

<https://www.fpml.org/spec/rec-fpml-4-2-2007-05-14/>

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

<http://www.fpml.org/spec/errata/fpml-4-2-11-tr-5-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/documents/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	7
1.1	AdditionalData	8
1.1.1	Description:	8
1.1.2	Contents:	8
1.1.3	Used by:	8
1.1.4	Derived Types:	8
1.1.5	Figure:	8
1.1.6	Schema Fragment:	8
1.2	ContractReferenceMessage	9
1.2.1	Description:	9
1.2.2	Contents:	9
1.2.3	Used by:	9
1.2.4	Derived Types:	9
1.2.5	Figure:	9
1.2.6	Schema Fragment:	9
1.3	ConversationId	10
1.3.1	Description:	10
1.3.2	Contents:	10
1.3.3	Used by:	10
1.3.4	Derived Types:	10
1.3.5	Figure:	10
1.3.6	Schema Fragment:	10
1.4	Message	11
1.4.1	Description:	11
1.4.2	Contents:	11
1.4.3	Used by:	11
1.4.4	Derived Types:	11
1.4.5	Figure:	11
1.4.6	Schema Fragment:	11
1.5	MessageAddress	12
1.5.1	Description:	12
1.5.2	Contents:	12
1.5.3	Used by:	12
1.5.4	Derived Types:	12
1.5.5	Figure:	12
1.5.6	Schema Fragment:	12
1.6	MessageHeader	13
1.6.1	Description:	13
1.6.2	Contents:	13
1.6.3	Used by:	13
1.6.4	Derived Types:	13
1.6.5	Figure:	13
1.6.6	Schema Fragment:	13
1.7	MessageId	14
1.7.1	Description:	14
1.7.2	Contents:	14
1.7.3	Used by:	14
1.7.4	Derived Types:	14
1.7.5	Figure:	14
1.7.6	Schema Fragment:	14
1.8	MessageRejected	15
1.8.1	Description:	15
1.8.2	Contents:	15
1.8.3	Used by:	15
1.8.4	Derived Types:	15
1.8.5	Figure:	15
1.8.6	Schema Fragment:	15
1.9	NotificationMessage	16
1.9.1	Description:	16
1.9.2	Contents:	16

1.9.3	Used by:	16
1.9.4	Derived Types:	16
1.9.5	Figure:	17
1.9.6	Schema Fragment:	17
1.10	NotificationMessageHeader	18
1.10.1	Description:	18
1.10.2	Contents:	18
1.10.3	Used by:	18
1.10.4	Derived Types:	18
1.10.5	Figure:	18
1.10.6	Schema Fragment:	18
1.11	PartyMessageInformation	19
1.11.1	Description:	19
1.11.2	Contents:	19
1.11.3	Used by:	19
1.11.4	Derived Types:	19
1.11.5	Figure:	19
1.11.6	Schema Fragment:	19
1.12	ProblemLocation	20
1.12.1	Description:	20
1.12.2	Contents:	20
1.12.3	Used by:	20
1.12.4	Derived Types:	20
1.12.5	Figure:	20
1.12.6	Schema Fragment:	20
1.13	Reason	21
1.13.1	Description:	21
1.13.2	Contents:	21
1.13.3	Used by:	21
1.13.4	Derived Types:	21
1.13.5	Figure:	21
1.13.6	Schema Fragment:	21
1.14	ReasonCode	23
1.14.1	Description:	23
1.14.2	Contents:	23
1.14.3	Used by:	23
1.14.4	Derived Types:	23
1.14.5	Figure:	23
1.14.6	Schema Fragment:	23
1.15	RequestMessage	24
1.15.1	Description:	24
1.15.2	Contents:	24
1.15.3	Used by:	24
1.15.4	Derived Types:	24
1.15.5	Figure:	25
1.15.6	Schema Fragment:	25
1.16	RequestMessageHeader	26
1.16.1	Description:	26
1.16.2	Contents:	26
1.16.3	Used by:	26
1.16.4	Derived Types:	26
1.16.5	Figure:	26
1.16.6	Schema Fragment:	26
1.17	RequestTradeStatus	27
1.17.1	Description:	27
1.17.2	Contents:	27
1.17.3	Used by:	27
1.17.4	Derived Types:	27
1.17.5	Figure:	27
1.17.6	Schema Fragment:	27
1.18	ResponseMessage	28
1.18.1	Description:	28
1.18.2	Contents:	28
1.18.3	Used by:	28

1.18.4	Derived Types:	28
1.18.5	Figure:	29
1.18.6	Schema Fragment:	29
1.19	<b>ResponseMessageHeader</b>	30
1.19.1	Description:	30
1.19.2	Contents:	30
1.19.3	Used by:	30
1.19.4	Derived Types:	30
1.19.5	Figure:	30
1.19.6	Schema Fragment:	30
1.20	<b>TradeNotFound</b>	31
1.20.1	Description:	31
1.20.2	Contents:	31
1.20.3	Used by:	31
1.20.4	Derived Types:	31
1.20.5	Figure:	31
1.20.6	Schema Fragment:	31
1.21	<b>TradeStatus</b>	32
1.21.1	Description:	32
1.21.2	Contents:	32
1.21.3	Used by:	32
1.21.4	Derived Types:	32
1.21.5	Figure:	32
1.21.6	Schema Fragment:	32
1.22	<b>TradeStatusItem</b>	33
1.22.1	Description:	33
1.22.2	Contents:	33
1.22.3	Used by:	33
1.22.4	Derived Types:	33
1.22.5	Figure:	33
1.22.6	Schema Fragment:	33
1.23	<b>TradeStatusValue</b>	34
1.23.1	Description:	34
1.23.2	Contents:	34
1.23.3	Used by:	34
1.23.4	Derived Types:	34
1.23.5	Figure:	34
1.23.6	Schema Fragment:	34
2	<b>Groups</b>	35
2.1	<b>Exception.model</b>	36
2.1.1	Description:	36
2.1.2	Contents:	36
2.1.3	Used by:	36
2.1.4	Figure:	36
2.1.5	Schema Fragment:	36
2.2	<b>MessageHeader.model</b>	37
2.2.1	Description:	37
2.2.2	Contents:	37
2.2.3	Used by:	37
2.2.4	Figure:	37
2.2.5	Schema Fragment:	37
3	<b>Schema listing</b>	39

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

## 1.1 AdditionalData

### 1.1.1 Description:

### 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:simpleContent>
    <xsd:extension base="xsd:string">
      <xsd:attribute name="additionalDataScheme" type="xsd:anyURI" use="optional"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
```



## 1.2 ContractReferenceMessage

### 1.2.1 Description:

Abstract base class for Contract notification messages that require Contract Reference only

### 1.2.2 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.

**contractReference** (exactly one occurrence; of the type ContractReference)

**party** (one or more occurrences; of the type Party)

### 1.2.3 Used by:

- Complex type: ContractCancelled

### 1.2.4 Derived Types:

- Complex type: ContractCancelled

### 1.2.5 Figure:

### 1.2.6 Schema Fragment:

```
<xsd:complexType name="ContractReferenceMessage" abstract="true">
  <xsd:annotation>
    <xsd:documentation xml:lang="en">
      Abstract base class for Contract notification messages that
      require Contract Reference only
    </xsd:documentation>
  </xsd:annotation>
  <xsd:complexContent>
    <xsd:extension base="NotificationMessage">
      <xsd:sequence>
        <xsd:element name="contractReference" type="ContractReference"/>
        <xsd:element name="party" type="Party" minOccurs="2" maxOccurs="unbounded"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

## 1.3 ConversationId

### 1.3.1 Description:

### 1.3.2 Contents:

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

•

### 1.3.3 Used by:

- Complex type: MessageHeader

### 1.3.4 Derived Types:

### 1.3.5 Figure:

### 1.3.6 Schema Fragment:

```
<xsd:complexType name="ConversationId">
  <xsd:simpleContent>
    <xsd:extension base="xsd:normalizedString">
      <xsd:attribute name="conversationIdScheme" type="xsd:anyURI" use="required"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
```

## 1.4 Message

### 1.4.1 Description:

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

### 1.4.2 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.

### 1.4.3 Used by:

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

### 1.4.4 Derived Types:

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

### 1.4.5 Figure:

### 1.4.6 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>
```

## 1.5 MessageAddress

### 1.5.1 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.

### 1.5.2 Contents:

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

•

### 1.5.3 Used by:

### 1.5.4 Derived Types:

### 1.5.5 Figure:

### 1.5.6 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>
```

## 1.6 MessageHeader

### 1.6.1 Description:

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

### 1.6.2 Contents:

**conversationId** (zero or one occurrence; of the type ConversationId) A unique identifier created by the initiator of a conversation allowing a group of message exchanges to be recognized as related.

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

### 1.6.3 Used by:

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

### 1.6.4 Derived Types:

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

### 1.6.5 Figure:

### 1.6.6 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">
          A unique identifier created by the initiator of a
          conversation allowing a group of message exchanges to be
          recognized as related.
        </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>
```

## 1.7 MessageId

### 1.7.1 Description:

The data type use for message identifiers.

### 1.7.2 Contents:

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

•

### 1.7.3 Used by:

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

### 1.7.4 Derived Types:

### 1.7.5 Figure:

### 1.7.6 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>
```

## 1.8 MessageRejected

### 1.8.1 Description:

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

### 1.8.2 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).

### 1.8.3 Used by:

### 1.8.4 Derived Types:

### 1.8.5 Figure:

### 1.8.6 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>
```

## 1.9 NotificationMessage

### 1.9.1 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.

### 1.9.2 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)

### 1.9.3 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

### 1.9.4 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

### 1.9.5 Figure:

### 1.9.6 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>
```

## 1.10 NotificationMessageHeader

### 1.10.1 Description:

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

### 1.10.2 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;

### 1.10.3 Used by:

- Complex type: NotificationMessage

### 1.10.4 Derived Types:

### 1.10.5 Figure:

### 1.10.6 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>
```

## 1.11 PartyMessageInformation

### 1.11.1 Description:

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

### 1.11.2 Contents:

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

### 1.11.3 Used by:

### 1.11.4 Derived Types:

### 1.11.5 Figure:

### 1.11.6 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>
```

## 1.12 ProblemLocation

### 1.12.1 Description:

### 1.12.2 Contents:

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

•

### 1.12.3 Used by:

- Complex type: Reason

### 1.12.4 Derived Types:

### 1.12.5 Figure:

### 1.12.6 Schema Fragment:

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

## 1.13 Reason

### 1.13.1 Description:

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

### 1.13.2 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).

### 1.13.3 Used by:

- Complex type: NovationConsentRefused

### 1.13.4 Derived Types:

### 1.13.5 Figure:

### 1.13.6 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>
```

## 1.14 ReasonCode

### 1.14.1 Description:

Defines a list of machine interpretable error codes.

### 1.14.2 Contents:

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

- 

### 1.14.3 Used by:

- Complex type: Reason

### 1.14.4 Derived Types:

### 1.14.5 Figure:

### 1.14.6 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>
```

## 1.15 RequestMessage

### 1.15.1 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.

### 1.15.2 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)

### 1.15.3 Used by:

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

### 1.15.4 Derived Types:

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

### 1.15.5 Figure:

### 1.15.6 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>
```

## 1.16 RequestMessageHeader

### 1.16.1 Description:

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

### 1.16.2 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;

### 1.16.3 Used by:

- Complex type: RequestMessage

### 1.16.4 Derived Types:

### 1.16.5 Figure:

### 1.16.6 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>
```

## 1.17 RequestTradeStatus

### 1.17.1 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.

### 1.17.2 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) An entity having a role 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.

### 1.17.3 Used by:

### 1.17.4 Derived Types:

### 1.17.5 Figure:

### 1.17.6 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">
              An entity having a role 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.
            </xsd:documentation>
          </xsd:annotation>
        </xsd:element>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

## 1.18 ResponseMessage

### 1.18.1 Description:

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

### 1.18.2 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)

### 1.18.3 Used by:

- Complex type: AcceptQuote
- Complex type: ConfirmationCancelled
- Complex type: NovationResponseMessage
- 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

### 1.18.4 Derived Types:

- Complex type: AcceptQuote
- Complex type: ConfirmationCancelled
- Complex type: NovationResponseMessage
- 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

### 1.18.5 Figure:

### 1.18.6 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>
```

## 1.19 ResponseMessageHeader

### 1.19.1 Description:

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

### 1.19.2 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;

### 1.19.3 Used by:

- Complex type: ResponseMessage

### 1.19.4 Derived Types:

### 1.19.5 Figure:

### 1.19.6 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>
```

## 1.20 TradeNotFound

### 1.20.1 Description:

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

### 1.20.2 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.

Either

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

**party** (exactly one occurrence; of the type fpml:Party) An entity having a role 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.

### 1.20.3 Used by:

### 1.20.4 Derived Types:

### 1.20.5 Figure:

### 1.20.6 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="fpml:TradeIdentifier">
            <xsd:annotation>
              <xsd:documentation xml:lang="en">
                An instance of a unique trade identifier.
              </xsd:documentation>
            </xsd:annotation>
          </xsd:element>
          <xsd:group ref="fpml:TradeOrTradeReference.model"/>
        </xsd:choice>
        <xsd:element name="party" type="fpml:Party">
          <xsd:annotation>
            <xsd:documentation xml:lang="en">
              An entity having a role 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.
            </xsd:documentation>
          </xsd:annotation>
        </xsd:element>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

## 1.21 TradeStatus

### 1.21.1 Description:

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

### 1.21.2 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) An entity having a role 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.

### 1.21.3 Used by:

### 1.21.4 Derived Types:

### 1.21.5 Figure:

### 1.21.6 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">
              An entity having a role 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.
            </xsd:documentation>
          </xsd:annotation>
        </xsd:element>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```



## 1.22 TradeStatusItem

### 1.22.1 Description:

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

### 1.22.2 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.

### 1.22.3 Used by:

- Complex type: TradeStatus

### 1.22.4 Derived Types:

### 1.22.5 Figure:

### 1.22.6 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>
```

## 1.23 TradeStatusValue

### 1.23.1 Description:

The type used to hold TradeStatusScheme values.

### 1.23.2 Contents:

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

- 

### 1.23.3 Used by:

- Complex type: TradeStatusItem

### 1.23.4 Derived Types:

### 1.23.5 Figure:

### 1.23.6 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:

### 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: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:sequence>
</xsd:group>
```

```
        </xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="partyMessageInformation" type="PartyMessageInformation" minOccurs="0" ma
      <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 targetNamespace="http://www.fpml.org/2005/FpML-4-2" elementFormDefault="qualified"
  <xsd:import namespace="http://www.w3.org/2000/09/xmldsig#" schemaLocation="xmldsig-core-schema.xsd"/>
  <xsd:include schemaLocation="fpml-doc-4-2.xsd"/>
  <xsd:complexType name="AdditionalData">
    <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="ContractReferenceMessage" abstract="true">
    <xsd:annotation>
      <xsd:documentation xml:lang="en">
        Abstract base class for Contract notification messages that
        require Contract Reference only
      </xsd:documentation>
    </xsd:annotation>
    <xsd:complexContent>
      <xsd:extension base="NotificationMessage">
        <xsd:sequence>
          <xsd:element name="contractReference" type="ContractReference"/>
          <xsd:element name="party" type="Party" minOccurs="2" maxOccurs="unbounded"/>
        </xsd:sequence>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>
  <xsd:complexType name="ConversationId">
    <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/ext
      </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">
            A unique identifier created by the initiator of a
            conversation allowing a group of message exchanges to be
            recognized as related.
          </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 it 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="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:simpleContent>
    <xsd:extension base="xsd:normalizedString">
      <xsd:attribute name="problemLocationScheme" type="xsd:anyURI"/>
    </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="unbound">
    <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:element>
                <xsd:element name="party" type="Party" maxOccurs="unbounded">
                    <xsd:annotation>
                        <xsd:documentation xml:lang="en">
                            An entity having a role 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 novator/transferor/transferee, broker,
                            calculation agent, etc.
                        </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: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="fpml:TradeIdentifier">
            <xsd:annotation>
              <xsd:documentation xml:lang="en">
                An instance of a unique trade identifier.
              </xsd:documentation>
            </xsd:annotation>
          </xsd:element>
          <xsd:group ref="fpml:TradeOrTradeReference.model"/>
        </xsd:choice>
        <xsd:element name="party" type="fpml:Party">
          <xsd:annotation>
            <xsd:documentation xml:lang="en">
              An entity having a role 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.
            </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">
              An entity having a role 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.
    </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: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">
  <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:schema>

```