XML Schema Documentation

Table of Contents

- Schema Document Properties
- Global Declarations
  - Element: **bankruptcy**
  - Element: **creditEvent**
  - Element: **creditEventNotice**
  - Element: **failureToPay**
  - Element: **obligationAcceleration**
  - Element: **obligationDefault**
  - Element: **repudiationMoratorium**
  - Element: **restructuring**
- Global Definitions
  - Complex Type: **BankruptcyEvent**
  - Complex Type: **CreditEvent**
  - Complex Type: **CreditEventNoticeDocument**
  - Complex Type: **CreditEventNotification**
  - Complex Type: **FailureToPayEvent**
  - Complex Type: **Language**
  - Complex Type: **ObligationAccelerationEvent**
  - Complex Type: **ObligationDefaultEvent**
  - Complex Type: **RepudiationMoratoriumEvent**
  - Complex Type: **Resource**
  - Complex Type: **ResourceId**
  - Complex Type: **ResourceLength**
  - Complex Type: **RestructuringEvent**
- Legend
- Glossary

top

Schema Document Properties

Target Namespace	**http://www.fpml.org/2007/FpML-4-4**
Version	$Revision: 2350 $
Element and Attribute Namespaces	• Global element and attribute declarations belong to this schema's target namespace. • By default, local element declarations belong to this schema's target namespace. • By default, local attribute declarations have no namespace.
Schema Composition	• This schema includes components from the following schema document(s): ○ fpml-posttrade-4-4.xsd

Declared Namespaces

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

fpml	http://www.fpml.org/2007/FpML-4-4
dsig	http://www.w3.org/2000/09/xmldsig#
xsd	http://www.w3.org/2001/XMLSchema

Schema Component Representation

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

top

Global Declarations

Element: bankruptcy

- *This element can be used wherever the following element is referenced:*
  - creditEvent

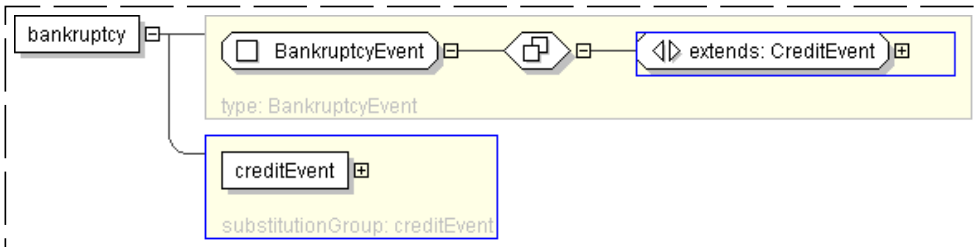
Name	bankruptcy
Type	BankruptcyEvent
Nillable	no
Abstract	no

Logical Diagram

XML Instance Representation

```
<bankruptcy/>
```

Diagram



Schema Component Representation

```
<xsd:element name="bankruptcy" type=" BankruptcyEvent " substitutionGroup="creditEvent"/>
```

top

Element: creditEvent

- *The following elements can be used wherever this element is referenced:*
  - bankruptcy
  - failureToPay
  - obligationAcceleration
  - obligationDefault
  - repudiationMoratorium
  - restructuring

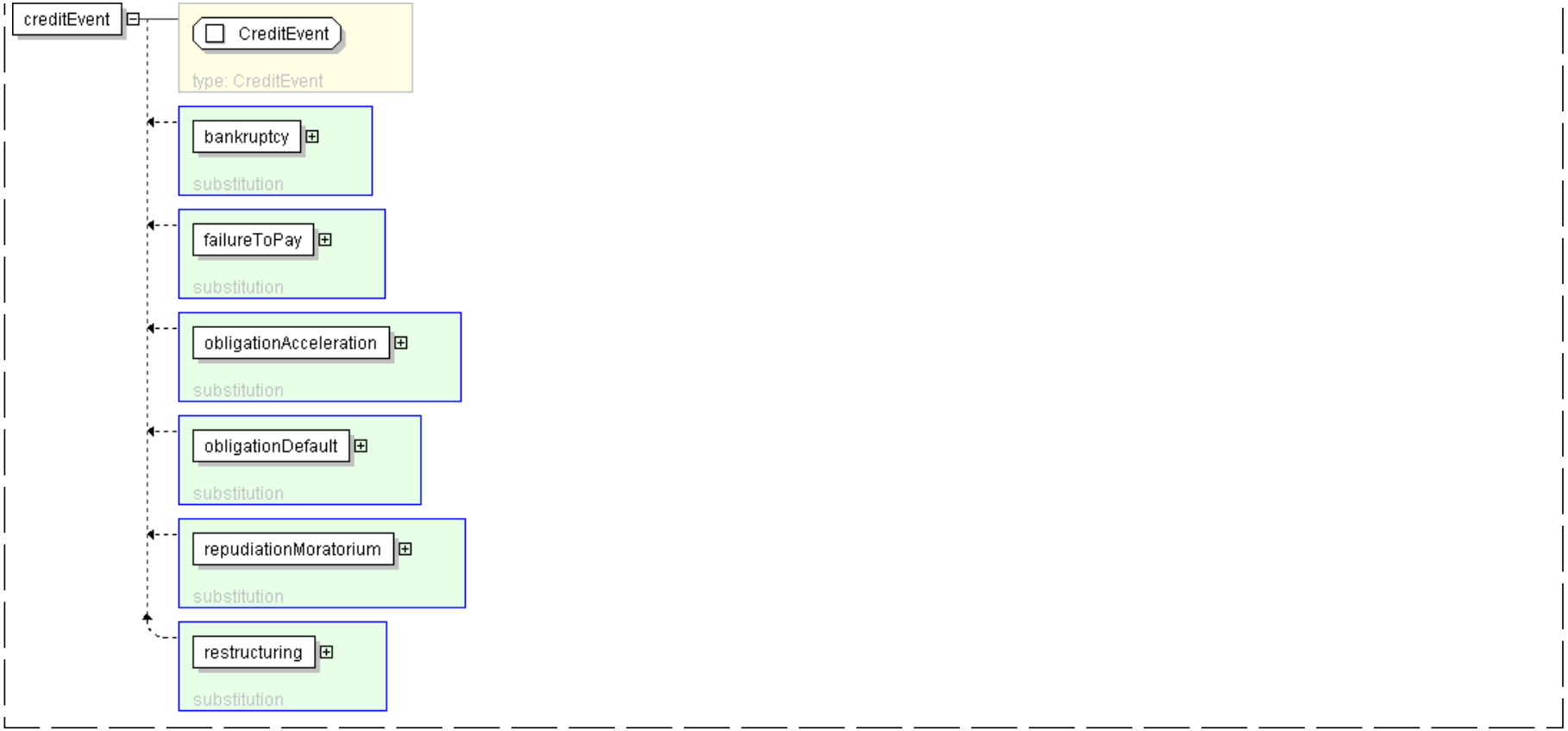
Name	creditEvent
Used by (from the same schema document)	Complex Type **CreditEventNoticeDocument**
Type	CreditEvent
Nillable	no
Abstract	yes

Logical Diagram

XML Instance Representation

```
<creditEvent/>
```

Diagram



Schema Component Representation

```
<xsd:element name="creditEvent" type=" CreditEvent " abstract="true"/>
```

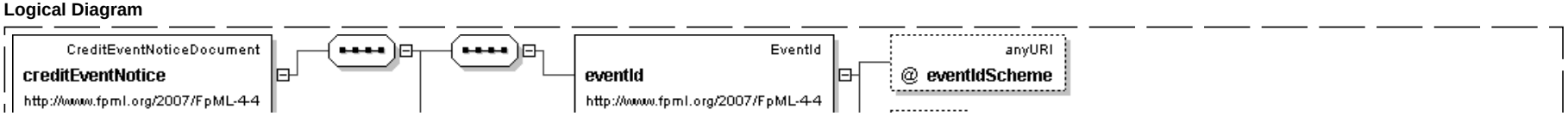
top

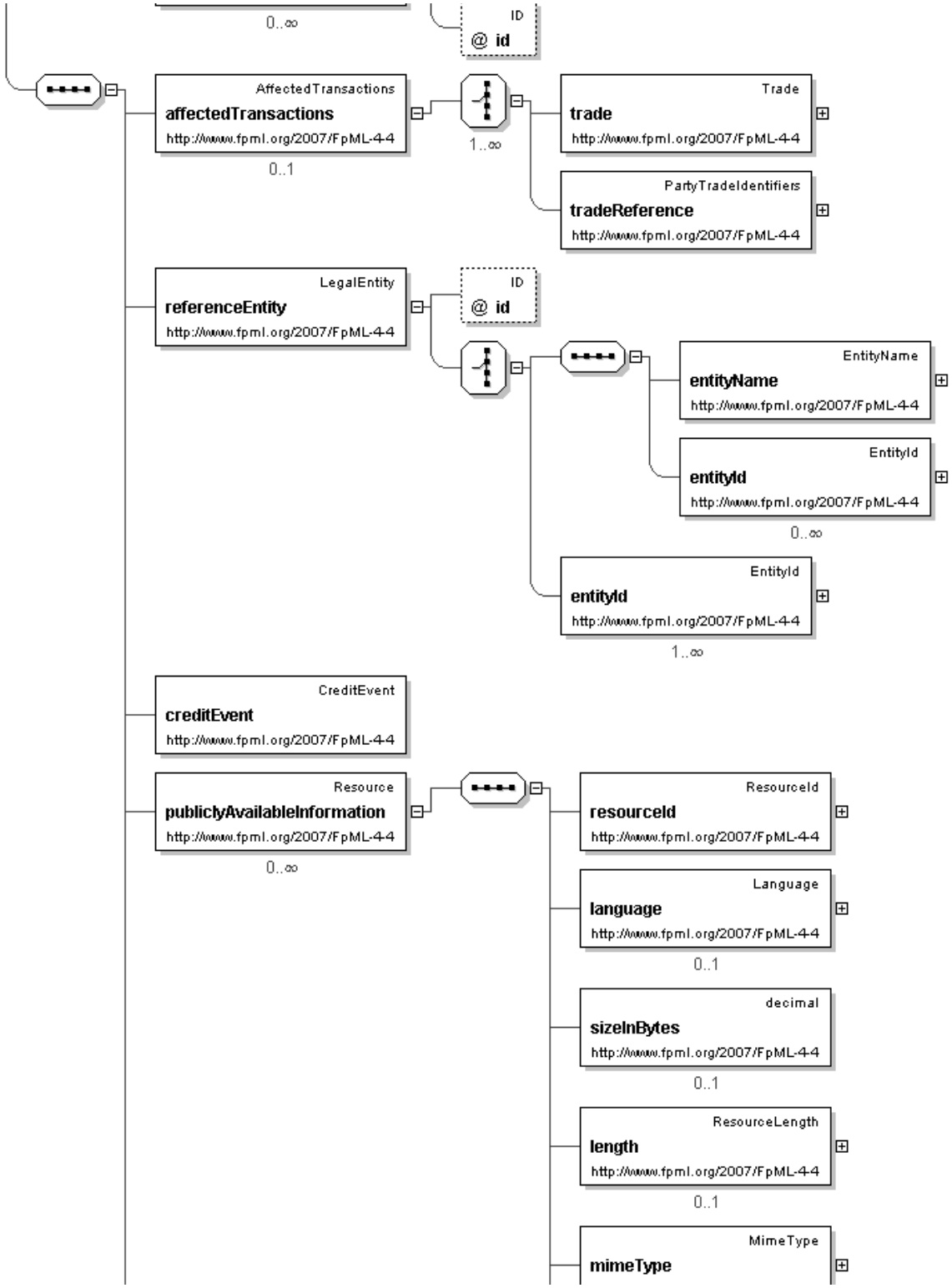
Element: creditEventNotice

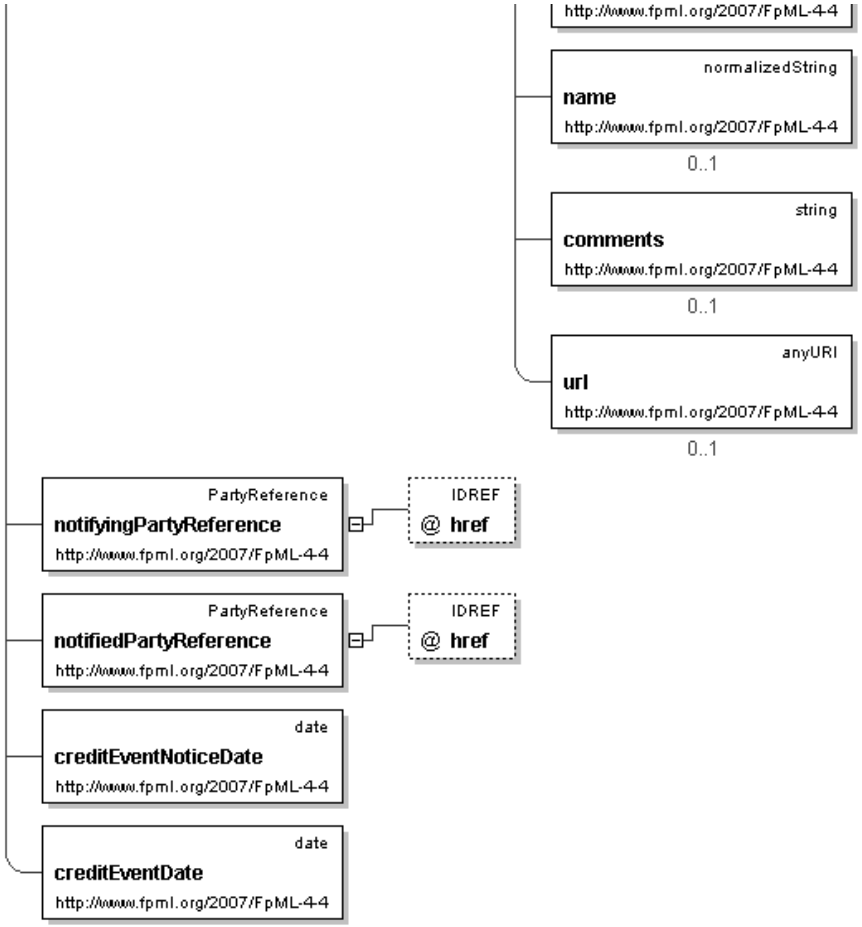
- *This element can be used wherever the following element is referenced:*
  - event

Name	creditEventNotice
Type	CreditEventNoticeDocument
Nillable	no
Abstract	no
Documentation	A global element used to hold CENs

Logical Diagram




0..∞
ID
@ id
AffectedTransactions
affectedTransactions
http://www.fpml.org/2007/FpML-4-4
0..1
1..∞
Trade
trade
http://www.fpml.org/2007/FpML-4-4
PartyTradeIdentifiers
tradeReference
http://www.fpml.org/2007/FpML-4-4
LegalEntity
referenceEntity
http://www.fpml.org/2007/FpML-4-4
ID
@ id
EntityName
entityName
http://www.fpml.org/2007/FpML-4-4
EntityId
entityId
http://www.fpml.org/2007/FpML-4-4
0..∞
EntityId
entityId
http://www.fpml.org/2007/FpML-4-4
1..∞
CreditEvent
creditEvent
http://www.fpml.org/2007/FpML-4-4
Resource
publiclyAvailableInformation
http://www.fpml.org/2007/FpML-4-4
0..∞
ResourceId
resourceId
http://www.fpml.org/2007/FpML-4-4
Language
language
http://www.fpml.org/2007/FpML-4-4
0..1
decimal
sizeInBytes
http://www.fpml.org/2007/FpML-4-4
0..1
ResourceLength
length
http://www.fpml.org/2007/FpML-4-4
0..1
MimeType
mimeType

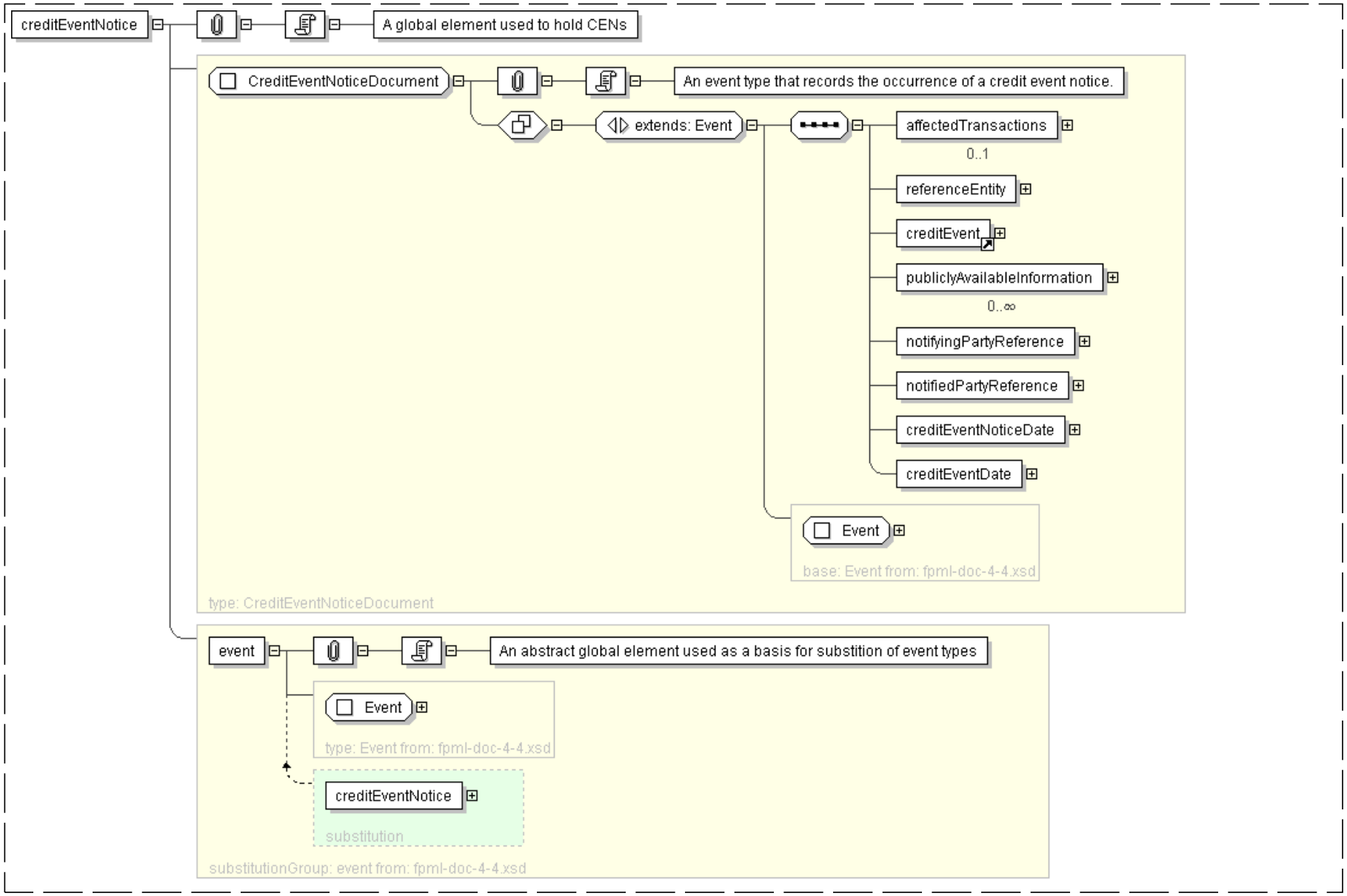


XML Instance Representation

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

   <affectedTransactions> AffectedTransactions </affectedTransactions> [0..1]
   'Trades affected by this event.'

   <referenceEntity> LegalEntity </referenceEntity> [1]
   <creditEvent> ... </creditEvent> [1]
   <publiclyAvailableInformation> Resource </publiclyAvailableInformation> [0..*]
   'A public information source, e.g. a particular newspaper or electronic news service, that
   may publish relevant information used in the determination of whether or not a credit event
   has occurred.'

   <notifyingPartyReference> PartyReference </notifyingPartyReference> [1]
   <notifiedPartyReference> PartyReference </notifiedPartyReference> [1]
   <creditEventNoticeDate> xsd:date </creditEventNoticeDate> [1]
   <creditEventDate> xsd:date </creditEventDate> [1]
```

```
</creditEventNotice>
```

Diagram



Schema Component Representation

```
<xsd:element name="creditEventNotice" type=" CreditEventNoticeDocument
" substitutionGroup="event"/>
```

top

Element: failureToPay

- *This element can be used wherever the following element is referenced:*
  - creditEvent

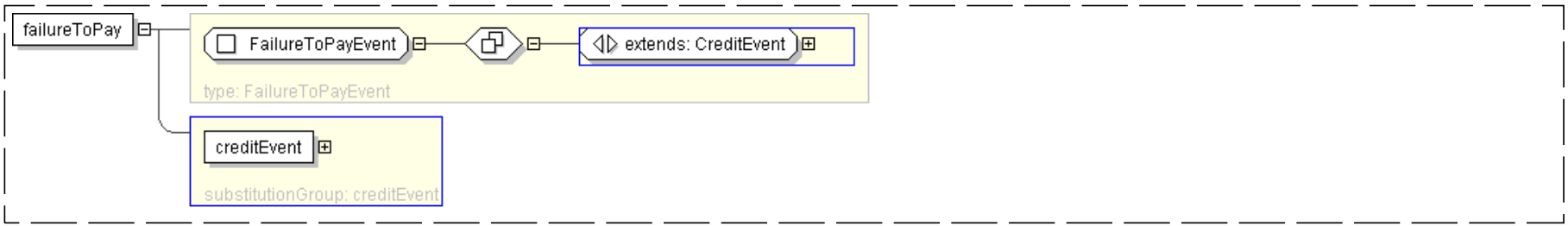
Name	failureToPay
Type	FailureToPayEvent
Nillable	no
Abstract	no

Logical Diagram

XML Instance Representation

```
<failureToPay/>
```

Diagram



Schema Component Representation

```
<xsd:element name="failureToPay" type=" FailureToPayEvent " substitutionGroup="creditEvent"/>
```

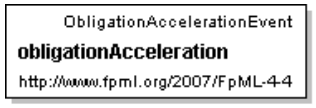
top

Element: obligationAcceleration

- *This element can be used wherever the following element is referenced:*
  - creditEvent

Name	obligationAcceleration
Type	ObligationAccelerationEvent
Nillable	no
Abstract	no

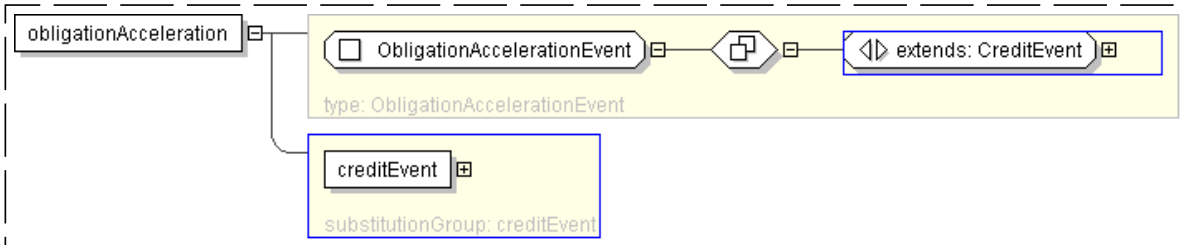
Logical Diagram



XML Instance Representation

```
<obligationAcceleration/>
```

Diagram



Schema Component Representation

```
<xsd:element name="obligationAcceleration" type=" ObligationAccelerationEvent
" substitutionGroup="creditEvent"/>
```

top

Element: obligationDefault

- *This element can be used wherever the following element is referenced:*
  - creditEvent

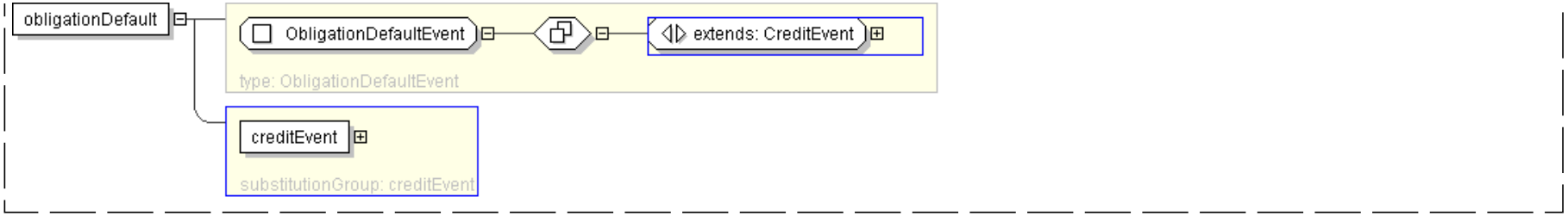
Name	obligationDefault
Type	ObligationDefaultEvent
Nillable	no
Abstract	no

Logical Diagram

XML Instance Representation

```
<obligationDefault/>
```

Diagram



Schema Component Representation

```
<xsd:element name="obligationDefault" type=" ObligationDefaultEvent
" substitutionGroup="creditEvent"/>
```

top

Element: repudiationMoratorium

- *This element can be used wherever the following element is referenced:*
  - creditEvent

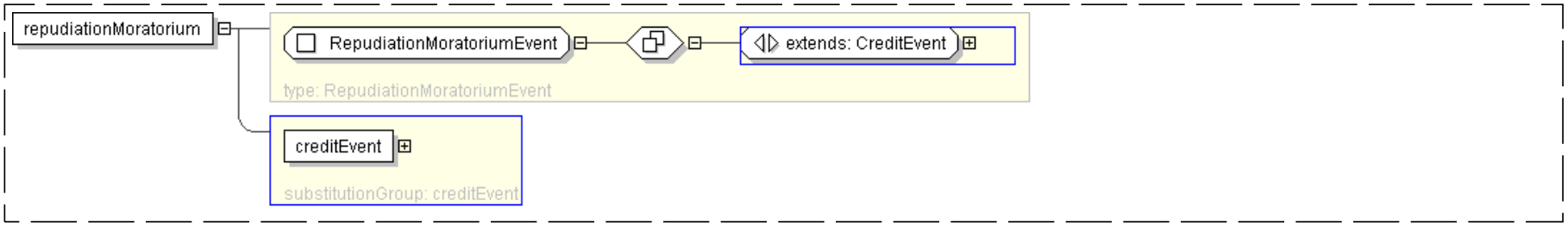
Name	repudiationMoratorium
Type	RepudiationMoratoriumEvent
Nillable	no
Abstract	no

Logical Diagram

XML Instance Representation

```
<repudiationMoratorium/>
```

Diagram



Schema Component Representation

```
<xsd:element name="repudiationMoratorium" type=" RepudiationMoratoriumEvent
" substitutionGroup="creditEvent"/>
```

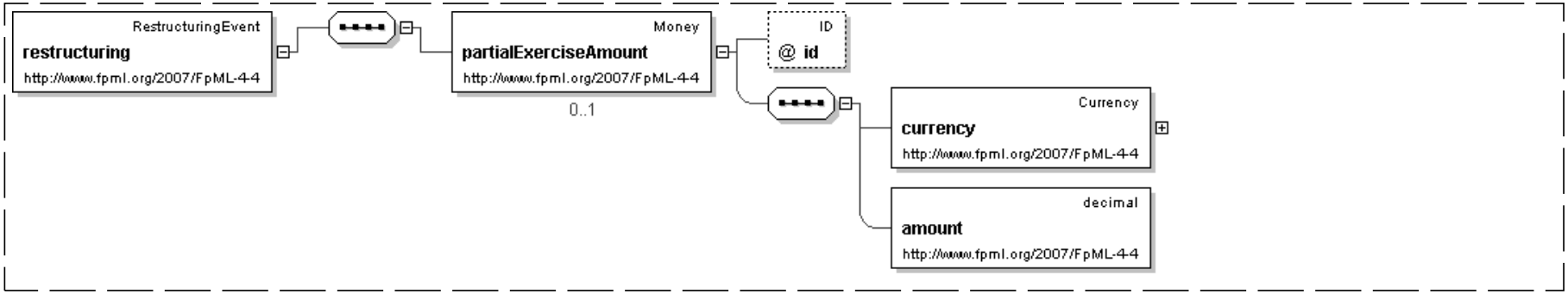
top

Element: restructuring

- *This element can be used wherever the following element is referenced:*
  - creditEvent

Name	restructuring
Type	RestructuringEvent
Nillable	no
Abstract	no

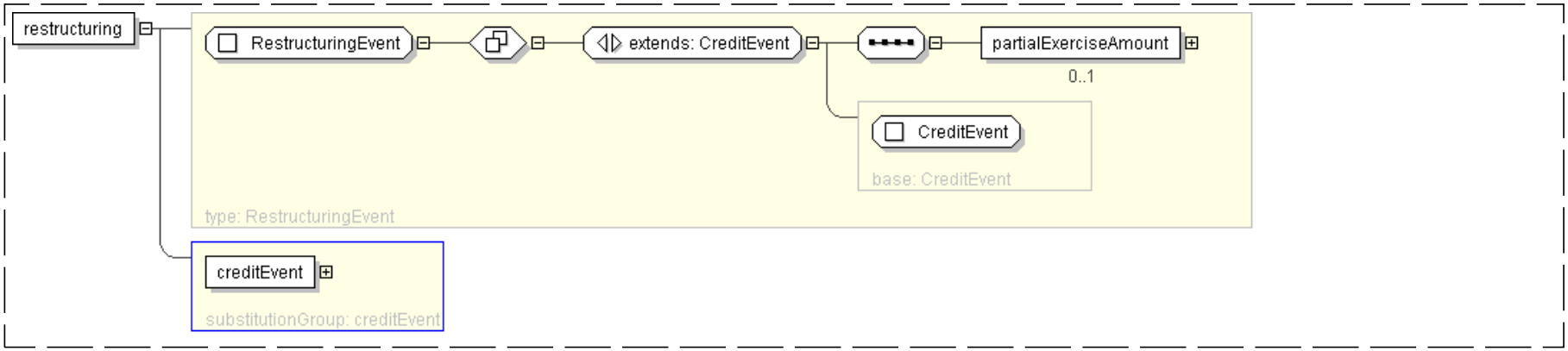
Logical Diagram



XML Instance Representation

```
<restructuring>
   <partialExerciseAmount> Money </partialExerciseAmount> [0..1]
</restructuring>
```

Diagram



Schema Component Representation

```
<xsd:element name="restructuring" type=" RestructuringEvent " substitutionGroup="creditEvent"/>
```

top

Global Definitions

Complex Type: BankruptcyEvent

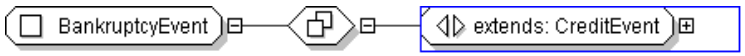
Super-types:	CreditEvent < **BankruptcyEvent** (by extension)
Sub-types:	None

Name	BankruptcyEvent
Used by (from the same schema document)	Element **bankruptcy**
Abstract	no

XML Instance Representation

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

Diagram



Schema Component Representation

```
<xsd:complexType name="BankruptcyEvent">
   <xsd:complexContent>
      <xsd:extension base=" CreditEvent "/>
   </xsd:complexContent>
</xsd:complexType>
```

top

Complex Type: CreditEvent

Super-types:	None
Sub-types:	• BankruptcyEvent (by extension) • FailureToPayEvent (by extension) • ObligationAccelerationEvent (by extension) • ObligationDefaultEvent (by extension) • RepudiationMoratoriumEvent (by extension) • RestructuringEvent (by extension)

Name	CreditEvent
Used by (from the same schema document)	Element **creditEvent**
Abstract	no

XML Instance Representation

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

Diagram

Schema Component Representation

```
<xsd:complexType name="CreditEvent"/>
```

top

Complex Type: CreditEventNoticeDocument

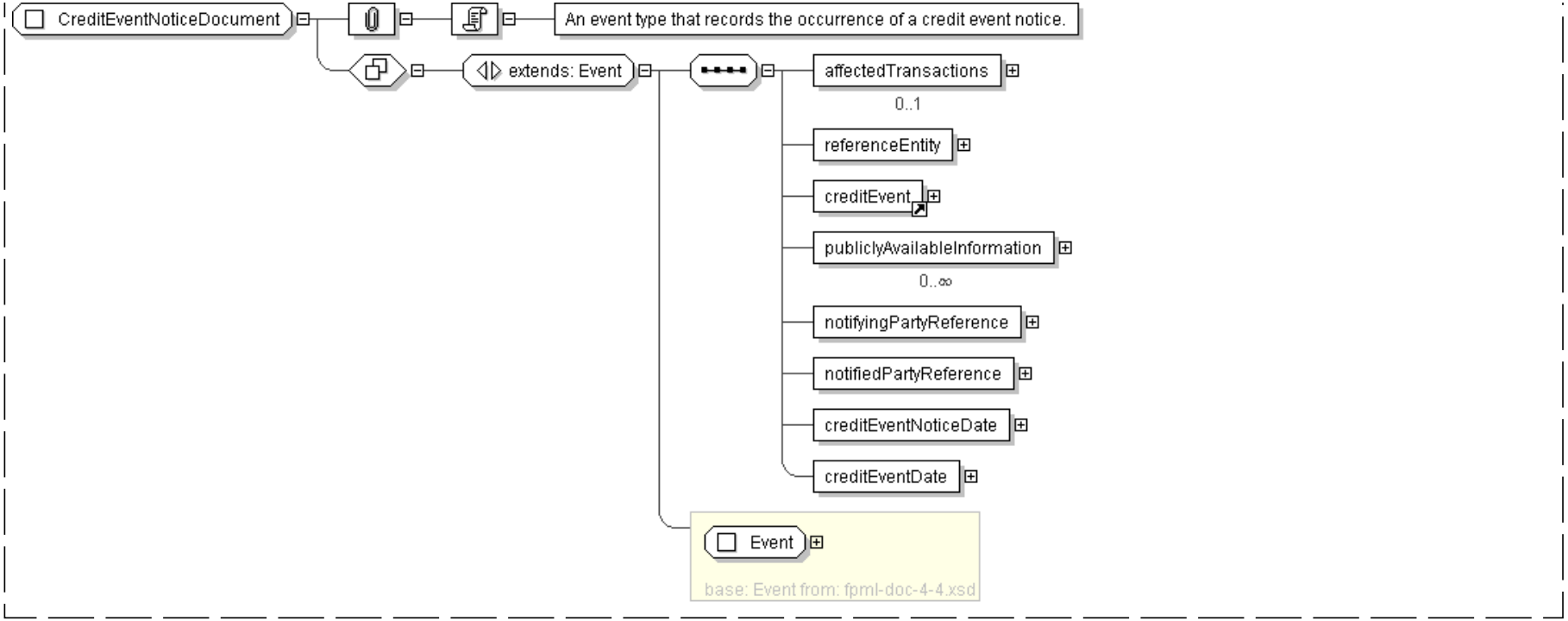
Super-types:	Event < **CreditEventNoticeDocument** (by extension)
Sub-types:	None

Name	CreditEventNoticeDocument
Used by (from the same schema document)	Complex Type **CreditEventNotification** , Element **creditEventNotice**
Abstract	no
Documentation	An event type that records the occurrence of a credit event notice.

XML Instance Representation

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

   <affectedTransactions> AffectedTransactions </affectedTransactions> [0..1]
   'Trades affected by this event.'

   <referenceEntity> LegalEntity </referenceEntity> [1]
   <creditEvent> ... </creditEvent> [1]
   <publiclyAvailableInformation> Resource </publiclyAvailableInformation> [0..*]
   'A public information source, e.g. a particular newspaper or electronic news service, that
   may publish relevant information used in the determination of whether or not a credit event
   has occurred.'

   <notifyingPartyReference> PartyReference </notifyingPartyReference> [1]
   <notifiedPartyReference> PartyReference </notifiedPartyReference> [1]
   <creditEventNoticeDate> xsd:date </creditEventNoticeDate> [1]
   <creditEventDate> xsd:date </creditEventDate> [1]
</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="CreditEventNoticeDocument">
   <xsd:complexContent>
      <xsd:extension base=" Event ">
         <xsd:sequence>
            <xsd:element name="affectedTransactions" type=" AffectedTransactions " minOccurs="0"/>
            <xsd:element name="referenceEntity" type=" LegalEntity "/>
            <xsd:element ref=" creditEvent "/>
            <xsd:element name="publiclyAvailableInformation" type=" Resource "
            minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="notifyingPartyReference" type=" PartyReference "/>
            <xsd:element name="notifiedPartyReference" type=" PartyReference "/>
            <xsd:element name="creditEventNoticeDate" type=" xsd:date "/>
            <xsd:element name="creditEventDate" type=" xsd:date "/>
         </xsd:sequence>
      </xsd:extension>
   </xsd:complexContent>
</xsd:complexType>
```

top

Complex Type: CreditEventNotification

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

Name	CreditEventNotification
Abstract	no

XML Instance Representation

```
<...
version=" xsd:token (value comes from list: {'4-0'|'4-1'|'4-2'|'4-3'|'4-4'}) [1]
 'Indicate which version of the FpML Schema an FpML message adheres to.'

"
expectedBuild=" xsd:positiveInteger [0..1]
 'This optional attribute can be supplied by a message creator in an FpML instance to
 specify which build number of the schema was used to define the message when it was generated.'

"
actualBuild="2 [0..1]
 'The specific build number of this schema version. This attribute is not included in
 an instance document. Instead, it is supplied by the XML parser when the document is
 validated against the FpML schema and indicates the build number of the schema file. Every
 time FpML publishes a change to the schema, validation rules, or examples within a version
 (e.g., version 4.2) the actual build number is incremented. If no changes have been
 made between releases within a version (i.e. from Trial Recommendation to Recommendation)
 the actual build number stays the same.'

">
   <header> NotificationMessageHeader </header> [1]
   <validation> Validation </validation> [0..*]
   <creditEventNotice> CreditEventNoticeDocument </creditEventNotice> [1]
   <party> Party </party> [2..*]
</...>
```

Diagram

Schema Component Representation

```
<xsd:complexType name="CreditEventNotification">
   <xsd:complexContent>
      <xsd:extension base=" NotificationMessage ">
         <xsd:sequence>
            <xsd:element name="creditEventNotice" type=" CreditEventNoticeDocument "/>
            <xsd:element name="party" type=" Party " minOccurs="2" maxOccurs="unbounded"/>
         </xsd:sequence>
      </xsd:extension>
   </xsd:complexContent>
</xsd:complexType>
```

top

Complex Type: FailureToPayEvent

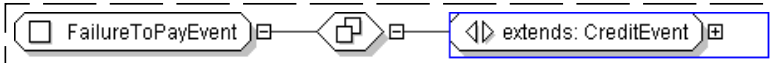
Super-types:	CreditEvent < **FailureToPayEvent** (by extension)
Sub-types:	None

Name	FailureToPayEvent
Used by (from the same schema document)	Element **failureToPay**
Abstract	no

XML Instance Representation

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

Diagram



Schema Component Representation

```
<xsd:complexType name="FailureToPayEvent">
   <xsd:complexContent>
      <xsd:extension base=" CreditEvent "/>
   </xsd:complexContent>
</xsd:complexType>
```

top

Complex Type: Language

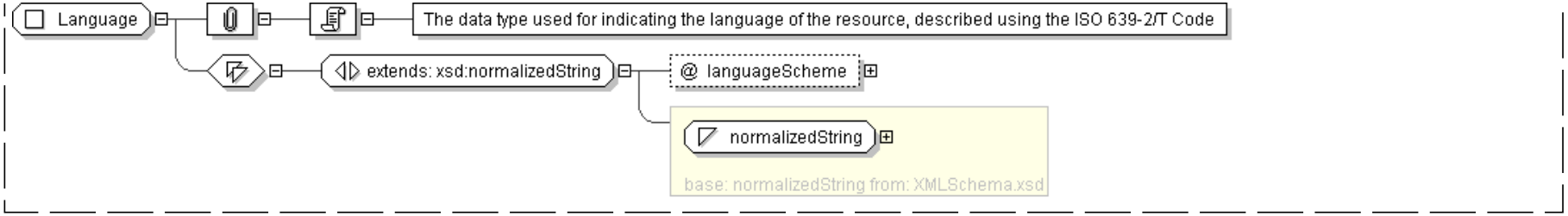
Super-types:	xsd:normalizedString < **Language** (by extension)
Sub-types:	None

Name	Language
Used by (from the same schema document)	Complex Type **Resource**
Abstract	no
Documentation	The data type used for indicating the language of the resource, described using the ISO 639-2/T Code

XML Instance Representation

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

Diagram



Schema Component Representation

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

top

Complex Type: ObligationAccelerationEvent

Super-types:	CreditEvent < **ObligationAccelerationEvent** (by extension)
Sub-types:	None

Name	ObligationAccelerationEvent
Used by (from the same schema document)	Element **obligationAcceleration**
Abstract	no

XML Instance Representation

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

Diagram

Schema Component Representation

```
<xsd:complexType name="ObligationAccelerationEvent">
   <xsd:complexContent>
      <xsd:extension base=" CreditEvent "/>
   </xsd:complexContent>
</xsd:complexType>
```

top

Complex Type: ObligationDefaultEvent

Super-types:	CreditEvent < **ObligationDefaultEvent** (by extension)

Sub-types:	None

Name	ObligationDefaultEvent
Used by (from the same schema document)	Element **obligationDefault**
Abstract	no

XML Instance Representation

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

Diagram

Schema Component Representation

```
<xsd:complexType name="ObligationDefaultEvent">
   <xsd:complexContent>
      <xsd:extension base=" CreditEvent "/>
   </xsd:complexContent>
</xsd:complexType>
```

top

Complex Type: RepudiationMoratoriumEvent

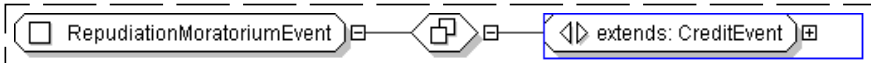
Super-types:	CreditEvent < **RepudiationMoratoriumEvent** (by extension)
Sub-types:	None

Name	RepudiationMoratoriumEvent
Used by (from the same schema document)	Element **repudiationMoratorium**
Abstract	no

XML Instance Representation

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

Diagram



Schema Component Representation

```
<xsd:complexType name="RepudiationMoratoriumEvent">
   <xsd:complexContent>
      <xsd:extension base=" CreditEvent "/>
   </xsd:complexContent>
</xsd:complexType>
```

top

Complex Type: Resource

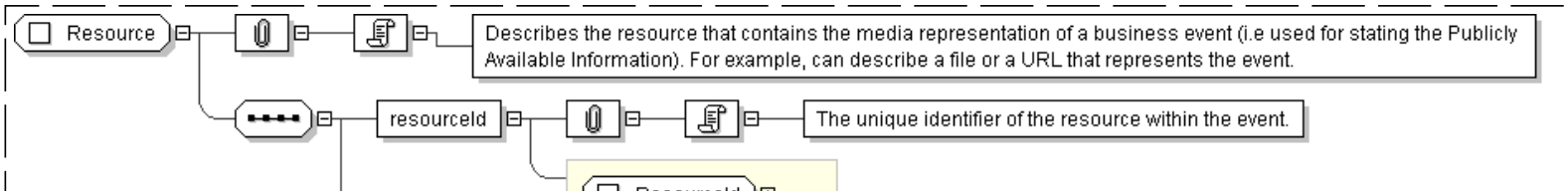
Super-types:	None
Sub-types:	None

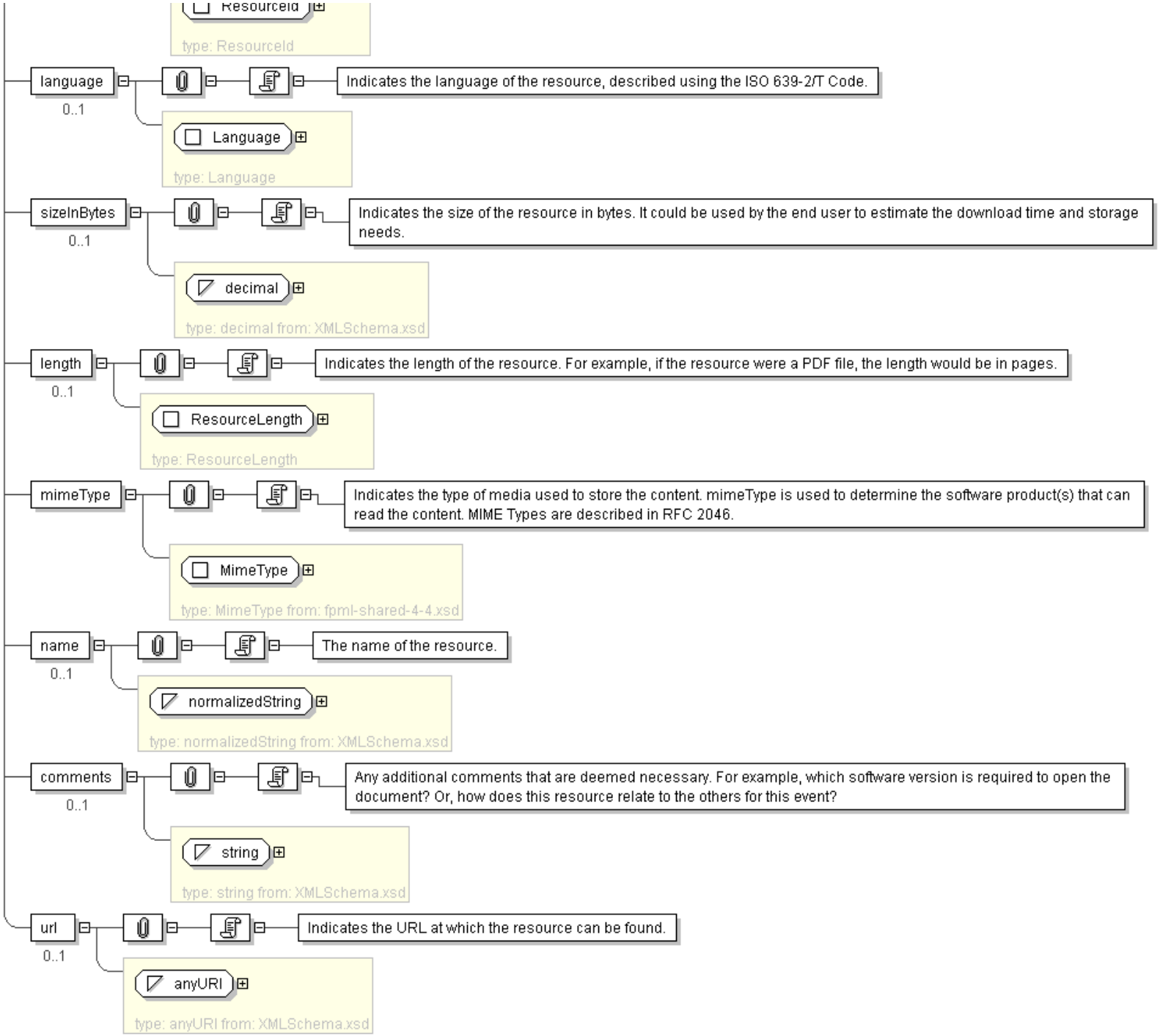
Name	Resource
Used by (from the same schema document)	Complex Type **CreditEventNoticeDocument**
Abstract	no
Documentation	Describes the resource that contains the media representation of a business event (i.e used for stating the Publicly Available Information). For example, can describe a file or a URL that represents the event.

XML Instance Representation

```
<...>
    <resourceId> ResourceId </resourceId> [1]
    'The unique identifier of the resource within the event.'

    <language> Language </language> [0..1]
    'Indicates the language of the resource, described using the ISO 639-2/T Code.'

    <sizeInBytes> xsd:decimal </sizeInBytes> [0..1]
    'Indicates the size of the resource in bytes. It could be used by the end user to estimate
    the download time and storage needs.'

    <length> ResourceLength </length> [0..1]
    'Indicates the length of the resource. For example, if the resource were a PDF file, the
    length would be in pages.'

    <mimeType> MimeType </mimeType> [1]
    'Indicates the type of media used to store the content. mimeType is used to determine
    the software product(s) that can read the content. MIME Types are described in RFC 2046.'

    <name> xsd:normalizedString </name> [0..1]
    'The name of the resource.'

    <comments> xsd:string </comments> [0..1]
    'Any additional comments that are deemed necessary. For example, which software version
    is required to open the document? Or, how does this resource relate to the others for
    this event?'

    <url> xsd:anyURI </url> [0..1]
    'Indicates the URL at which the resource can be found.'

</...>
```

Diagram





Schema Component Representation

```
<xsd:complexType name="Resource">
```

```
   <xsd:sequence>
      <xsd:element name="resourceId" type=" ResourceId "/>
      <xsd:element name="language" type=" Language " minOccurs="0"/>
      <xsd:element name="sizeInBytes" type=" xsd:decimal " minOccurs="0"/>
      <xsd:element name="length" type=" ResourceLength " minOccurs="0"/>
      <xsd:element name="mimeType" type=" MimeType "/>
      <xsd:element name="name" type=" xsd:normalizedString " minOccurs="0"/>
      <xsd:element name="comments" type=" xsd:string " minOccurs="0"/>
      <xsd:element name="url" type=" xsd:anyURI " minOccurs="0"/>
   </xsd:sequence>
</xsd:complexType>
```

top

Complex Type: ResourceId

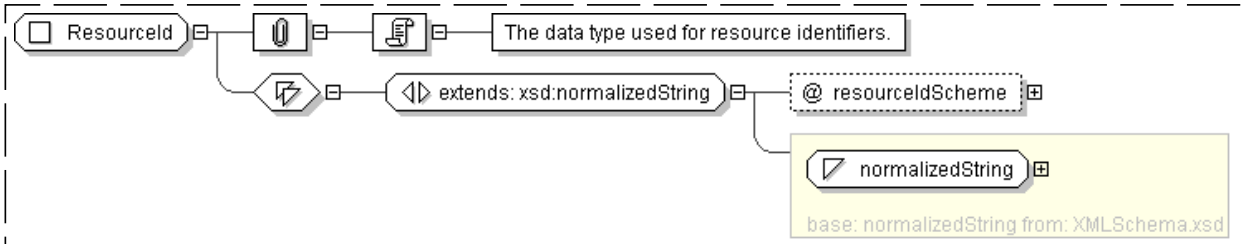
Super-types:	xsd:normalizedString < **ResourceId** (by extension)
Sub-types:	None

Name	ResourceId
Used by (from the same schema document)	Complex Type **Resource**
Abstract	no
Documentation	The data type used for resource identifiers.

XML Instance Representation

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

Diagram



Schema Component Representation

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

top

Complex Type: ResourceLength

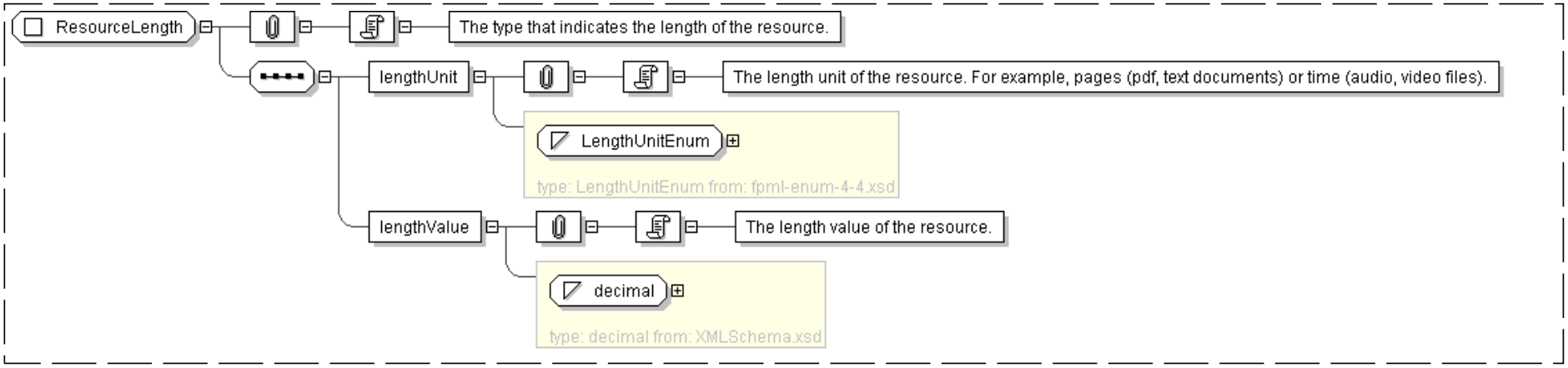
Super-types:	None
Sub-types:	None

Name	ResourceLength
Used by (from the same schema document)	Complex Type **Resource**
Abstract	no
Documentation	The type that indicates the length of the resource.

XML Instance Representation

```
<...>
   <lengthUnit> LengthUnitEnum </lengthUnit> [1]
   'The length unit of the resource. For example, pages (pdf, text documents) or time
   (audio, video files).'

   <lengthValue> xsd:decimal </lengthValue> [1]
   'The length value of the resource.'

</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="ResourceLength">
   <xsd:sequence>
      <xsd:element name="lengthUnit" type=" LengthUnitEnum "/>
      <xsd:element name="lengthValue" type=" xsd:decimal "/>
   </xsd:sequence>
</xsd:complexType>
```

top

Complex Type: RestructuringEvent

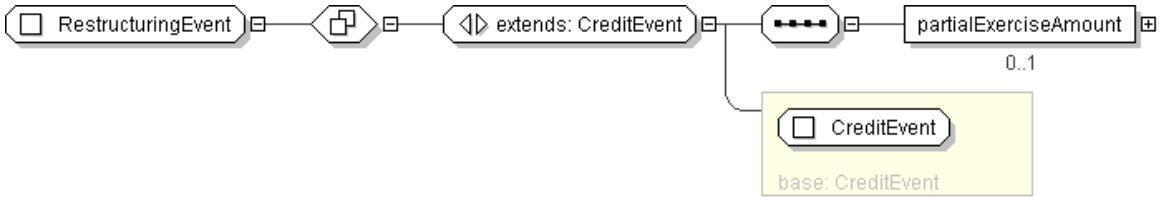
Super-types:	CreditEvent < **RestructuringEvent** (by extension)
Sub-types:	None

Name	RestructuringEvent
Used by (from the same schema document)	Element **restructuring**
Abstract	no

XML Instance Representation

```
<...>
   <partialExerciseAmount> Money </partialExerciseAmount> [0..1]
</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="RestructuringEvent">
   <xsd:complexContent>
      <xsd:extension base=" CreditEvent ">
         <xsd:sequence>
            <xsd:element name="partialExerciseAmount" type=" Money " minOccurs="0"/>
         </xsd:sequence>
      </xsd:extension>
   </xsd:complexContent>
</xsd:complexType>
```

top

Legend

Complex Type: Schema Component Type

AusAddress Schema Component Name

Super-types:	Address < AusAddress (by extension)
Sub-types:	• QLDAddress (by restriction)

If this schema component is a type definition, its type hierarchy is shown in a gray-bordered box.

Name	AusAddress
Abstract	no

The table above displays the properties of this schema component.

XML Instance Representation

```
<... country="Australia" >
<unitNo> string </unitNo> [0..1]
<houseNo> string </houseNo> [1]
<street> string </street> [1]
Start Choice [1]
<city> string </city> [1]
<town> string </town> [1]
End Choice
<state> AusStates </state> [1]
<postcode> string <<pattern = [1-9][0-9]{3}>> </postcode> [1]
</...>
```

The XML Instance Representation table above shows the schema component's content as an XML instance.

- The minimum and maximum occurrence of elements and attributes are provided in square brackets, e.g. [0..1].
- Model group information are shown in gray, e.g. Start Choice ... End Choice.
- For type derivations, the elements and attributes that have been added to or changed from the base type's content are shown in **bold**.
- If an element/attribute has a fixed value, the fixed value is shown in green, e.g. country="Australia".
- Otherwise, the type of the element/attribute is displayed.
  - If the element/attribute's type is in the schema, a link is provided to it.
  - For local simple type definitions, the constraints are displayed in angle brackets, e.g. <<*pattern* = [1-9][0-9]{3}>>.

Schema Component Representation

```
<complexType name="AusAddress">
<complexContent>
<extension base=" Address ">
<sequence>
<element name="state" type=" AusStates "/>
<element name="postcode">
<simpleType>
<restriction base=" string ">
<pattern value="[1-9][0-9]{3}"/>
</restriction>
</simpleType>
</element>
</sequence>
<attribute name="country" type=" string " fixed="Australia"/>
</extension>
</complexContent>
</complexType>
```

The Schema Component Representation table above displays the underlying XML representation of the schema component. (Annotations are not shown.)

top

Glossary

Abstract (Applies to complex type definitions and element declarations). An abstract element or complex type cannot used to validate an element instance. If there is a reference to an abstract element, only element declarations that can substitute the abstract element can be used to validate the instance. For references to abstract type definitions, only derived types can be used.

All Model Group Child elements can be provided *in any order* in instances. See: http://www.w3.org/TR/xmlschema-1/#element-all.

Choice Model Group *Only one* from the list of child elements and model groups can be provided in instances. See: http://www.w3.org/TR/xmlschema-1/#element-choice.

Collapse Whitespace Policy Replace tab, line feed, and carriage return characters with space character (Unicode character 32). Then, collapse contiguous sequences of space characters into single space character, and remove leading and trailing space characters.

Disallowed Substitutions (Applies to element declarations). If *substitution* is specified, then substitution group members cannot be used in place of the given element declaration to validate element instances. If *derivation methods*, e.g. extension, restriction, are specified, then the given element declaration will not validate element instances that have types derived from the element declaration's type using the specified derivation methods. Normally, element instances can override their declaration's type by specifying an `xsi:type` attribute.

Key Constraint Like Uniqueness Constraint, but additionally requires that the specified value(s) must be provided. See: http://www.w3.org/TR/xmlschema-1/#cIdentity-constraint_Definitions.

Key Reference Constraint Ensures that the specified value(s) must match value(s) from a Key Constraint or Uniqueness Constraint. See: http://www.w3.org/TR/xmlschema-1/#cIdentity-constraint_Definitions.

Model Group Groups together element content, specifying the order in which the element content can occur and the number of times the group of element content may be repeated. See: http://www.w3.org/TR/xmlschema-1/#Model_Groups.

Nillable (Applies to element declarations). If an element declaration is nillable, instances can use the `xsi:nil` attribute. The `xsi:nil` attribute is the boolean attribute, *nil*, from the *http://www.w3.org/2001/XMLSchema-instance* namespace. If an element instance has an `xsi:nil` attribute set to true, it can be left empty, even though its element declaration may have required content.

Notation A notation is used to identify the format of a piece of data. Values of elements and attributes that are of type, NOTATION, must come from the names of declared notations. See: http://www.w3.org/TR/xmlschema-1/#cNotation_Declarations.

Preserve Whitespace Policy Preserve whitespaces exactly as they appear in instances.

Prohibited Derivations (Applies to type definitions). Derivation methods that cannot be used to create sub-types from a given type definition.

Prohibited Substitutions (Applies to complex type definitions). Prevents sub-types that have been derived using the specified derivation methods from validating element instances in place of the given type definition.

Replace Whitespace Policy Replace tab, line feed, and carriage return characters with space character (Unicode character 32).

Sequence Model Group Child elements and model groups must be provided *in the specified order* in instances. See: http://www.w3.org/TR/xmlschema-1/#element-sequence.

Substitution Group Elements that are *members* of a substitution group can be used wherever the *head* element of the substitution group is referenced.

Substitution Group Exclusions (Applies to element declarations). Prohibits element declarations from nominating themselves as being able to substitute a given element declaration, if they have types that are derived from the original element's type using the specified derivation methods.

Target Namespace The target namespace identifies the namespace that components in this schema belongs to. If no target namespace is provided, then the schema components do not belong to any namespace.

Uniqueness Constraint Ensures uniqueness of an element/attribute value, or a combination of values, within a specified scope. See: http://www.w3.org/TR/xmlschema-1/#cIdentity-constraint_Definitions.

top

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