Specification

Nobuhiro Sasagawa, Western Illinois University
Laurence Leff, Western Illinois University
Daniel Greenwood
Massachusetts Institute of Technology, Electronic Commerce Architecture Program
David Marvit
Fujitsu

XML [legalxml-econ-standard-1.0.xml]
HTML [legalxml-econ-standard-1.0.html]
PDF (A4) [legalxml-econ-standard-1.0.pdf]
{OASIS Document Number}
http://docs.oasis-open.org/legalxml-econ/
http://docs.oasis-open.org/templates/DocBook/spec-1.0/template/legalxml-econ-standard-1.0.html
http://www.oasis-open.org/spectools/docs/wd-spectools-docbook-template-03.html

OASIS LegalXML eContracts TC

Copyright © 2006 OASIS Open, Inc. All Rights Reserved.

Related Work

Status

This document was last revised or approved by the LegalXML eContracts Technical Committee on the above date. The level of approval is also listed above. Check the current location noted above for possible later revisions of this document. This document is updated periodically on no particular schedule.

Technical Committee members should send comments on this specification to the Technical Committee’s email list. Others should send comments to the Technical Committee by using the ”Send A Comment” button on the Technical Committee’s web page at http://www.oasis-open.org/committees/legalxml-econtracts.

For information on whether any patents have been disclosed that may be essential to implementing this specification, and any offers of patent licensing terms, please refer to the Intellectual Property Rights section of the Technical Committee web page (http://www.oasis-open.org/committees/legalxml-econtractsipr.php).

The non-normative errata page for this specification is located at http://www.oasis-open.org/committees/legalxml-econtracts
[http://www.oasis-open.org/committees/legalxml-econtracts].
Notices

For committees formed after April 15, 2005 and those other committees that have transitioned to the new IPR policy, the notices required for OASIS documents can be found at http://www.oasis-open.org/who/intellectualproperty.php#notices and, as of the time of writing this document read as follows (square-bracketed language need only appear in specification documents) though it is block quoted solely for explanatory reasons and would not be so quoted when used:

Copyright © OASIS Open 2006. All Rights Reserved.

All capitalized terms in the following text have the meanings assigned to them in the OASIS Intellectual Property Rights Policy (the "OASIS IPR Policy"). The full Policy may be found at the OASIS website.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published, and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this section are included on all such copies and derivative works. However, this document itself may not be modified in any way, including by removing the copyright notice or references to OASIS, except as needed for the purpose of developing any document or deliverable produced by an OASIS Technical Committee (in which case the rules applicable to copyrights, as set forth in the OASIS IPR Policy, must be followed) or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by OASIS or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY OWNERSHIP RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

[OASIS requests that any OASIS Party or any other party that believes it has patent claims that would necessarily be infringed by implementations of this OASIS Committee Specification or OASIS Standard, to notify OASIS TC Administrator and provide an indication of its willingness to grant patent licenses to such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced this specification.]

[OASIS invites any party to contact the OASIS TC Administrator if it is aware of a claim of ownership of any patent claims that would necessarily be infringed by implementations of this specification by a patent holder that is not willing to provide a license to such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced this specification. OASIS may include such claims on its website, but disclaims any obligation to do so.]

[OASIS takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on OASIS procedures with respect to rights in any document or deliverable produced by an OASIS Technical Committee can be found on the OASIS website. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this OASIS Committee Specification or OASIS Standard, can be obtained from the OASIS TC Ad-
For those committees that have not transitioned to the new IPR policy, the notices read as follows:

OASIS takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on OASIS’s procedures with respect to rights in OASIS specifications can be found at the OASIS website. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementors or users of this specification, can be obtained from the OASIS Executive Director.

OASIS invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to implement this specification. Please address the information to the OASIS Executive Director.

Copyright © OASIS Open 2006. All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to OASIS, except as needed for the purpose of developing OASIS specifications, in which case the procedures for copyrights defined in the OASIS Intellectual Property Rights document must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by OASIS or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREBIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

08 August 2006

This specification defines XML Markup to enable the efficient creation, maintenance, management, exchange and publication of contract documents and contract terms. This, as the first specification from this TC, will emphasize the use case of "Contracts with negotiated narrative terms." Use cases covered include ticket contracts, standard form business and consumer contracts and the documents associated with click-through agreements such as End-User License Agreements (EULAs)

In particular, the specification is designed to support back-end processing, where documents or document fragments are used to create new documents. [meyer05]
Introduction and Scope

The mission of the OASIS Legal XML Member Section eContracts Technical Committee, hereinafter called the TC, is to provide a standard for XML markup to enable "the efficient creation, maintenance, management, exchange and publication of contract documents and contract terms." [RD]

This, the first edition of the first specification, will focus on 'contracts with negotiated narrative terms.' They are usually prepared by lawyers on behalf of their clients and are the most common type of contract document prepared by law firms, corporate counsels and the legal departments in governments. Sometimes, a legal department has identified a precedent contract, which is used by a non-lawyer in the negotiation of the final narrative terms.

In this scenario, the parties would often reach an agreement, or "hand shake deal," on the business terms and basic parameters of the contract. Then, one of the lawyers would prepare a draft contract document. The other party's lawyer would review it and the two lawyers would negotiate the terms until a final document could be prepared. Both parties would then assent to the document, conventionally, with pen-and-ink signatures.[RD]

This contract document, or fragments thereof, could be used as the basis for other contract documents, often with other parties. That is, this document would become a precedent document.

Meanwhile, as both parties planned their performance under the contract and assessed the other party's performance, they would refer to this document. Often, many different professions would use this contract. E. G., architects and general contractors would look at a contract for constructing a building. Accountants would view the contract document in the course of preparing financial statements of the firms involved. Various professionals would read the terms to extract information for management information systems, including those vended as contract management systems.

And, should there be a dispute, lawyers and others in each firm would read the contract document to determine their rights. And, if the dispute would go that far, it would form the basis of the actions by the arbitrator or judge.

A contract document of this form, prepared by lawyers, would often be the basis for a "ticket" or "adhesion contract." These are situations where a printed set of terms is offered to the buyer of a good or service on a take-it-or-leave-it basis. Examples include the terms on the back of a parking or travel ticket and the shrink-wrap terms included with the physical packing of software. Similarly, there are situations involving businesses and consumers such as housing and car finance as well as most insurance contracts where there are printed terms but there is no negotiation. In some cases, contract assembly operations might customize the contract document with selected service options.

Sometimes, these terms are offered online to be agreed to. Examples include End-User-License Agreements when one downloads software, and the terms to which one must agree to use popular travel and auction web sites. (We note that several members of the TC are working on software and the issues involved in assent to these agreements and mechanisms by which end-users can track the terms of the contracts to which they agree. That will not be further discussed in this specification.)

Either way, the drafting of these terms in XML would not differ significantly from the use case of contracts with negotiated terms.

Contract documents are often the basis for a library of terms, clauses, and draft contracts that may be the starting point for the creation of future contracts. Creation of new documents can be done by document management systems [meyer05] or computer-aided-drafting systems, specialized to the creation of contracts [daska95]. The latter would use metadata about the document fragments being assembled.

What is not covered

As this is the first version of the first specification from this TC, and the only attempt to our knowledge to provide a standard for XML and contracts, there are many use cases and possibilities for which we have not yet provided.
In many electronic commerce contracts, a large enterprise might set up a system for the procurement of particular goods and services. Invited parties join the system and assent to that enterprise's master contract. Another model would be a peer-to-peer exchange. The Universal Business Language and ebXML provide models for the XML transmission of the transactions under the enterprise which would be considered as specific contracts covered by the master contract.

Several members of the TC identified that one could form an XML model of the master document and its relation to the XML forming electronic transactions and subcontracts. This could be used to relate this to the real-world obligations and give meaning to the underlying XML forming the transactions in such a trading community. However, this specification does not reflect any of this use case and vision.

Also, a member of this TC and several researchers have proposed systems where computer systems automatically negotiate contracts. The parties would define the parameters that would be determined by the negotiated process. They may be business terms such as price, quantity and delivery terms. They could also be parameters of the legal terms such as the choice of law.

Each party would use an instance of the same negotiation server software. These would have the value function of the parties. The negotiation servers for the parties would exchange the message and presumably come with a "win-win" set of values for the parameters.

These parameters would have to be substituted into a contract template to determine the contract between the parties. The specification does not specifically address this technology vision. However, it does include markup for embedded values. This would be very useful for such technology.

This specification is aimed at a broad community over those preparing, negotiating and using contract documents. The TC identified a need for customization for specific industries, e.g. insurance or financial terms. This would require specific vocabularies or could involve embedding XML from such sources as XBRL or UBL in a contract. However, this version of the schema does not address such issues.

Lastly, there is a broad class of research known as deontic contracts. These are formal expressions of the obligations of parties. E.g., party A would deliver goods, units of obligation can be related. E.g., Party B will pay $100.00 within ten days of receipt of goods from Party A. And, there have been representations of the handling of problem conditions. If the generator delivered does not work, party B has thirty days to notify party C. And if Party C does not affect a satisfactory repair within ten days of notification, it must take back the generator and refund the purchase price.

Again, this specification provides no representation of these obligations, short of allowing the parties to express them in words in the natural language.

The TC has noted that law firms would prefer to use the same schema for all the documents they prepare: contracts, wills, litigation documents, etc.

### Terminology

The key words must, must not, required, shall, shall not, should, should not, recommended, may, and optional in this document are to be interpreted as described in [RFC 2119].

- **application programmer**: The person or person(s) developing a system that examines or parses the XML conforming to this specification, or a version customized according to the recommendations herein. This would include the developer of style sheets developed in a style sheet language such as XSLT, whether they be used for producing a rendition or other XML [XSLT].

- **contract**: An agreement between parties that is intended to be legally enforceable. A contract may be oral, partly oral and partly written or wholly recorded in writing. The terms of a contract may be contained in many contract documents.
contract document  A document that records some or all of the draft or agreed contract terms. Contract terms are traditionally expressed in a natural language but it is assumed that some or all of the terms of a contract could be expressed in a deontic contract language (q.v.). As mentioned in the Scope section, this version of the specification does not include deontic contract language.

deonitc contract language  Means a language that can express the rights and obligations of parties to a contract in a form that can be parsed by software applications and processed with other data to determine state information about matters governed by the contract.

contract writer  The person or software tool preparing the XML described by this specification and defined by the Schemas being distributed with the specification.

embedded data value  This refers to a piece of information such as a product or service description, date, name, address, quantity or monetary amount that is embedded in the natural language expression of the contract terms.

machine readable information  This is information in the contract document that refers to information about contract rights, obligations, or states, that can be extracted from the document by a computer system. It includes information represented in deontic contract language, contract metadata and embedded data values. It does not refer to the computer readable characters in the text unless the meaning of that text can be determined by a computer system. For example, a monetary amount that can be read from the text is not machine readable information unless the system can determine useful information about the statement of that amount in the contract such as who must pay it, to whom it must be paid, and at what time is it to be paid for what purpose is it paid.

natural language  This includes the mode of expression of contract narrative as it is commonly written by lawyers.

precedent contract  This is a document that is used by the drafter of a new contract document as a starting point or template to assist in creating that new contract.

rendition  The output of a transformation or styling process by which XML documents conforming to a particular schema are rendered with human-readable layout in a particular format such as RTF, PDF, HTML or displayed by a computer using a particular kind of software.

schema customizer  The individual providing for changes, particularly additional elements or attributes to meet the needs of a set of users, e.g., a particular vertical market. As recommended in Customization, this would probably be done in the file eContracts.rnc

TC  This refers, in this document, to the Organization for the Advancement of Structured Information Standards Legal XML member Section eContracts Technical Committee.
Using the Supplied Schemas

Validating your Document

The examples below and the attached schema were tested using the Jing validator from thaiopensource.com in a Linux environment on the RelaxNG Schemas. Assuming one has this software installed (and in the CLASSPATH), one simply needs to type: `java -jar jing.jar -c path/eContracts.rnc contract xml file` providing the indicated path to the root schema file and the name of the file containing the Contract XML to be validated.

The section Customizing for the loose, standard, and tight models., below, will give you information on other versions of the schema representing the "tight" and "standard" model, discussed below.

Structure of the Schema Files

eContracts.rnc is the file against which one validates one's contract xml. It brings in the following files:

1. eContracts-core.rnc

This contains definitions for most of the elements below. This defines all items in the contract name space.

2. dc-metadata.rnc

The specification allows the user to enter a metadata element at the beginning of the contract. This contains several elements from the Dublin Core. These are defined in this file.

3. xi-include.rnc

This defines the xi:include element from World Wide Web Consortium ML Inclusions recommendation [http://www.w3.org/TR/2004/PR-xinclude-2004930].

Customizing for the loose, standard, and tight models.

Customization, below, explains in general, how to customize these schema. The Three Different Models, below, explains the meaning of loose, standard, and tight. The following elements allow the user to configure for loose, standard and tight models. Each of these is done by changing the appropriate grammar element:

Table 1. Container Model Customization Table

<table>
<thead>
<tr>
<th>element</th>
<th>Model to change</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>body.structure.model</td>
</tr>
<tr>
<td>back</td>
<td>back.structure.model</td>
</tr>
<tr>
<td>attachment</td>
<td>attachment.structure.model</td>
</tr>
<tr>
<td>item</td>
<td>item.structure.model</td>
</tr>
<tr>
<td>inclusion</td>
<td>inclusion.structure.model</td>
</tr>
</tbody>
</table>

There are certain technical issues in the specification that are not supported in DTD and XML Schema but are supported in Relax NG. [meyer05a]
The TC decided to use Relax as the schema in the specification, "at least for initial drafts." It was anticipated that a version in XML Schema and possibly DTD would be included. [aa2618]
The person configuring the schema can set each of these to either `tight.structure.model`, `standard.structure.model`, or `loose.structure.model`. We recommend that the schema customizer change them by adding lines in `eContracts.rnc` such as

```
body.structure.model        = tight.structure.model
back.structure.model        = tight.structure.model
attachment.structure.model  = tight.structure.model
item.structure.model        = tight.structure.model
inclusion.structure.model   = tight.structure.model
```

This sets all five text containers to require the tight structure model.

**Customization**

Customization explains the mechanics of adding a customization layer to the Relax schema, that is, where and how to add the statements. It also explains some of the things done when the schema was written to make many common customizations easy. The TC anticipates that users will make customization [MIN0216]. Some of these extensions will be for vertical areas such as real estate where the PRIA and MISMA standards are in use. [MIN0119]. UBL provides ‘a set of business elements’ and one might wish to combine the standard described in this specification to provide narrative terms and use UBL, or similar standards, to document 'business terms.' Although, the TC has chosen not to standardize how this might be done at this time, Customization provides examples of how it might be done.

**Expository Description of Writing Contract Documents Using This Schema**

**Components of contract**

`contract` is the root tag for contracts. A contract MAY contain the following, each of which is discussed in more detail below:

1. **metadata**
2. **title and subtitle** - the contract writer MUST include a **title**.
3. **contract-front**
4. **body** - this MUST appear
5. **back**
6. **attachments**

The following example shows a skeleton of a contract to illustrate these parts. To illustrate the overall structure, many of the elements are empty that would normally contain text:

**Example 1. level1.xml**

```xml
<?xml version="1.0" encoding="utf-8"?>
<contract xmlns="urn:oasis:names:tc:eContracts:1:0"
          xmlns:dc="http://purl.org/dc/elements/1.1/"
          >
```
An absolute minimal contract showing that a contract MAY have only a title and an empty body:

**Example 2. contra1.xml**

```xml
<?xml version="1.0" encoding="utf-8"?>
<contract xmlns="urn:oasis:names:tc:eContracts:1:0"
xmlns:dc="http://purl.org/dc/elements/1.1/"
xmlns:xi="http://www.w3.org/2001/XInclude"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsilocation="urn:oasis:names:tc:eContracts:1:0 ../XMLSchema/eContracts.xsd"
<title><text>Minimum XML sample</text></title>
<body>
<br>"/body>
</contract>
```
The Three Different Models

Customizing for the loose, standard, and tight models shows how to configure the three different "models:" "loose," "standard" and "tight."

In the loose model, one MAY intermix arbitrarily in a text container, [MIN0119], the two tags: block and item. This is shown by the example below in the body tag.

**Example 3. valid-loose.xml**

```xml
<?xml version="1.0" encoding="utf-8"?>
<contract xmlns="urn:oasis:names:tc:eContracts:1:0"
    xmlns:dc="http://purl.org/dc/elements/1.1/"
    xmlns:xi="http://www.w3.org/2001/XInclude">
    <title><text>Sample Loose model data</text></title>
    <body>
        <block> </block>
        <block> </block>
        <item>  </item>
        <item>
            <block></block>
        </item> 
        <block> </block>
        <item> </item>
        <block></block>
        <item> </item>
        <block> </block>
        <item> </item>
    </body>
</contract>
```

(The following are valid "text containers:" body, back, attachment, item, inclusion:

If a "text container" is set at the "tight: model," that means that the text containers MUST have: block in a text container or the text container MUST have only item. The contract writer MAY NOT mix these in the same text containers.

**Example 4. valid-tight.xml**

```xml
<?xml version="1.0" encoding="utf-8"?>
<contract xmlns="urn:oasis:names:tc:eContracts:1:0"
    xmlns:dc="http://purl.org/dc/elements/1.1/"
    xmlns:xi="http://www.w3.org/2001/XInclude">
    <title><text>Sample Tight model data</text></title>
    <body>
        <item>
            <block></block>
        </item>
    </body>
</contract>
```
If a "text container" is set at the "standard model," then the contract writer MAY put block and item elements in the same container. However, all the block MUST be put first. A text container MAY also have all block elements or it can have all item elements in the "standard model." In the example below, the information in body would be valid under the "standard model." However, back would not be valid; the blocks should come first.

**Example 5. valid-stand.xml**

```xml
<?xml version="1.0" encoding="utf-8"?>
<contract xmlns="urn:oasis:names:tc:eContracts:1:0"
xmlns:dc="http://purl.org/dc/elements/1.1/"
xmlns:xi="http://www.w3.org/2001/XInclude">
<title><text>Sample Standard model data</text></title>
<subtitle>body is valid as standard model. back is not.</subtitle>
<body>
  <block> </block>
  <block> </block>
  <item>  </item>
  <item>  </item>
</body>

<back>
  <item>  </item>
  <item>  </item>
  <block> </block>
  <block> </block>
</back>
</contract>
```

In [MIN0119], the loose model was identified as useful for quoted and attached material. It is also useful when one is converting a document that was sent as a file in a word processor format and for exchanging materials between environments. However, it makes it difficult to apply numbering in the output rendering.

The item is the basic building block of the document hierarchy and represents structures that may have a title or a number. These may be known in documents as "chapters," "parts," "sections," "clauses" and "subclauses." It is also intended to represent items in a list. The TC was careful not to use a name for elements that matches any of that list so as to avoid confusion or biasing the reader's idea of how they might be used:[meyer03]
The block is the container element for a structural or grammatical paragraph. The actual text is put in text elements which contain the elements.

Within a block, an item will be considered an element of a list. A block MAY contain text element directly, an item, as well as definition, table, inclusion, party or person-record. On the block, the user may have the attribute number-type which can be

- **manual**: The contract writer SHOULD indicate a list of items by putting a num element in each of the item elements directly within this block.
- **none**: None of the enclosing tags have a number-type. (This situation is not defined by the schema)
- **disc**: The application programmer SHOULD render list items by a disc or bullet.
- **line**: List items are marked by a line or dash.
- **number**: The application programmer SHOULD render list items preceded by numerals ('1', '2', '3', ...)
- **loweralpha**: The application programmer SHOULD render the list items by proceeding them with 'a', 'b', 'c', .... 'z', 'aa', 'ab', 'ac', 'ad' ... 'za', 'zb', 'zc'...zz'
- **upperalpha**: The application programmer SHOULD render list items proceeded by 'A', 'B', 'C', ... 'Z', 'AA', 'AB', 'AC', 'AD'... 'ZA', 'ZB', ...'ZZZ'
- **lowerroman**: The application programmer SHOULD render list items as 'i', 'ii', 'iii', 'iv'
- **upperroman**: The application programmer SHOULD render list items as 'I', 'II', 'III', 'IV'...

### Conditional processing

Often, the creator of a form contract needs to provide for certain text to be included in certain circumstances. E. g., certain riders are only included in insurance policies when the customer pays for them. Each jurisdiction might happen to require only certain text or disclaimers.

The contract writer indicates this by putting the attribute: condition on either a separate condition element that MAY appear in-line in the text element, or in the block or item tags.

The schema customizer MAY use the conditions element to define which conditions apply to a specific contract. The conditions SHOULD be added to the metadata element, if added at all. Alternatively, the application developer MAY look for the condition element in other locations. The conditions element SHALL contain one or more groups. Each SHALL have a name and a list of condition elements of the form <condition name="condition-value," Then any block, item or text containing a condition attribute that lists the condition-values in all of the condition elements in any group SHALL be included in the rendition.

The following example shows that the user has specified to include text that is specific for only in the US in the conditions in the metadata:

#### Example 6. conditions.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<contract xmlns="urn:oasis:names:tc:eContracts:1:0"
         xmlns:dc="http://purl.org/dc/elements/1.1/"
         
```

2 We have an example of the result of rendering a sample contract. That application program used the fields specified by num.
This shows that if there are multiple condition elements in the group within conditions and metadata, then all these MUST be specified for a block to be rendered:

**Example 7. conditions.xml**

```xml
<?xml version="1.0" encoding="utf-8"?>
<contract xmlns="urn:oasis:names:tc:eContracts:1:0"
         xmlns:dc="http://purl.org/dc/elements/1.1/"
         xmlns:xi="http://www.w3.org/2001/XInclude">
  <metadata>
    <conditions>
      <group><name>Jurisdiction</name>
        <condition name="US" />
        <condition name="AU"/>
      </group>
    </conditions>
  </metadata>
  <title><text>Sample of conditions</text></title>
  <body>
    <block condition="US">
      <text>A United States - specific statement in the contract</text>
    </block>
  </body>
</contract>
```
Here is how one can do an "or:"

**Example 8. conditionsOr.xml**

```xml
<?xml version="1.0" encoding="utf-8"?>
<contract xmlns="urn:oasis:names:tc:eContracts:1:0"
    xmlns:dc="http://purl.org/dc/elements/1.1/"
    xmlns:xi="http://www.w3.org/2001/XInclude">

    <metadata>
        <conditions>
            <group><name>Jurisdiction</name>
                <condition name="US"/>
            </group>
            <group><name>blah</name>
                <condition name="AU"/>
            </group>
        </conditions>
    </metadata>

    <title><text>Sample of conditions</text></title>

    <body>
        <block condition="US">
            <text>This will be included in the contract</text>
        </block>
        <block condition="AU">
            <text>This will be included in the contract</text>
        </block>
        <block condition="AU US">
            <text>This block will be included</text>
        </block>
        <block>
            <text>Included for all jurisdictions</text>
        </block>
    </body>

</contract>
```
The schema also provides that the schema customizer MAY customize these attributions by setting `condition-al.attributes` and `condition.attributes`. We provide options that the schema customizer and the application developer See Customization.

**id attributes**

The schema defines one id attribute. Note that this is defined as a XML ID while references to it are not currently defined an IDREF. The schema customizer MAY change this. This means that upon validation, there will be no error if a reference to an ID does not have a corresponding ID. This allows one to validate fragments of contract documents, particularly in clause libraries. [Min0216]

**xinclude**

The schema implements an `xi:include` element which SHALL implement [xinclude]. That is, it indicates that one file is to be brought into another. A sample use might be for a law firm that has certain standard boilerplate that is put in every lease. At the appropriate point in the xml text for the lease, one would have: `<xi:include href="boilerplate.xml"/>` Currently, only the `href` attribute is implemented in the schemas. As per the X-Include standard, we provide for an `xi:fallback`. This gives the information to be included when the indicated file is not located.

**field**

Fields SHALL be placed in a contract document in two scenarios. First, an application that will read the XML form of the contract document to extract information. For example, a trucking firm leases many trucks from many different financing firms. These firms provide the lease document in XML. The trucking company's management information department writes an application to read each one and extract the payment information to be put in the accounting system as a liability.

The other scenario is input into the firm. Example: an apartment complex has a database table containing the apartment numbers and monthly rent. They have an XML contract document acting as a template for the lease. It contains a field for the rent amount. Their application programmer develops a utility to locate the field for the rent amount and replaces it with the value looked up in the database. Thus, field can be used for input and output.

This is illustrated below using a field to specify the amount paid.

**Example 9. field.xml**

```xml
<?xml version="1.0" encoding="utf-8"?>
<contract xmlns="urn:oasis:names:tc:eContracts:1:0"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:xi="http://www.w3.org/2001/XInclude">
  <title><text>Sample of field</text></title>
  <body>
    <block>
      <text>In this licence:</text>
      <definition>
        <term>BNML Standard Schema</term>
      </definition>
      <block>
        <text>means the XML Schema called Pay</text>
        <field source="ida" name="PaymentAmount"></field>.
      </block>
    </block>
  </body>
</contract>
```
The field element contains the value of field and MUST support the following attributes that the contract writer MAY use:

1. **name**, usually used for extraction and processing purposes, e.g., the name of a column in a table.

2. **type**, field type, e.g., integer, string, currency.

3. **label**, intended for display purposes. For example, an application with a graphical user interface could search a contract document for all fields. It would display a label and text box to allow the user to enter the data. On start up, it SHOULD copy any text content to the text box and on closure, it SHOULD copy the text to which the user might have changed it back into the field.

4. **source**, source of the field information. It could be a database SQL query or an XPath statement.

5. **action**, the application programmer SHOULD use this to determine what action their application would take.

6. **length**, the length of the field when used for data entry. The units of measurement are not defined in this specification.

7. **class**

## Attachments

The contract structure provides for the attachments. These can contain appendices or other items associated with (or "attached") to the contract such as an engineering drawing. These contain one or more attachment elements. Each attachment may have metadata, a num, title and subtitle. Like the body, one can set the contents of the attachment as either the loose, standard, or tight model (q.v.). Observe, in the example, that there is only one attachment, numbered Appendix A, containing a title and two blocks.

### Example 10. attachments.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<contract xmlns="urn:oasis:names:tc:eContracts:1:0"
xmlns:dc="http://purl.org/dc/elements/1.1/
xmlns:xi="http://www.w3.org/2001/XInclude">
  <title><text>Sample of
    - attachments
    - attachment
    - num
    - title
    - block
    - text
  </text></title>
</contract>
```
<body></body>

<attachments>
  <attachment class="appendix" id="a1">
    <num>Appendix A</num>
    <title><text>Form of notice in Schema files</text></title>
    <block><text>They must be sent regular mail.</text></block>
    <block><text>Standard Schema</text>
  </attachment>
</attachments>

</contract>

---

**Metadata**

The contract document MAY contain metadata which is done using elements from the Dublin Core [dc]such as dc:title. These are intended as descriptions of any resource where a resource is 'anything that has an identity.' (These are from the name space http://purl.org/dc/elements/1.1.) This schema only defines the fields, title, creator, contributor, subject, publisher, date and rights from the standard. The date must match the ISO standard for dates, e.g. 2005-09-20. Note, the contract writer MAY include or not include any fields, including having an empty metadata tag in their contract document:

**Example 11. metadata.xml**

```xml
<?xml version="1.0" encoding="utf-8"?>
<contract xmlns="urn:oasis:names:tc:eContracts:1:0"
  xmlns:dc="http://purl.org/dc/elements/1.1/
  xmlns:xi="http://www.w3.org/2001/XInclude">
  <metadata>
    <dc:title>WIU CS Department notice 001</dc:title>
    <dc:creator>Julie Sasa</dc:creator>
    <dc:contributor>Keith Wallen</dc:contributor>
    <dc:subject>Scope of licence</dc:subject>
    <dc:publisher>WIU</dc:publisher>
    <!-- <dc:date>2005-09-20</dc:date> -->
    <dc:rights>Copyright, Elkera Pty Limited, 2005</dc:rights>
  </metadata>
  <title><text>Sample of metadata and its child elements</text></title>
  <body></body>
</contract>
```

---

3 As of August 27th, the check for validation is not working at the Western Illinois University site.
Party List

Contracts, conventionally, have a list of the parties to the contract at the front. [aug11] recognizes the differences in how this might be handled between the United States on one side and the United Kingdom and Australia, on the other. In the United Kingdom and Australia, this would often be of the form:

This agreement is made on the ___ day of ____

between the

first party (short-name)

and

second party (short-name)

[crc], [harrop03], [oxford]

Alternatively, in the United States, one can have:

This contract is made and entered into between first party, hereinafter referred to as "short-name" and second party, hereinafter referred to as "short-name" (Note that here this information is rendered as a single paragraph [CM], [MIN1226].)

Or taking the form from the American Institute of Architects standard contract:

AGREEMENT made as of the day of in the year

BETWEEN the short-name:

name and address

and the short-name:

name and address [aia]

the short-name is then used throughout the contract. E. g., One might write that "Acme Business Systems, Inc." would be referred to as "the Company" hereinafter.

Signature List

The party-signature element occurs within the matter in the back tag. This is the place that the contract writer SHOULD put in the markup for what would be considered material regarding signature. In this case where this is rendered to a physical document, this is where a person would apply a "pen-and-ink" signature and/or affix a seal. The TC recognized that it is important to provide for precise formatting of the signature block, noting "that people are quite fussy" as to how "this part of the contract is formatted."[MIN1226]. The contract writer MAY use a block to provide more flexible content and extra information. Some contracts will have one or more witnesses for each signature. The contract writer SHALL mark up each such set as a signature-record.

The party-signature element has an attribute, layout, with a default value of from-left. Alternatively, layout MAY be set to right-column-only (which means that the elements are rendered right justified.

The party-id is used, as specified in [Min1226], to link a party element within the list of parties to a particular line for that person's signature.
The contract writer MAY group signatory-record into groups using the signatory-group element. Within the signatory-group element, the contract writer MAY put zero or more blocks which SHALL be followed by one or more signatory-records. Each signatory-record MAY contain tags for other signatory-record or they MAY contain one or more signatory-records followed a single witness record or they MAY contain one or more witness tags followed by a signatory tag. These MAY be preceded or followed by block records. The signature-group MAY have an align-records attribute which, if included, MUST be horizontal (unless customized), meaning that the application programmer SHOULD render the signatures horizontally across the page, usually in two columns, or vertical, in which case the application programmer SHOULD render the signatures vertically across the page (unless customized as described in Customization). The contract writer MAY include the brace attribute, in which case the application writer SHOULD render a brace or thick column separator to the right. This will have the value block-only unless customized.

A signature or witness MAY contain signature-line and block tags. The signature-line consists of text and field lines.

Our first example consists of a single signatory-record containing both a signatory-record and a witness within the party-signature record:

**Example 12. signature.xml**

```xml
<?xml version="1.0" encoding="utf-8"?>
<contract xmlns="urn:oasis:names:tc:eContracts:1:0"
   xmlns:dc="http://purl.org/dc/elements/1.1/"
   xmlns:xi="http://www.w3.org/2001/XInclude">
  <title><text>Sample of :
      - back
      - party-signature
      - signatory-group
      - signatory-record
      - signatory
      - signature-line
  </text></title>

  <body></body>

  <back>
    <party-signature>
      <signatory-group>
        <block> </block>
        <signatory-record>
          <signatory id="T0001" xml:lang="ja">
            <signature-line id="I0001" xml:lang="en_US">
              <text>Mr. Signatory Jr.</text>
              <field>Field for a text.</field>
            </signature-line>
          </signatory>
        </signatory-record>
      </signatory-group>
    </party-signature>
  </back>
</contract>
```
A simple party signature by John Doe is here. Observe the initial block to give the text "Signed by John Doe" A signatory-record, signatory and signature-line

**Signed by John W. Doe**

John W. Doe

**Example 13. SIG1frag.xml**

Example 14. SIG2frag.xml
Observe the align-record="horizontal" which forces both Mr. Doe and Mr. Smith's signature lines to be side by side.

Similarly, one can include a witness element in a signatory-record:

Signed by John W. Smith in the presence of:

..................................................................................................................
..................................................................................................................
Witness .................................................. John W. Smith

Example 15. SIG3frag.xml

<party-signature><block><text>Signed by John W. Smith in the presence of:</text></block><signatory-record align-signatory-witness="horizontal"><witness><signature-line><field type="dotleader" length="50"/><text>Witness</text></signature-line></witness><signatory><signature-line><field type="dotleader" length="50"/><text>John W. Smith</text></signature-line></signatory></signatory-record></party-signature>

Observe the align-signatory-witness attribute which appears on the signatory-record. Here is a similar example, but where the align-signatory-witness attribute is specified to be vertical:

Signed by John W. Smith:

..................................................................................................................
John W. Smith

..................................................................................................................
Witness

Example 16. SIG4frag.xml

The user may put the attribute, `layout="right-column-only"` on the `party-signature` to have several lines of text as well as the place where the user will sign lined up on the right side of the page:

Accepted and agreed:
Signed on behalf of XYZ Corporation Limited, by John W. Smith, its authorized director:

<table>
<thead>
<tr>
<th>Signature of witness</th>
<th>John W. Smith</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name of witness</td>
<td></td>
</tr>
</tbody>
</table>

Example 17. SIG5frag.xml

```
<party-signature layout="right-column-only">
  <block>
    <text>Accepted and agreed:</text>
  </block>
  <block>
    <text>Signed on behalf of XYZ Corporation Limited, by John W. Smith, its authorized director:</text>
  </block>
</party-signature>
```

This example shows the use of the `brace` attribute on the `signatory-record`:

```
<signatory-record>
  <signatory>
    <signature-line>
      <field length="50" type="dotleader"/>
      <text>John W. Smith</text>
      <text>Director</text>
    </signature-line>
  </signatory>
</signatory-record>
```

Example 18. SIG9frag.xml

```
<signatory-records>
  <signatory-record>
  ...
  </signatory-record>
  ...
</signatory-records>
```

`signatory-records` can be nested to show delegation. This is rendered by a small indentation for the information within the nested `signatory-record`. 
"Lender"
Roadway Finance Company, a Delaware limited company
By: XYZ Limited Partnership, a Delaware limited partnership, its sole member
By: XYZ Office Properties Trust, a Delaware real estate investment trust, its general partner
By: /s/ John A. Doe
Name: John A. Doe
Title: Senior Vice President
1000 North Riverside Plaza
Chicago, IL 60606
Attention: General Counsel

Table
The table element is taken from [table] and is processed in accordance with [table1]. Here is an example showing how it is used in documents conforming to this contracts specification:

Example 19. table.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<contract xmlns="urn:oasis:names:tc:eContracts:1:0"
    xmlns:dc="http://purl.org/dc/elements/1.1/
    xmlns:xi="http://www.w3.org/2001/XInclude">
  <title><text>Sample of
    - table
    - tgroup
  </text></title>
</body>
```
In-line text elements

The standard allows for formatting, similar to that in Docbook or HTML, of individual characters or phrases. These are:

em  Text to be emphasized. The exact rendering, e.g., bold or italic, is determined by the application or rendering style sheet.

statutory-em  This means that the application program MUST render enclosed text as emphasized in accordance with the applicable statute. The TC identified this need in [Min0216].

strike  Text struck-through

Example 20. em.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<contract xmlns="urn:oasis:names:tc:eContracts:1:0"
    xmlns:dc="http://purl.org/dc/elements/1.1/"
    xmlns:xi="http://www.w3.org/2001/XInclude">
  <title><text>Sample of inline elements</text></title>
  <body>
    <block>
      <text>The location to deliver <em>the item</em> <strike>has to be one of <statutory-em>designated terminals.</statutory-em></strike> will be any places.</text>
    </block>
  </body>
</contract>
```

sub  Text that is part of a subscript.
sup Text that is superscripted.

**Example 21. subsup.xml**

```xml
<?xml version="1.0" encoding="utf-8"?>
<contract xmlns="urn:oasis:names:tc:eContracts:1:0"
    xmlns:dc="http://purl.org/dc/elements/1.1/"
    xmlns:xi="http://www.w3.org/2001/XInclude">
  <title><text>Sample of sub and sup</text></title>
  <body>
    <block>
      <text>The Elkera<sup>®</sup> Business Narrative Markup Language (BNML<sup>â€“</sup>)</text>
    </block>
    <block>
      <text>A<sub>10</sub></text>
    </block>
  </body>
</contract>
```

phrase The text MAY contain phrase

**Example 22. inclusion.xml**

```xml
<?xml version="1.0" encoding="utf-8"?>
<contract xmlns="urn:oasis:names:tc:eContracts:1:0"
    xmlns:dc="http://purl.org/dc/elements/1.1/"
    xmlns:xi="http://www.w3.org/2001/XInclude">
  <title><text>Sample of - item - num - block - inclusion - phrase</text></title>
  <body>
    <item>
      <num>2.1</num>
      <block number-type="none">
        <text>BNML Standard consists of 6 BNML specific files and 2 files that incorporate external</text>
      </block>
    </item>
  </body>
</contract>
```
features into the application.
The bnml-standard is the main file which includes all of the other files:</text>

<inclusion class="code-listing">
  <block number-type="none">
    <text>/BNML Standard</text>
    <item>
      <block number-type="none">
        <text>/RelaxNG</text>
        <item>
          <block>
            <text>bnml-standard.rnc</text>
            <text>bnml-document.rnc</text>
            <text>bnml-contract.rnc</text>
            <text>bnml-correspondence.rnc</text>
            <text>bnml-core.rnc</text>
            <text>bnml-structure.rnc</text>
            <text>dc-metadata.rnc</text>
            <text>xi-include.rnc</text>
          </block>
        </item>
      </block>
    </item>
  </block>
</inclusion>

<inclusion class="example">
  <block>
    <text>
      <phrase class="code">bnml-s-application name.rnc</phrase>
    </text>
  </block>
</inclusion>

example.html

Example 23. note.xml

<?xml version="1.0" encoding="utf-8"?>
<contract xmlns="urn:oasis:names:tc:eContracts:1:0"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:xi="http://www.w3.org/2001/XInclude">
  <title><text>Sample of note and note-in-line.</text></title>
<block>
  <text>Send me a shirt.<br/>
  <note-in-line>standard size.</note-in-line></text>
</block>

<block>
  <text>Send me desks as listed below.<br/>
  <note>
    <num>1</num><block><item></item></block>
  </note>
  <note>
    <num>2</num><block><item></item></block>
  </note>
</text>
</block>

note SHOULD NOT be rendered with surrounding text. The application programmer would typically render this as a note or footnote, but this decision is left to them. The application developer SHOULD render note-in-line with the text that surrounded it in the contract XML document.

object The contract writer MAY use this to bring in external objects such as pictures or audio wave forms. The contract writer SHALL provide a data and fallback element. The application program SHALL use the fallback if the data location is not reachable.

Example 24. fallback.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<contract xmlns="urn:oasis:names:tc:eContracts:1:0"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:xi="http://www.w3.org/2001/XInclude">
  <title><text>Sample of :
    - object
    - fallback
    - data
  </text></title>
  <body>
    <block>
      <text>
        <object type="video/mpeg">
          <!-- Video content goes here ->
        </object>
      </text>
    </block>
  </body>
</contract>
```
Real World Example

We provide a software license agreement to exemplify how this might be used in the "real world." In the package of schemas, there is the file, BNML_public_licence_v1.00.PDF It was rendered by an application from BNMLS-schemaPublicLicence.xml, also provided. Observe, at lines five to ten, the meta data. These have the prefix dc as they come from the Dublin Core. The provided schemas contain definitions of the elements that are relevant to this contract standard. (These, of course, are not shown in the rendering.)

On lines eleven to thirteen, observe the title and subtitle (on line fourteen) which are rendered at the top of the PDF.

The introductory material (lines fifteen to twenty-five) is kept in the contract-front and contains a single paragraph marked up as a block containing only a single text element.

The remainder of the contract is a series of numbered sections, item, in the body. Observe these starting at line 28. Each item is manually numbered.

Example 25. BNMLSContextMenuLicence.xml

1: <?xml version="1.0" encoding="utf-8"?>
2: <contract xmlns="urn:oasis:names:tc:eContracts:1:0"
4:  xmlns:xi="http://www.w3.org/2001/XInclude">
5:  <metadata>
7:   <dc:publisher>Elkera</dc:publisher>
8:   <dc:creator>Peter Meyer</dc:creator>
10:  </metadata>
11:  <title>
12:   <text>BNML Schema Public Licence</text>
13:  </title>
Elkera Pty Limited is the original developer of the XML schema and accompanying documents described as the BNML Schema. The BNML Schema is designed to be adapted and extended for use in a wide variety of applications. This document sets out the terms on which any person may copy, use, reproduce, create derivative works from and sublicence, the BNML Schema. Use of the BNML Schema other than as authorised under this licence or under applicable law is prohibited.

Interpretation

In this licence:

- **BNML Standard Schema** means the XML Schema called BNML Standard and all explanatory or reference documentation provided, or associated, with that Schema, described in the notice required by an applicable law.

- **Contributor** means an individual or entity who creates or contributes to the creation of a Derivative work.

- **Derivative work** means a work that reproduces a substantial part of the Schema, or that is an adaptation of the Schema, including a version of the Schema, whether or not in the language, code or notation in which the Schema was originally expressed.

- **Elkera** means Elkera Pty Limited, Australian Company Number (ACN) 092 447 428.
<text>means a work which combines the Schema or portions of it with computer software or documentation not covered by this Licence.</text>

<block></block>

<definition>
<term>Licensor</term>
[block]
<text>means Elkera and each Contributor.</text>
[block]
</definition>

<definition>
<term>Rights</term>
[block]
<text>mean the rights:</text>
[item]
<num>(a)</num>
[block]
<text>to reproduce the Schema;</text>
[block]
</item>
[item]
<num>(b)</num>
[block]
<text>to create and reproduce Derivative works;</text>
[block]
</item>
[item]
<num>(c)</num>
[block]
<text>to publish the Schema;</text>
[block]
</item>
[item]
<num>(d)</num>
[block]
<text>to communicate the Schema to the public; and</text>
[block]
</item>
[item]
<num>(e)</num>
[block]
<text>to sublicence the Schema.</text>
[block]
</item>
</block>
</definition>

<definition>
<term>Schema</term>
[block]
<text>means the BNML Standard Schema and any Derivative works of a Contributor.</text>
[block]
</definition>

<definition>
<term>You</term> or <term>Your</term>
</definition>
means an individual or entity exercising rights under this Licence.

Grant by Elkera

Subject to the terms and conditions of this Licence, Elkera hereby grants You a worldwide, royalty-free, non-exclusive, perpetual (for the duration of the applicable copyright) licence to exercise the Rights.

Grant by each Contributor

Subject to the terms and conditions of this Licence, each Contributor hereby grants You a worldwide, royalty-free, non-exclusive, perpetual (for the duration of the applicable copyright) licence to exercise the Rights.

Your acceptance of this licence

Your rights to copy and use the Schema are derived solely from this Licence or from rights granted by law. By using or reproducing the Schema or by creating a Derivative work you accept and agree to be bound by all the terms and conditions of this Licence.

Trade marks

This License does not grant any rights to use the trade marks or trade names â€œBNMLâ€‌ or â€œBNML Standardâ€‌ or any other trade marks, service marks, logos belonging to Elkera (collectively trade mark, logo or trade name belonging to any Contributor. You agree not to use any
in or as part of the name of products derived from the Schema or to endorse or promote products derived from the Schema, except that you may use the name BNML as permitted by clause 4.2.

Nothing in this License will impair Elkera’s right to acquire, license, develop, have others develop for it, market and/or distribute technology or products that perform the same or similar functions as, or otherwise compete with, Derivative Works, Larger Works, technology or products that You may develop, produce, market or distribute.

The Schema may be distributed only under the terms of this Licence. You must include a copy of this Licence with every copy of the Schema You distribute.

You may not impose a limitation or restriction on the rights of any other individual or entity in the use or licensing of the Schema that is not set out in this Licence but you are not responsible for enforcing compliance by third parties with this Licence.

You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of the Schema. However, You may not impose any such warranty, support, indemnity or liability obligation on any other Licensor or contributor of the Schema.
hereby agree to indemnify each other Licensor for any liability incurred by them of warranty, support, indemnity or liability terms You offer.

You may create a Larger work by combining the Schema with other works not this Licence and distribute the Larger work as a single product. However, You must the terms of this Licence in respect of the Schema.

You must include a notice substantially in the form of the applicable notice in each file of the Schema You distribute. If it is not possible to include notice in a particular file due to its structure, You must include the notice in the package of files where a user would be likely to see it. If You have created a Derivative work, You may add Your name and copyright statement as a Contributor to the notice described in.

You must include a notice substantially in the form of the applicable notice in each file of the Schema You distribute. If it is not possible to include notice in a particular file due to its structure, You must include the notice in the package of files where a user would be likely to see it. If You have created a Derivative work, You may add Your name and copyright statement as a Contributor to the notice described in.

You may create a Larger work by combining the Schema with other works not this Licence and distribute the Larger work as a single product. However, You must the terms of this Licence in respect of the Schema.
This is a modification that can be validated against BNML Standard using an XML parser. This occurs when the modification only adds attribute value enumerations, changes default values for attributes, prescribes the data type of attributes or removes non-required elements, including a BNML document type.

This is a modification that cannot be validated against BNML Standard but which retains the core patterns defined in the Schema file distributed by Elkera. These core patterns are the recursive item element, the text elements and the item element as a direct child of the block element. These are further explained in the <reference>Guide to BNML Schema configuration</reference> or similar guide distributed with the Schema by Elkera.

This is any other modification that is not a BNML Subset or a BNML Variant.

You must not use the prefix "BNML" in the name of a schema that is a Derivative work unless the schema is a BNML Subset or a BNML Variant.

If You create a modified version of BNML Standard, whether a BNML Subset or a BNML Variant, You must name the Schema so created in such a way that it is clearly differentiated from BNML Standard. The <reference>Guide to BNML Schema configuration</reference> or similar guide distributed with the Schema by Elkera includes recommended naming conventions.
Changes to this Licence

Elkera may publish revised versions of this Licence from time to time. Each revision is given a distinguishing number.

A new version of the Licence will adhere to open source licensing principles and will not alter the terms under which You may use the Schema. However, You may choose to use the Schema under the terms of a revised version of this Licence.

No warranty

Except as expressly required by law, the Schema is provided under this licence on an isâ basis, without warranty of any kind, either expressed or implied, including, without limitation, warranties that the Schema is free of defects, merchantable or fit for a particular purpose.

You understand that although Elkera and each Contributor grants the Licence are provided by Elkera or any Contributor that the Schema does not infringe the intellectual property rights of any other entity. Elkera and each Contributor disclaim any liability to You for claims brought by any other entity based on infringement of property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, You hereby assume sole responsibility to secure any other intellectual rights needed, if any. For example, if a third party patent license is required to allow You to distribute the Schema, it is Your responsibility to acquire that license before distributing the Schema.

Elkera and each Contributor represent that to its knowledge it has sufficient copyright rights in its contribution to the Schema, if any, to grant the copyright license set forth in this Licence.

Termination

This Licence and the rights granted by it will terminate automatically if You comply with the terms of this Licence and fail to cure the breach within 30 days of becoming aware of the breach. All sublicences to the Schema which are properly granted survive termination of this Licence. Provisions which, by their nature, must remain in effect after termination of this Licence survive that termination.
To the full extent permitted by applicable law, and except for any liability arising from contrary mutual agreement referred to. 

If applicable legislation implies warranties or conditions, or imposes obligations or liability on the Licensor in respect of this Licence that cannot be wholly excluded, restricted or modified, the Licensor's liability is limited, to the full extent permitted by applicable legislation, at its option, to:

(a) in the case of goods, any one or more of the following:

(i) the replacement of the goods or the supply of equivalent goods;

(ii) the repair of the goods;

(iii) the payment of the cost of replacing the goods or of acquiring equivalent goods;

(iv) the payment of the cost of having the goods repaired; or

(b) in the case of services:

(i) the repair of the goods;
the supplying of the services again; or

(ii) the payment of the cost of having the services supplied again.

9. Moral rights

Except as otherwise agreed in writing by the Licensor, if the Licensor is the author of the Schema or a part of the Schema, then to the extent permitted by applicable law, the Licensor unconditionally and irrevocably consents to all acts or omissions permitted by this Licence that would otherwise infringe any rights of the Licensor under moral rights law of integrity of authorship in respect of the Schema. This consent applies whether the relevant acts or omissions occur before or after the consent is given, and is given for the benefit of You, Your licensees and successors in title, and anyone authorised by You or an assignee to commit the relevant acts or omissions.

Assignment of Elkera's rights under this Licence

Without limitation of Elkera's right to transfer its intellectual property in the Schema, Elkera may assign the copyright in the Schema created by it and all its rights under this Licence, including the right to change the terms of this Licence, to a standards body whose objects promote open source software licensing.

Entire agreement

This Licence represents the entire agreement between You and the Licensor with respect to the licensing of the Schema. To the full extent permitted by applicable laws, the
understandings, agreements or representations with respect to the use of the Schema not specified in this Licence.

No waiver

No term or provision of this Licence is deemed to be waived and no breach consented to unless the waiver or consent is in writing and signed by the Licensor against whom the waiver or consent is asserted.

No term or provision of this Licence is deemed to be waived and no breach consented to unless the waiver or consent is in writing and signed by the Licensor against whom the waiver or consent is asserted.

If a provision of this Licence is invalid or unenforceable under applicable law, the validity or enforceability of the remaining terms of this Licence is not affected and without further action by the parties to this Licence, the invalid or unenforceable provision must be re-formed to the minimum extent necessary to make it valid and enforceable.

This construction, validity and performance of this License is governed by the laws of New South Wales, Australia.

The first section has several definitions, for which the schema provides definition elements. Observe these starting on line 34. The number of the clause is associated with the item. You will observe the match using ide9815a73-d1da-4ff0-8f04-19cd3c5af8bb involving two locations. It occurs on the item entitled Changes to this License (line 338). It also appears in the reference and field in line 259.

Also, observe in the (line 79) definition of Rights, that there is a text followed by several item tags. Each of these contain a num tag with an alphabetic sequencing such as (a), (b), etc.

At line 280, we observe the block for BNML Subset, BNML Variant, etc. Here, the number-type was set to none, corresponding to these being rendered with no numbers. The citation on line 332 was rendered in italics.

As stated in Attachments, the contract writer MUST enclose attachment in the attachments. You can see this in line 541. Note the use of the class="appendix" attribute on the attachment element. This attribute might affect how the attachment is rendered.

**Customization**

The [Relax], Section 9.2, shows how one can add to an attribute list or grammar. We recommend that they be added to the main relax schema for this specification, eContracts.rnc. Then, add the lines needed to add the modifications you wish. This is illustrated below. (I removed some of the comments in eContracts-core.rnc, as well as the definition of the WAI compatibility options, to save space.)

**Example 26. Modified.rnc**

default namespace ec = "urn:oasis:names:tc:eContracts:1:0"
namespace xs = "http://www.w3.org/2001/XMLSchema"
namespace xi = "http://www.w3.org/2001/XInclude"
namespace a = "http://relaxng.org/ns/compatibility/annotations/1.0"

datatypes xsd = "http://www.w3.org/2001/XMLSchema-datatypes"

grammar {

# This section includes Dublin core metadata elements into the application.
#
include "dc-metadata.rnc"

# INCLUDE eCONTRACTS CORE
include "eContracts-core.rnc" {

    metadata.content &= dc.metadata.content

    # change the structure model patterns
    body.structure.model        = tight.structure.model
    back.structure.model        = tight.structure.model
    attachment.structure.model  = tight.structure.model
    item.structure.model        = tight.structure.model
    inclusion.structure.model   = tight.structure.model

}

div {

    # add the condition attribute to all eContract elements.
    conditional.attributes &= condition.attribute
    inline.content.inner |= conditional

    # add the conditions structure to the metadata element
    metadata.content &=

        # define the conditions element and its children
        element conditions {
            element group {
                element name { xsd:string },
                element condition {
                    empty,

}}}
attribute name { xsd:string },
attribute default { xsd:boolean }?
}+
}
)?
}

div {
include "xi-include.rnc"
item.reuse.model |= xiInclude
}

# # # test addition to add a and b elements to item
PaySomeone = element PaySomeone {
xsd:integer
}
div {
block.item.attlist.extensions &= attribute a {xsd:string}?
block.item.attlist.extensions &= attribute b {xsd:string}?
text.content.inner |= PaySomeone
}

The above shows how one can change to the tight model to add two attributes, a and b to the item element. These are added to block.item.attlist.extensions. Note that item is defined in two different places in eContracts-core.rnc. Here is a sample contract showing the use of the a and b attributes on item tags as well as including PaySomeone in the text tag:

Example 27. TestModification.xml

<?xml version="1.0" encoding="utf-8"?>
<contract xmlns="urn:oasis:names:tc:eContracts:1:0"
xmns:dc="http://purl.org/dc/elements/1.1/
xmns:xi="http://www.w3.org/2001/XInclude">
<title>
<text>Data validate against Loose model based on Mr. Meyer's email on Aug20.</text>
</title>
<body>
<block> </block>
</body>
If the schema customizer wishes to add a customization to the item that might appear directly within a block tag, they change block.item.attlist. On the other hand, if one wishes to provide a customization to an item that might appear anywhere else, one adds it to item.attlist.extensions. In examining the definition of item in eContracts-core.rnc, one sees that one can add new elements to item by changing item.structure.module. It would have "worked" if the schema customizer adds attributes by changing common.attributes, item.class.attribute, conditional.attributes, stop-contents.attribute, item.numbering.attributes, and item.attlist.extensions. However, these have special meanings and most are used elsewhere. Thus, the schema customizer SHOULD NOT make this change there. As you can see, the schema often uses the convention, element-name.attlist.extensions to add attributes to element-name. As another example, attachment.attlist.extensions is where the schema customizer MAY put additional attributes that are to appear in the attachment element. Many elements also provide a customization opportunity of the form element-name.class.list For example, the schema defines inclusion.class.attribute. Also, the following elements have places to enter numbering options:

Table 2. Container Model Customization Table

<table>
<thead>
<tr>
<th>element</th>
<th>Label to change</th>
</tr>
</thead>
<tbody>
<tr>
<td>attachment</td>
<td>attachment.numbering.attributes</td>
</tr>
<tr>
<td>back</td>
<td>back.numbering.attributes</td>
</tr>
<tr>
<td>background</td>
<td>background.numbering.attributes</td>
</tr>
<tr>
<td>block</td>
<td>block.numbering.attributes</td>
</tr>
<tr>
<td>contract</td>
<td>contract.numbering.attributes</td>
</tr>
<tr>
<td>entry</td>
<td>entry.numbering.attributes</td>
</tr>
<tr>
<td>inclusion</td>
<td>inclusion.numbering.attributes</td>
</tr>
<tr>
<td>item</td>
<td>item.numbering.attributes</td>
</tr>
</tbody>
</table>

The following modification file is built upon a shortened version of eContracts.rnc:

Example 28. ModifiedTwo.rnc

default namespace ec = "urn:oasis:names:tc:eContracts:1:0"
namespace xs = "http://www.w3.org/2001/XMLSchema"
namespace xi = "http://www.w3.org/2001/XInclude"
namespace a = "http://relaxng.org/ns/compatibility/annotations/1.0"
datatypes xsd = "http://www.w3.org/2001/XMLSchema-datatypes"
It does add the attributes \( I_1 \) and \( I_2 \) to the inclusion element, by updating the the \( \text{inclusion.class.attribute} \) and \( \text{inclusion.attlist.extensions} \), respectively. Also, observe that it adds the attribute \( E \) to all elements. This minimized, but still valid, contract shows the effect on the possible attributes:

**Example 29. TestModification2.xml**

```xml
<?xml version="1.0" encoding="utf-8"?>
<contract xmlns="urn:oasis:names:tc:eContracts:1:0"
         xmlns:dc="http://purl.org/dc/elements/1.1/
         xmlns:xi="http://www.w3.org/2001/XInclude">
  <title><text>Modification Test</text></title>
  <body E="3">
    <item E="5">
      <inclusion I1="9"></inclusion>
      <inclusion I2="10"></inclusion>
    </item>
  </body>
</contract>
```

Here are the places at which the schema has specifically provided for customization and types of customization provided. To avoid repetition, the names are of the form \( \text{element-name.attlist.extensions} \) or \( \text{element-name.class.attribute} \)
common.attributes  The schema customizer MAY use this to make an attribute required or optional on every single element.

Currently, this is defined to be id.attributes, which is a single attribute, id, given the semantics of an XML ID. Obviously, the contract writer uses this to give a unique ID to any element, so that other elements can link to it. (As noted in id attributes, this schema does not declare references to be IDREF.)

These are other opportunities to customize the schema, which the schema customizer MAY use to affect more than one element:

text.content.inner  Some elements having the equivalent of the XML Schema mixed model, that is, a mixture of characters and tags will bring in text.content.inner. These are date-block and text. These include object, term, phrase, field, note, note-in-line, name, address, and date. They also include the elements in inline.content.inner.

text.content.inner defines the element tag names permissible as contents of the above list. Currently, the schema defines these to be the tags:

- object
- term
- phrase
- field
- note
- note-in-line
- name
- address
- date

inline.content.inner  These are intended to include markup that is needed to format characters in mixed text and include: text, reference, em, statutory-em, strike, sub, sup, field. The schema customizer changing inline.content.inner would add elements to the following elements that are defined to include this list:

- num
- subtitle
- terms
- term
- name
- address
- date
- note-in-line
- fallback
- reference
- citation
- phrase
- conditional
- em
- statutory-em
- strike
- sub
- sup

case-numbering.attributes

The elements in `contract-numbering.attributes` are added to the `contract` root tag—now defined as empty.

orient.attribute

These are added to `contract`, `attachment`, `inclusion`, `table`. `orient.attribute` is currently defined to be portrait or landscape. (The writers of this specification do not anticipate the schema customizer wanting to amplify this.)

metadata.content

This defines the possible `metadata` contents. Currently, `eContracts.rnc` sets `metadata.content` to include the elements defined in `dc-metadata.rnc`.

conditional.attributes

The contract writer MAY apply these attributes to `item`, `block`, `conditional`. Currently, `eContracts.rc` is configured to use the mechanism described in Customization.

stop-contents.attribute

If set to `below` (default value), then the application writer SHALL NOT put information for any of the the elements within this tag into the table of contents. Will be added to `attachment`, `item` which is not directly within `block`.

party-signature.layout.attribute

This will add attributes to the `party-signature` tag. Currently set to a single attribute, `layout`. That attribute will have one of the values given by the `party-signature.layout.values`, currently, `right-column-only` and `from-left`.

signatory-group.align-records-attribute

Added to the possible attributes for `signatory-group`. Currently defaulted at the attribute `align-records`, whose value is given by the `signatory-group.align-records.values`, which is now `horizontal` or `vertical`. 

45
signatory-group-brace.attribute
This is added to the signatory-group element and is currently set to the value brace, which has the possible values in signatory-group.brace.values, not set to block-only.

signatory-record-align-signatory-witness.attribute
This is added to the signatory-record element, and will have the values given by align-signatory-witness.values, now set at horizontal and vertical.

signatory-record.brace.attribute
Adds elements to the signatory-record element. It is currently set at brace which has the values given by brace.values which is to include only one possible value, block-only.

object.type.attribute
This is added to the object tag and is currently set to type. The Schema Customizer SHOULD use this to change the information relating to the MIME type of the object.

object.scale.attribute
This is added to the object tag and is currently set to scale. The Schema Customizer SHOULD use this for information relating to the scaling of the object.

object.rotate.attribute
This is added to the object tag and is currently set to rotate. The Schema Customizer SHOULD use this for information relating to the rotation of the object.

data.src.entry
This is added to the data tag and is currently set to src. The Schema Customizer SHOULD use this for information relating to the source of data for the enclosing object.

data.height.attribute
This is added to the data tag and is currently set to height. The Schema Customizer SHOULD use this for information relating to the height of the enclosing object.

data.width.attribute
This is added to the data tag and is currently set to width. The Schema Customizer SHOULD use this for information relating to the width of the enclosing object.

Dictionary

Common Attributes
These occur on all elements. They are summarized here once for brevity and to make the attributes that occur on many elements stand out.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>xsd:ID</td>
<td>none</td>
</tr>
</tbody>
</table>

Class Attributes
This attribute is given here once for brevity and to make the attributes that occur on many elements stand out.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>class</td>
<td>xsd:string</td>
<td>none</td>
</tr>
</tbody>
</table>
**condition.attribute**

This attribute is given here once for brevity and to make the attributes that occur on many elements stand out.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>condition</td>
<td>xsd:string</td>
<td>none</td>
</tr>
</tbody>
</table>

**orient.attribute**

This specifies whether the element contents should be rendered as **portrait** or **landscape**:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>orient</td>
<td>xsd:string</td>
<td></td>
</tr>
</tbody>
</table>

**Stop-Contents Attribute**

This stops the generation of table-of-contents entries for the elements contained within this element.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>stop-contents</td>
<td>xsd:string</td>
<td>below</td>
</tr>
</tbody>
</table>

**address**

**Synopsis**

**Content Model**

```
address ::=  

  • Zero or More of
    • conditional
    • em
    • field
    • reference
    • statutory-em
    • strike
    • sub
    • sup
    • text
```
Attributes

Common Attributes and Class Attributes

Additional Attributes:

• name

Description

This can contain an entire address (see example under party. Or it may contain part of an address, with each part having an appropriate category (see example below).

Attributes

Common Attributes Class Attributes

Additional attributes:

• "name"

Parents

text person-record

Children

The following elements occur inside: address

em, field, reference, statutory-em, strike, sub, sup, text

See Also

name, party and person-record

Examples

Example 30. party.xml

<?xml version="1.0" encoding="utf-8"?>
<contract xmlns="urn:oasis:names:tc:eContracts:1:0"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <title><text>Sample of :
      - contract-front
      - parties
      - party
      - person-record
      - name
      - address
    </text></title>
    </contract-front>

</contract>
attachment

Synopsis

Content Model

attachment ::=  
  
  • Sequence of  
    
    • Zero or one metadata  
    
    • Zero or one num  
    
    • Zero or one title  
    
    • Zero or more subtitle  
    
    • Zero or one of:  
      
      • contract  
      
      • If loose model is selected, sequence of: Zero or more of:  
        
        • block  
        
        • inclusion  
        
        • item  
        
        • xi:include  
      
      • If standard model is selected, sequence of:  
        
        • Zero or more inclusion
• Zero or more sequences of
  • One block
  • Zero or more inclusion

• Sequence of
  • One or more choice of
    • item
    • xi:include
  • Zero or more inclusion

• If tight structure model is selected, sequence of
  • Zero or more inclusion
  • Zero or one of:
    • One or more Sequences of
      • One block
      • Zero or more inclusion

• Sequence of
  • One or more sequences of
    • item
    • xi:include
  • Zero or more inclusion

Attributes

Common Attributes Class Attributes orient.attribute Stop-Contents Attribute

Description

This is an item attached to the contract which may take the form of an appendix (in which case it will probably be text obeying the model chosen for attachement.) It also might be a picture such as a blue print of the building to be built under the contract or another contract.
Attributes

Common Attributes Class Attributes orient.attribute Stop-Contents Attribute

Parents

attachment appears inside: attachments

Children

The following elements occur inside attachment: block, contract inclusion, item (outside of a block), metadata num subtitle title xi:include

Examples

Please see attachments

attachments

Synopsis

Content Model

attachments ::= 

• One or more attachment

Attributes

Common Attributes

Description

This is the container for one or more attachment elements.

Attributes

Common Attributes

Parents

attachments appears inside: contract

Children

The following elements occur inside attachments: attachment

See Also:

object and back
Examples:

Example 31. attachments.XML

```xml
<?xml version="1.0" encoding="utf-8"?>
<contract xmlns="urn:oasis:names:tc:eContracts:1:0"
         xmlns:dc="http://purl.org/dc/elements/1.1/"
         xmlns:xi="http://www.w3.org/2001/XInclude">
  <title><text>Sample of
           - attachments
           - attachment
           - num
           - title
           - block
           - text</text></title>
  <body></body>
  <attachments>
    <attachment class="appendix" id="a1">
      <num>Appendix A</num>
      <title><text>Form of notice in Schema files</text></title>
      <block><text>They must be sent regular mail.</text></block>
      <block><text>Standard Schema
              <reference href="http://www.elker. coma" print-url="http://www.elker.com/FindThis">
          </reference></text></block>
    </attachment>
  </attachments>
</contract>
```

back

back -- contains elements that follows the body of the contract including signatures.

Synopsis

Content model

back ::= 
  • Sequence of 
    • One of title 
    • Interleave of 
      • Zero or more of 
        • If loose model is selected, sequence of: Zero or more of: 
          • block
• inclusion
• item
• xi:include

• If standard model is selected, sequence of:
  • Zero or more inclusion
  • Zero or more sequences of
    • One block
    • Zero or more inclusion

• Sequence of
  • One or more choice of
    • item
    • xi:include
    • Zero or more inclusion

• If tight structure model is selected, sequence of
  • Zero or more of inclusion
  • Zero or one of:
    • One or more Sequences of
      • One block
      • Zero or more inclusion

• Sequence of
  • One or more sequences of
    • item
    • xi:include
    • Zero or more inclusion
• party-signature

• Zero or one date-block

Attributes
Common Attributes

Description
The back element is the content that follows the main part of the contract, represented by body.

Attributes
Common Attributes

Parents
This element contains back:contract

Children
The following elements occur in back:block, inclusion, itemregular xi:include, and title

Examples
The following shows the back within a contract including one possible way that signature material could appear:

Example 32. signature.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<contract xmlns="urn:oasis:names:tc:eContracts:1:0"
    xmlns:dc="http://purl.org/dc/elements/1.1/"
    xmlns:xi="http://www.w3.org/2001/XInclude">
    <title><text>Sample of :
      - back
      - party-signature
      - signatory-group
      - signatory-record
      - signatory
      - signatory-line
    </text></title>
    <body></body>
    <back>
        <party-signature>
        <signatory-group>
```
background

Synopsis

Content Model

background ::= 
  • Sequence of
    • Zero or one title
    • Zero or more occurrences of choice of:
      • item
      • xi:include

Attributes

Common Attributes
**Description**

This contains recitals and other "background" information.

**Attributes**

Common Attributes

**Parents**

background appears inside: contract-front

**Children**

The following elements occur inside background: item title xi:include

**Examples**

Please see parties.XML under parties.

**block**

block -- The container for a structural or grammatical paragraph.

**Synopsis**

**Content model**

block ::=  
  • Zero or more of:  
    • text  
    • definition  
    • table  
    • inclusion  
    • party  
    • person-record  
    • item inside a block

**Attributes**

Common Attributes

Additional attributes:  
  • number-type enumeration)
• "manual"
• "none"
• "disc"
• "line"
• "number"
• "loweralpha"
• "upperalpha"
• "lowerroman"
• "upperroman"

**Additional Constraints**

If this block contains a `number-type = manual`, then of the enclosing tags should have a `number-type` specified.

If this block contains a `number-type = manual`, then the contract writer SHOULD indicate a list item by putting a num in each of the item elements directly within this block.

**Description**

The block is the container element for a structural or grammatical paragraph. The actual text is put in text elements which contain the elements.

Within a block, an item will be considered an element of a list. A block MAY contain text element directly, an item, as well as definition, table, inclusion, party or person-record.

**Processing expectations**

On the block, the user may have the attribute `number-type` which can be

- **manual**: The contract writer SHOULD indicate a list items by putting a num element in each of the item elements directly within this block.
- **none**: None of the enclosing tags have a `number-type`. (This situation is not defined by the schema)
- **disc**: The application program SHOULD render List items by a disc or bullet.
- **line**: List items are marked by a line or dash.
- **number**: The application program should render List items proceeded by numerals ('1', '2', '3', ...) 
- **loweralpha**: The application program should render list items proceeded by 'a', 'b', 'c', .... 'z', 'aa', 'ab', 'ac', 'ad' ... 'za', 'zb', 'zc' ... 'zz'
- **upperalpha**: The application program should render list items proceeded by 'A', 'B', 'C', ... 'Z', 'AA', 'AB', 'AC', 'AD'... 'ZA', 'ZB', ...'ZZZ'

---

4 We have an example of the result of rendering a sample contract. That application program used the fields specified by num.
lowerroman The application programmer should render items as 'i', 'ii', 'iii', 'iv'

upperroman The application program SHOULD render list items as 'I', 'II', 'III', 'IV'...

**Attributes**

Common Attributes

Additional attributes:

- **number-type** Indicates that this is a list and it is to be processed as indicated

**Table 3. (enumeration)**

<table>
<thead>
<tr>
<th>number-type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;manual&quot;</td>
<td>The contract writer SHOULD indicate list items by putting a num element in each of the item tags directly enclosed within this block</td>
</tr>
<tr>
<td>&quot;none&quot;</td>
<td>The application program SHOULD render List items by a disc or bullet.</td>
</tr>
<tr>
<td>&quot;disc&quot;</td>
<td>List items are marked by a line or dash</td>
</tr>
<tr>
<td>&quot;line&quot;</td>
<td>The application program SHOULD render List items proceeded by numerals ('1', '2', '3', ...)</td>
</tr>
<tr>
<td>&quot;loweralpha&quot;</td>
<td>The application should render list items proceeded by 'a', 'b', 'c', .... 'z', 'aa', 'ab', 'ac', 'ad' ... 'za', 'zb', 'ze'...'zz'</td>
</tr>
<tr>
<td>&quot;upperalpha&quot;</td>
<td>The application program should render list items proceeded by 'A', 'B', 'C', ... 'Z', 'AA', 'AB', 'AC', 'AD'... 'ZA', 'ZB', ...'ZZZ'</td>
</tr>
<tr>
<td>&quot;number&quot;</td>
<td>The application program should render List items proceeded by numerals ('1', '2', '3', ...)</td>
</tr>
<tr>
<td>&quot;lowerroman&quot;</td>
<td>The application programmer should render items as 'i', 'ii', 'iii', 'iv'</td>
</tr>
</tbody>
</table>

**Parents**

These elements contain block: attachment definition inclusion itemregular item (inside a block), note party-signature signatory signatory-group signatory-record witness

**Children**

The following elements in occur in block: item (inside a block), definition inclusion block.item table text

**Examples**

See body or The Three Different Models.

**body**

body - The Body of the contract
Synopsis

Content model

body ::= 

• Zero or one of
  • title
  • If loose model is selected, sequence of: Zero or more of:
    • block
    • inclusion
    • item
    • xi:include
  
  • If standard model is selected, sequence of:
    • Zero or more inclusion
    • Zero or more sequences of
      • One block
      • Zero or more inclusion
    
    • Sequence of
      • One or more choice of
        • item
        • xi:include
    
    • Zero or more inclusion
  
  • If tight structure model is selected, sequence of
    • Zero or more of inclusion
    
    • Zero or one of:
      • One or more Sequences of
        • One block
        • Zero or more inclusion
- Sequence of
  - One or more sequences of
    - item
    - xi:include
  - Zero or more inclusion

**Attributes**

Common Attributes

**Description**

A `body` must appear in the XML for a contract document. As discussed in *The Three Different Models*, `body` is a text container and can be configured with a tight, standard or loose model.

**Parents**

These elements contain `body`: contract.

**Children**

The following elements occur in body: block, inclusion, item (outside of a block), title. `xi:include`

**Examples**

Here is a valid `body` under the loose model:

**Example 33. valid-loose.xml**

```xml
<?xml version="1.0" encoding="utf-8"?>
<contract xmlns="urn:oasis:names:tc:eContracts:1.0"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:xi="http://www.w3.org/2001/XInclude">
  <title><text>Sample Loose model data</text></title>
  <body>
    <block> </block>
    <block> </block>
    <item> </item>
    <item>
      <block></block>
    </item>
    <item> </item>
    <item>
      <block></block>
      <item> </item>
    </block></item>
    <block></block>
  </body>
</contract>
```
Here is a body under the standard model:

**Example 34. valid-stand.xml**

```xml
<?xml version="1.0" encoding="utf-8"?>
<contract xmlns="urn:oasis:names:tc:eContracts:1:0"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:xi="http://www.w3.org/2001/XInclude">
  <title><text>Sample Standard model data</text></title>
  <subtitle>body is valid as standard model. back is not.</subtitle>
  <body>
    <block> </block>
    <block> </block>
    <item>  </item>
    <item>  </item>
  </body>
  <back>
    <item>  </item>
    <item>  </item>
    <block> </block>
    <block> </block>
  </back>
</contract>
```

Here is a valid body under the tight model:

**Example 35. valid-tight.xml**

```xml
<?xml version="1.0" encoding="utf-8"?>
<contract xmlns="urn:oasis:names:tc:eContracts:1:0"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:xi="http://www.w3.org/2001/XInclude">
  <title><text>Sample Tight model data</text></title>
  <body>
    <item>
      <block></block>
      <block></block>
    </item>
  </body>
</contract>
```
citation

Synopsis

Content Model

citation ::= il

• Zero or more of
  • conditional
  • em
  • field
  • reference
  • statutory-em
  • strike
  • sub
  • sup
  • text

Attributes

Common Attributes

Description

This is text for a description of a work.

Processing Expectations

It is expected that this text SHOULD be rendered in-line.

Attributes

Common Attributes
Parents

citation appears inside reference

Children

The following elements occur inside citation:
em, field, reference, statutory-em, strike, sub, sup, text

Examples

See reference

colspec

Synopsis

This is used to provide formatting information and designators for a table within a tgroup.

Content Model

colspec is empty.

Attributes

Common Attributes

Additional Attributes:
• align
• colname
• colnum
• colsep
• colwidth
• rowsep

Description

The contract writer SHOULD provide one colspec for each column of their table. The contract writer MAY provide a name (colnum), a number (colnum) to designate their column as well as formatting information with the remaining attributes.

Attributes

Common Attributes

Additional Attributes:

align If a cell within this tgroup does not have an enclosed formatting style, then align it as indicated:
### Table 4. (enumeration)

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>left</td>
<td>align to the left (default)</td>
</tr>
<tr>
<td>right</td>
<td>align to the right (default)</td>
</tr>
<tr>
<td>center</td>
<td>center the text</td>
</tr>
<tr>
<td>justify</td>
<td>justify the text</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>colnam</th>
</tr>
</thead>
<tbody>
<tr>
<td>colnum</td>
</tr>
</tbody>
</table>

| colsep | If true, display a line to the right of each entry. |
|---|
| colwidth |

| rowsep | If true, draw a line below each cell. |
|---|

**Parents**

`colspec` appears inside `tgroup`.

**Children**

No elements occur inside `colspec`.

**See Also**

Some of this information may be overridden in `row` and `entry` elements.

**Example**

Please see `table`.

---

### conditional

**Synopsis**

**Content Model**

`conditional ::=`  
- Zero or More of  
  - `conditional`  
  - `em`  
  - `field`  
  - `reference`  
  - `statutory-em`
• strike
• sub
• sup
• text

Attributes

Common Attributes

Description

This is for text that will only be included in some versions of the document. See `condition.meta` and `conditions`.

Attributes

Common Attributes

Parents

text, address, citation, conditional, date, date-block, em, fallback, name, note-in-line, num, phrase, reference, statutory-em, strike, sub, subtitle, sup, term, terms, and text

xi:fallback

Children

The following elements occur inside: `conditional` and `em`

`field, reference, statutory-em, strike, sub, sup, text`

See Also

Examples

See `conditions`.

condition inside a group within the metadata

Synopsis

This defines one of the required strings that must match a `condition` attribute in a block or similar element in order for that content to be included in a particular instance of a contract.

Remember that this element is defined inside the group element and can only be used there. Please do not confuse it with the conditional element that can be used in inline content in the `text` and similar elements.

Content Model

condition is empty
Attributes

- "name"
- "default"

Description

Each of the group elements in the conditions element would have a list of these condition elements. These are matched against the condition attributes on such elements as the block and item or a conditional tag. Each will be a list of strings such as AU US. Only those elements having the condition attribute containing a list that matches any all of the condition of this or some other group will be included in the rendered content.

Attributes

name The value of this is matched against the list of strings in a conditional, which is within the regular text or a condition attribute on an element in the regular contract text.

default This is a boolean true or false

Parents

collection appears inside: group

Children

No elements occur inside condition.

See Also:

conditional.

Examples

See the two examples under conditions

conditions

Synopsis

Content Model

conditions ::=  
- One or more  
  - group

Attributes

none
Description

Each of the group elements in the conditions element would have a list of condition elements. These are matched against the condition attributes on such elements as the block and item. Each will be a list of strings such as AU US. Only those elements having the condition containing a list that matches any of the group's specified will match.

Thus, in conditions.XML, the metadata has specified that only those block that are marked for both Australia and US are to be included. In conditionsOr.xml, the metadata has specified that blocks containing either US or AU are to be included in the rendered contract.

Attributes

none

Parents

conditions appears inside: metadata

Children

The following elements occur inside conditions: group

See Also

conditional inside a regular inline element.

Examples

Example 36. conditions.XML

<?xml version="1.0" encoding="utf-8"?>
<contract xmlns="urn:oasis:names:tc:eContracts:1:0"
         xmlns:dc="http://purl.org/dc/elements/1.1/"
         xmlns:xi="http://www.w3.org/2001/XInclude">
  <metadata>
    <conditions>
      <group><name>Jurisdiction</name>
        <condition name="US" />
        <condition name="AU" />
      </group>
    </conditions>
  </metadata>
  <title><text>Sample of conditions</text></title>
  <body>
    <block condition="US">
      <text>A United States - specific statement in the contract</text>
    </block>
    <block condition="AU">
      <text>A Australia-specific statement in the contract</text>
    </block>
  </body>
</contract>
Example 37. conditionsOr.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<contract xmlns="urn:oasis:names:tc:eContracts:1:0"
          xmlns:dc="http://purl.org/dc/elements/1.1/
          xmlns:xi="http://www.w3.org/2001/XInclude">
  <metadata>
    <conditions>
      <group><name>Jurisdiction</name>
        <condition name="US" />
      </group>
      <group><name>blah</name>
        <condition name="AU"/>
      </group>
    </conditions>
  </metadata>
  <title><text>Sample of conditions</text></title>
  <body>
    <block condition="US">
      <text>This will be included in the contract</text>
    </block>
    <block condition="AU">
      <text>This will be included in the contract</text>
    </block>
    <block condition="AU US">
      <text>This block will be included</text>
    </block>
    <block>
      <text>Included for all jurisdictions</text>
    </block>
  </body>
</contract>
```

**contract**

contract (root element)

This is the root element of the eContracts schema.
Synopsis

Content Model

contract ::= 
  • Sequence of
    • Zero or one metadata
    • one title
    • Zero or more subtitle
    • Zero or one contract-front
    • One body
    • Zero or one back
    • Zero or more attachments

Attributes

Common Attributes Class Attributes orient.attribute

Description

This is the root element for any contract and this specification.

Parents

contract may appear inside attachment --it is the root of the contract space so its most visible location will not be inside of any element.

Children

The following elements occur inside contract: attachments back body metadata subtitle title

Examples

This illustrates the basic structure of a contract:

Example 38. level1.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<contract xmlns="urn:oasis:names:tc:eContracts:1:0"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:xi="http://www.w3.org/2001/XInclude">
  <title><text>Sample of the five elements at level 1 </text></title>
  <subtitle>This file is to test :
Here is an example showing a minimal contract which would consist only a title and a body:

**Example 39. contra1.xml**

```xml
<?xml version="1.0" encoding="utf-8"?>
<contract xmlns="urn:oasis:names:tc:eContracts:1:0"
   xmlns:dc="http://purl.org/dc/elements/1.1/"
   xmlns:xi="http://www.w3.org/2001/XInclude"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="urn:oasis:names:tc:eContracts:1:0 ../XMLSchema/eContracts.xsd">
   <title><text>Minimum XML sample</text></title>
   <body>
   </body>
</contract>
```
**contract-front**

**Synopsis**

**Content Model**

contract-front::=
- Sequence of
  - Choice of
    - Sequence of
      - Zero or more date-block
    - One parties
    - One or more block
    - Zero or one background

**Attributes**

Common Attributes Class Attributes

**Description**

This contains the date-block, parties and background elements that represent information and formatting that traditionally one puts at the beginning of a contract.

**Attributes**

Common Attributes Class Attributes

**Parents**

contract-front appears inside contract

**Children**

The following elements occur inside contract-front: background block date-block parties

**Examples**

**Example 40. level1.xml**

```xml
<?xml version="1.0" encoding="utf-8"?>
<contract xmlns="urn:oasis:names:tc:eContracts:1:0"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:xi="http://www.w3.org/2001/XInclude">
```

---

71
Sample of the five elements at level 1

This file is to test:
  - title
  - sub-title
  - contract-front
    - date-block
    - parties
    - party
    - body
    - back
    - attachments
    - attachment

Example 41. contract-front.XML

```xml
<?xml version="1.0" encoding="utf-8"?>
<contract xmlns="urn:oasis:names:tc:eContracts:1:0"
    xmlns:dc="http://purl.org/dc/elements/1.1/
    xmlns:xi="http://www.w3.org/2001/XInclude">
    <title> <text>Sample of:
      - contract-front
      - block
      - text
      - citation
      - background
    </text>
    </title>
</contract>
```
Elkera Pty Limited is the original developer of the XML schema and accompanying documents described as the <citation>BNML Schema</citation>. The schema is designed to be adapted and extended for use in a wide variety of applications.

data

Synopsis

Content Model

There is no content in the data model.

Attributes

Common Attributes

• height
• src
• width

Description

This is used to reference the source for a multimedia object, particularly pictures.

Attributes

Common Attributes

height This integer gives the height of the picture. (Units are not defined by this specification)
src This gives the file path or URI for the multimedia information
width This integer gives the width of the picture. (Units are not defined by this specification.)

Parents

data appears inside: object

Children

No elements occur inside data:
See Also
fallback

Examples
Please see fallback.XML under fallback

date

Synopsis

Content Model
date ::= 
  • Zero or More of 
    • conditional
    • em
    • field
    • reference
    • statutory-em
    • strike
    • sub
    • sup
    • text

Attributes
Common Attributes and Class Attributes

Description
This is used to markup date. This schema does not specify the format and the user may use the format appropriate.

Attributes
Common Attributes Class Attributes

Additional Attributes:
name
Parents
date-block and text

Children

The following elements occur inside: date
em, field, reference, statutory-em, strike, sub, sup, text

See Also
date-block name field party person-record

Examples

See level1.xml inside contract.

date-block

Synopsis

Content Model
date-block ::= 
• Zero or More of 
  • em 
  • field 
  • party 
  • person-record 
  • reference 
  • statutory-em 
  • strike 
  • sub 
  • sup 
  • text

Attributes

Common Attributes
Description

This is used to format a date on a separate line.

Attributes

Common Attributes

Parents

date-block appears inside: contract-front and date-block

Children

The following elements occur inside date-block: em, field, party, person-record, reference, statutory-em, strike, sub, sup, and text

See Also

parties

Example:

Please see level1.xml inside contract.

dc:contributor

Synopsis

This is from the Dublin Core Metadata Element Set [dc]

Content Model

dc:contributor ::= string

Attributes

This element has no attributes

Description

This element appears within the metadata. A person, organization, service or other entity which made a contribution to the content.

Attributes

This element has no attributes

Parents

dc:contributor appears inside metadata
**Children**

No elements occur inside `dc:contributor`.

**See Also**

`dc:creator` `dc:publisher`

**Examples**

Please see `metadata`.

**dc:creator**

**Synopsis**

This is from the Dublin Core Metadata Element Set [dc]

**Content Model**

`dc:creator := string`

**Attributes**

This element has no attributes.

**Description**

This element, that appears within the `metadata`, will contain the name of the "entity primarily responsible for making the content" of this contract. It may be a person, organization or service.

**Attributes**

This element has no attributes.

**Parents**

`dc:creator` appears inside `metadata`

**Children**

No elements occur inside `dc:creator`.

**Examples**

Please see `metadata`.

**dc:date**

**Synopsis**

This is from the Dublin Core Metadata Element Set [dc]
Content Model
dc:date ::= string

Attributes
This element has no attributes

Description
This element appears within the metadata. This is formally described to be any date associated with the "lifecycle" of the resource but typically it will be the date of creation or availability. Its format will consist of a four-digit year, a dash, a two digit month, and a two digit day of the month.

Attributes
This element has no attributes

Parents
dc:date appears inside metadata

Children
No elements occur inside dc:date.

Examples
Please see metadata

dc:description

Synopsis
This is from the Dublin Core Metadata Element Set [dc]

Content Model
dc:description ::= string

Attributes
This element has no attributes

Description
This element appears within the metadata.

Attributes
This element has no attributes
Parents
dc:description appears inside metadata

Children
No elements occur inside dc:description.

See Also
dc:subject

Examples
Please see metadata

dc:publisher

Synopsis
This is from the Dublin Core Metadata Element Set [dc]

Content Model
dc:publisher ::= string

Attributes
This element has no attributes

Description
This element appears within the metadata. This is the entity responsible for making the resource available.

Attributes
This element has no attributes

Parents
dc:publisher appears inside metadata

Children
No elements occur inside dc:publisher.

See Also
dc:creator

Examples
Please see metadata
**dc:rights**

**Synopsis**
This is from the Dublin Core Metadata Element Set [dc]

**Content Model**
dc:rights ::= string

**Attributes**
This element has no attributes

**Description**
This element appears within the metadata. Intellectual Property rights, including copyright, for the text of this contract.

**Attributes**
This element has no attributes

**Parents**
dc:right appears inside metadata

**Children**
No elements occur inside dc:rights.

**Examples**
Please see metadata

**dc:subject**

**Synopsis**
This is from the Dublin Core Metadata Element Set [dc]

**Content Model**
dc:title := string

**Attributes**
This element has no attributes

**Description**
This element appears within the metadata. It may often be a list of key words or selected from a "controlled vocabulary."
Attributes
This element has no attributes

Parents

dc:subject appears inside metadata

Children
No elements occur inside dc:subject.

See Also
dc:description

Examples
Please see metadata

dc:title

Synopsis
This is from the Dublin Core Metadata Element Set [dc]

Content Model
dc:title := string

Attributes
This element has no attributes

Description
This element, that appears within the metadata, will contain the name given to the contract and will be the name by which it will be formally known.

Please do not confuse this with title that is used throughout the rest of the contract.

Attributes
This element has no attributes

Parents

dc:title appears inside metadata

Children
No elements occur inside dc:title.
Examples

Please see metadata

definition

Synopsis

Content Model

definition ::= 
  • Sequence of 
    • Zero or More Of 
      • term 
      • terms

Attributes

Common Attributes Class Attributes

Description

This is for a formal structure containing terms and the text which defines either. One uses the terms when there are multiple terms associated with one unit of text.

Processing Expectations

Each definition SHOULD appear on a separate line and MAY appear with a hanging indentation.

Attributes

Common Attributes Class Attributes

Parents

definition appears inside: block.

Children

The following elements occur inside definition: block, term and terms

Examples

See field.XML inside field and body01.xml inside term.
**em**

**Synopsis**

**Content Model**

\[\text{em} ::= \]

- Zero or More of
  - conditional
  - \text{em}
  - field
  - reference
  - statutory-em
  - strike
  - sub
  - sup
  - text

**Attributes**

Common Attributes and Class Attributes

**Description**

The content is emphasized in some way. This specification does not specify how. However, the rendering application MAY be programmed to recognize \texttt{class=italic} or \texttt{class=bold}.

**Attributes**

Common Attributes and Class Attributes

**Parents**

\text{text}

**Children**

The following elements occur inside: \text{em}

\text{em, field, reference, statutory-em, strike, sub, sup, text}

**See Also**

statutory-em strike
Examples

Example 42. em.XML

```xml
<?xml version="1.0" encoding="utf-8"?>
<contract xmlns="urn:oasis:names:tc:eContracts:1:0"
         xmlns:dc="http://purl.org/dc/elements/1.1/"
         xmlns:xi="http://www.w3.org/2001/XInclude">
  <title><text>Sample of inline elements</text></title>
  <body>
    <block>
      <text>The location to deliver <em>the item</em> <strike>has to be one of <statutory-em>designated terminals.</statutory-em></strike> will be any places.</text>
    </block>
  </body>
</contract>
```

entry

Synopsis

A cell of a table

Content Model

entry ::= 

• Zero or more of
  • block
  • item (outside of a block)
  • xi:include
  • inclusion
  • signature-line

Attributes

Common Attributes

Additional Attributes

• align
• colname
Description

Reminder: table is taken from the CALS Exchange Table Model [table1],[table2].

Attributes

Common Attributes

Additional Attributes:

align If a cell within this tgroup does not have an enclosed formatting style, then align it as indicated:

Table 5. (enumeration)

<table>
<thead>
<tr>
<th></th>
<th>Align to the left (default)</th>
</tr>
</thead>
<tbody>
<tr>
<td>left</td>
<td>align to the left (default)</td>
</tr>
<tr>
<td>right</td>
<td>align to the right (default)</td>
</tr>
<tr>
<td>center</td>
<td>center the text</td>
</tr>
<tr>
<td>justify</td>
<td>justify the text</td>
</tr>
</tbody>
</table>

colname

colsep If true, display a rule to the right of this cell in the table.

namest

nameend

morerows

rowsep

valign This is the default for all row and entry within the tbody element.

Table 6. (enumeration)

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>top</td>
<td></td>
</tr>
<tr>
<td>middle</td>
<td>approximately vertically centered</td>
</tr>
<tr>
<td>bottom</td>
<td></td>
</tr>
</tbody>
</table>
Parents

entry appears inside: row

Children

The following elements occur inside entry: block item (outside of a block) xi:include inclusion signature-line

fallback

Synopsis

Content Model

fallback::=

• Zero or More of
  • conditional
  • em
  • field
  • reference
  • statutory-em
  • strike
  • sub
  • sup
  • text

Attributes

Common Attributes

Description

This element falls inside the object element. That would contain a data element that says where to retrieve the document. If the application cannot retrieve, find or render that object, it would go to the fallback element. The fallback element would normally contain other object or text data.

This recursion provides a series of sources of information to try. And the application will continue to try each object in turn, starting from the outer one.

Attributes

Common Attributes
Parents

fallback appears inside: object.

Children

The following elements occur inside fallback:

em, field, reference, statutory-em, strike, sub, sup, text

See Also

data

Examples

Example 43. fallback.XML

<?xml version="1.0" encoding="utf-8"?>
<contract xmlns="urn:oasis:names:tc:eContracts:1:0"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:xi="http://www.w3.org/2001/XInclude">
  <title><text>Sample of:
      - object
      - fallback
      - data</text></title>
  <body>
    <block>
      <text>
        <object type="video/mpeg">
          <data src="mympeg.mpeg" />
        </object>
        <fallback>
          <object type="image/png">
            <data src="myimage.png" />
            <fallback>The alternate text that should be displayed if both
            the video and image cannot be displayed.
          </fallback>
        </fallback>
      </text>
    </block>
  </body>
</contract>
field

Synopsis

Content Model

field ::=  
  • text

Attributes

Common Attributes and Class Attributes

Additional attributes:
  • action
  • label
  • length
  • name
  • source

Description

Fields are put in the contract document in two scenarios. An application will read the XML representation of the contract document to extract information. An Example scenario would be a firm that leases products from many different firms. These firms provide the lease document in XML using this specification. The company's management information systems department writes an application to read each one and extracts the amount to be paid each month. This is used to update the accounts receivable and liabilities section of the general ledger program.

The other scenario would be a firm generated form contracts from database information. An apartment complex has a table containing for each apartment, the legal description of the apartment and the monthly rent. The contract XML would have field elements for each of these two inputs. Again, their MIS department would write an application. In this case, it would look for the appropriate tuple in the database. It would replace the field elements by the corresponding fields or columns from that tuple.

Processing Expectations

In a printed contract, this might generate a printed space where information would be put. For example, in a signature-line, field appears as a place on which to place the signature. In other locations in a conventional document, a blank line may be provided on which the person may handwrite information.

In a display on a graphical user interface, such as a client-server program using Swing or in HTML on a browser, this may correspond to a text box in which the user may place information to be stored in a document.

Attributes

Common Attributes and Class Attributes

Additional attributes:
action Used to specify the action that the application program should take. (See Description.)

label This attribute is for display purposes. For example, a form-based graphical-user interface program could display a text box for each field element it finds. The label attribute would be used to label this box so the user knows what to put there.

length The length of the above field. (Units are not defined by this specification.)

name Not fully defined by the specification, but is intended to be used in processing. For example, if this application is extracting data from the contract and putting it in a database, it might be the name of the database column in which to place the data. Conversely, if the application is creating a contract, one for each tuple in a database table, it would copy the column indicated by this name attribute to the location for the field element.

source This would identify where to obtain the information to put for this box. For example, it might be a SQL or XPATH query.

type field type, e.g., integer, string, currency. (The exact values that would be accepted here would be dependent upon the application and is not defined by this specification.)

Parents

address, citation, conditional, date, date-block, em, fallback, name, note-in-line, num, phrase, reference, statutory-em, strike, sub, subtitle, sup, term, terms, and text xi:fallback

Children

The following elements occur inside: field: text

See Also

note, -in-linenote, reference, party, phrase, and term.

Examples

Example 44. field.XML

```xml
<?xml version='1.0' encoding='utf-8'?>
<contract xmlns='urn:oasis:names:tc:eContracts:1:0'
    xmlns:dc='http://purl.org/dc/elements/1.1/'
    xmlns:xi='http://www.w3.org/2001/XInclude'>

<title><text>Sample of field</text></title>

<body>
    <block>
        <text>In this licence:</text>
        <definition>
            <term>BNML Standard Schema</term>
            <block>
                <text>means the XML Schema called Pay</text>
                <field source="ida" name="PaymentAmount"/>
            </block>
        </definition>
    </block>
</body>
</contract>
```
**group**

**Synopsis**

Inside the metadata conditions element It provides a mechanism for selectively including certain content in a contract document.

**Content Model**

```
group::=  
  • Sequence of  
    • name (Inside the metadata conditions)  
    • condition inside a group within the metadata
```

**Attributes**

None

**Description**

This is used inside conditions within metadata. Please see that element for a detailed discussion. (Note that this element is defined within this location and thus cannot be used outside of this element.)

**Parents**

`group` appears only inside conditions.

**Children**

The following elements occur inside `group`: name (Inside the metadata conditions) and condition inside a group within the metadata

**Example**

Please see conditions.
**inclusion**

**Synopsis**

**Content Model**

inclusion ::=  
  • Sequence of  
    • Zero or one metadata  
    • Zero or one num  
    • title.  
  • If loose model is selected, sequence of: Zero or more of:  
    • block  
    • inclusion  
    • item  
    • xi:include  
  • If standard model is selected, sequence of:  
    • Zero or more inclusion  
    • Zero or more sequences of  
      • One block  
      • Zero or more inclusion  
  • Sequence of  
    • One or more choice of  
      • item  
      • xi:include  
      • Zero or more inclusion  
  • If tight structure model is selected, sequence of  
    • Zero or more of inclusion  
  • Zero or one of:
• One or more Sequences of
  • One block
  • Zero or more inclusion

• Sequence of
  • One or more sequences of
    • item
    • xi:include
  • Zero or more inclusion

### Attributes

Common Attributes Class Attributes orient.attribute

Additional Attributes:

- align

### Description

This is used for quotations, annotations, notes and examples and provides a title or number. It is thus useful to contain pictures and tables.

### Processing Expectations

This will be numbered and SHOULD be kept separate from the main contract provisions.

### Attributes

Common Attributes Class Attributes orient.attribute

Additional Attributes:

align Specifies the alignment, as given by ShortAlignment pattern:

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;left&quot;</td>
<td></td>
</tr>
<tr>
<td>&quot;center&quot;</td>
<td></td>
</tr>
<tr>
<td>&quot;right&quot;</td>
<td></td>
</tr>
</tbody>
</table>

Table 7. (enumeration)
Parents

inclusion appears inside: block entry attachment, back, block, inclusion and item

Children

The following elements occur inside inclusion: inclusion item, metadata, num item xi:include

See Also:

object

Examples:

Example 45. inclusion.XML

```xml
<?xml version="1.0" encoding="utf-8"?>
<contract xmlns="urn:oasis:names:tc:eContracts:1:0"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:xi="http://www.w3.org/2001/XInclude">
  <title><text>Sample of
  - item
  - num
  - block
  - inclusion
  - phrase
</text></title>
</contract>
```

<item>

<num>2.1</num>

</item>

<block number-type="none">

<text>BNML Standard consists of 6 BNML specific files and 2 files that incorporate external features into the application. The bnml-standard is the main file which includes all of the other files:</text>

</block>

</body>

<inclusion class="code-listing">

<block number-type="none">

<text>BNML Standard</text>

</block>

</inclusion>

<item>

<block number-type="none">

<text>RelaxNG</text>

</block>

</item>

<block>

<text>bnml-standard.rnc</text>

</block>

<text>bnml-document.rnc</text>
```
item (outside of a block)

The item is the basic building block of the document hierarchy and represents structures that may have a title or a number. These may be known in documents as "chapters," "parts," "sections," "clauses" and "subclauses." It is also intended to represent items in a list. The TC was careful not to use a name for elements that matches any of that list so as to avoid confusion or biasing the reader's idea of how they might be used.[meyer03]

Synopsis

Content Model

item ::= 

- Sequence of:
  - Zero or one metadata
  - Zero or one num
  - Zero or one title
  - .so model.XmL
Attributes

Common Attributes Class Attributes condition.attribute Stop-Contents Attribute

Description

This is the basic element that appears several times in the document hierarchy as listed below. Please see item (inside a block) as item is redefined when it appears directly inside a block.

Processing expectations

Formatted as a displayed block. Frequently causes a forced page break in print media. If stop-contents is not used, will be presented in the table of contents. May be suppressed based upon value of the condition element.

Attributes

Common Attributes Class Attributes condition.attribute Stop-Contents Attribute

Parents

attachment, back, block, inclusion and item

Children

inclusion item, metadata, num item xi:include

item (inside a block)

item (inside a block)

Synopsis

Content Model

item ::= 

• Sequence of

  • Zero or one metadata

  • Zero or one num

  • Zero or one title

  • Zero or more of

    • block

    • inclusion

Attributes

Common Attributes, Class Attributes and condition.attribute
Additional attributes:

- number-restart-index

**Description**

Relax NG supports the ability to define an element with a different content model when it is directly under some other element. That is, it creates a context-sensitive grammar. See [sperberg] and [Relax]. This feature is used only once in the grammar for this specification, an item has a different definition inside a block than when it appears in other elements.

**Process Expectations**

This is intended to be used as a list item. It will thus be numbered according to the number-type on the enclosing block. number, lowercase, uppercase, lowerroman and upperroman all create a sequence which will be changed by the number-restart-index attribute for this item.

**Attributes**

Common Attributes, Class Attributes and condition.attribute

number-restart-index To specify the starting index for the item when it is automatically renumbered (that is when the enclosing tag has a block with a number-type with a value other than manual or none.

**Parents**

As explained above, this type of item only occurs in a block

**Children**

The following elements occur in this special redefinition of item: metadata num title block inclusion

**See Also**

item (outside of a block)

**Examples**

See body.

**metadata**

**Synopsis**

**Content Model**

metadata ::= 

- Zero or more instances of
  - conditions
  - dc:contributor
Attributes
Common Attributes

Description
This contains metadata and is currently defined to include several elements from the Dublin Core standard defined in dc-metadata.rnc[dc] and the conditions element. The Dublin Core Standard are intended to describe 'anything that has an identity.' This schema defines the fields indicated above.

Processing Exceptions
It is intended that the information in metadata tags SHOULD NOT be rendered.

Attributes
Common Attributes

Parents
metadata appears inside: attachment item (inside a block) contract inclusion item (outside of a block)

Children

See Also:
um and title

Example

Example 46. metadata.XML

```xml
<?xml version="1.0" encoding="utf-8"?>
<contract xmlns="urn:oasis:names:tc:eContracts:1:0"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:xi="http://www.w3.org/2001/XInclude">
```
name

Synopsis

Content Model

name :=

• Zero or More of
  • conditional
  • em
  • field
  • reference
  • statutory-em
  • strike
  • sub
  • sup
  • text

Attributes

Common Attributes Class Attributes
Description

This provides a text content for a name of a person, organization or company. It is often used in a party-record to mark up the name of the party to a contract.

Attributes

Common Attributes

name

Parents

name appears inside: date-block person-record text

Children

The following elements occur inside name:

em, field, reference, statutory-em, strike, sub, sup, text

See Also:

address name

Examples

See party.xml under address and party.xml under parties

name (Inside the metadata conditions)

Synopsis

This is defined within the group element within the metadata. It can only be used here. Please do not confuse it with the name element used throughout the rest of this specification and which appears in many other places within the contract.

Content Model

name ::= string

Attributes

none

Description

This provides a way of labeling a set of condition inside a group within the metadata

Attributes

None
Parents
This name appears inside: group

Children
No elements occur inside this name

See also
condition inside a group within the metadata

Examples:
Please see inside the conditions element.

note

Synopsis

Content Model

note ::= 
  • Sequence of 
    • Zero or one num 
    • One or more block

Attributes

Common Attributes and Class Attributes

Description

This is used for a foot note, end note or other type of note. These would be notes that are NOT rendered in line with its surrounding text.

Attributes

Common Attributes and Class Attributes

Parents

note appears inside: date-block and text

Children

The following elements occur inside note: num and block
See Also

note-in-line

Example

Example 47. note.xml

<?xml version="1.0" encoding="utf-8"?>
<contract xmlns="urn:oasis:names:tc:eContracts:1:0"
    xmlns:dc="http://purl.org/dc/elements/1.1/"
    xmlns:xi="http://www.w3.org/2001/XInclude">
    <title><text>Sample of note and note-in-line.</text></title>
    <body>
        <block>
            <text>Send me a shirt.
                <note-in-line>standard size.</note-in-line>
            </text>
        </block>
        <block>
            <text>Send me desks as listed below.
                <note>
                    <num>1</num><block><item></item></block>
                </note>
                <note>
                    <num>2</num><block><item></item></block>
                </note>
            </text>
        </block>
    </body>
</contract>

note-in-line

Synopsis

Content Model

note-in-line ::= 

• Zero or More of
  • conditional
  • em
Attributes

Common Attributes and Class Attributes

Description

This is used to markup text to be in a note that would be kept in-line.

Processing Expectations

Typically, this would be rendered so the text comprising the note can be easily distinguished from the other text surrounding it.

Attributes

Common Attributes Class Attributes

Parents

address, citation, conditional, date, date-block, em, fallback, name, note-in-line, num, phrase, reference, statutory-em, strike, sub, subtitle, sup, term, terms, and text xi:fallback

Children

The following elements occur inside: note-in-line

em, field, reference, statutory-em, strike, sub, sup, text

See Also

note phrase reference

Examples

Example 48. note.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<contract xmlns="urn:oasis:names:tc:eContracts:1:0"
    xmlns:dc="http://purl.org/dc/elements/1.1/"
```
_send me a shirt.
_send me desks as listed below.

<table>
<thead>
<tr>
<th>num</th>
<th>content</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>item</td>
</tr>
<tr>
<td>2</td>
<td>item</td>
</tr>
</tbody>
</table>

**Synopsis**

**Content Model**

num ::= 

- Zero or More of 
  - conditional 
  - citation 
  - em 
  - field 
  - reference 
  - statutory-em 
  - strike 
  - sub
• sup
• text

Attributes

Common Attributes

Description

This is used to number things such as items within a block, attachment element within the attachments container element, inclusion note.

This allows the user to enter their own numbers rather than have automatic numbering. E. G. it would make less sense to include a num element within a series of item in a block with number-type=upperroman than one where the number-type was manual.

Processing Expectations

Obviously, the text content of the num element should be used as the number for the appropriate object.

Attributes

Common Attributes

Parents

num appears inside: attachment item inside a block inclusion item note

Children

The following elements occur inside num:

em, field, reference, statutory-em, strike, sub, sup, text

See Also:

block and item

Examples:

Please see parTY.xml in party inclusion and note

object

Synopsis

Content Model

object ::= 
• Sequence of
  • data
• object

**Attributes**

Common Attributes and Class Attributes

Additional Attributes:

• type
• scale
• rotate

**Description**

This is how one includes an image or multimedia file in one's contract document, e.g. a GIF file or Audio Wave form.

**Processing Expectations**

The data element indicates the source of the object, e.g. a file name or URL. It is possible that the application may not be able to locate the object or may not be able to render it once reached. For example, the application may not know how to deal with a format or the network may fail or the object data might be corrupted. If so, the fallback contains the item or text to be displayed. As discussed under fallback, this may be a cascading or recursive set of fallbacks.

**Attributes**

Common Attributes and Class Attributes

Additional Attributes:

• type This specifies the MIME type of the object, e.g. Image/Jpeg or Audio/Basic [Grand93]
• scale This allows the object to be scaled. This specification does not state how this should be written.
• rotate This is the rotation. A suggested standard is that this is the degrees in the anti-clockwise direction.

**Parents**

Object appears inside: date-block and text

**Children**

The following elements occur inside object: data and fallback

**example**

Please see fallback.
parties

Synopsis

Content Model

parties ::= 

• Sequence of
  
  • Zero or one title
  
  • One or more party

Attributes

Common Attributes and Class Attributes

Description

This is used to layout the parties as separate line under a title. This is common in Australia, New Zealand and the United Kingdom.

Processing Expectations

This SHOULD layout the title at the top but with the information for each party on a separate line. Here is an example of how the front might look for such a contract. Please note that this does NOT match any of the xml examples provided:

Example party-signature markup

Dated: 2006

Parties

  XYZ Limited of Suite 101, 12 Main Street, Sometown, State 00000. *(XYZ)*
  John W. Doe of 12 Long Street, Suburbia State 00000. *(JWD)*

Party

Attributes

Common Attributes and Class Attributes

Parents

parties appears inside: contract-front

Children

The following elements occur inside parties:title and party
See Also

party

Examples

Example 49. parTY.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<contract xmlns="urn:oasis:names:tc:eContracts:1:0"
    xmlns:dc="http://purl.org/dc/elements/1.1/"
    xmlns:xi="http://www.w3.org/2001/XInclude">
    <title><text>
        Sample of:
        - contract-front
        - parties
        - party
        - person-record
        - name
        - address
        - background
    </text></title>

    <contract-front>

    <date-block>Agreement dated:
        <field class="date" type="blank" name="contract_date" length="75mm"></field></date-block>

    <parties><title><text>Parties</text></title>
        <party><person-record><name>ABC Ventures Limited</name> having
            its office at <address>100 Main Street, Sydney, NSW 2000</address>
            </person-record>, hereafter referred to as
            "<term>General Partner</term>"
        </party>
        <party><person-record><name>John A. Doe</name> of
            <address>10 Ramrod Drive, Sydney, NSW 2000</address>
            and <person-record><name>John W. Smith</name> of
            <address>25 Pine Road, Plainsville, NSW, 0000</address>
            </person-record>, hereafter collectively referred to as
            "Limited Partners"
        </party>
    </parties>
    <background><title><text>Background</text></title>
        <item><num>A</num><block>
            <text>This is the text of the first recital</text></block></item>
        <item><num>B</num><block>
            <text>Text of the second recital</text></block></item>
    </background>
</contract-front>
</contract>
```
Example 50. parties.XML

```xml
<?xml version="1.0" encoding="utf-8"?>
<contract xmlns="urn:oasis:names:tc:eContracts:1:0"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:xi="http://www.w3.org/2001/XInclude">
  <title><text>Party Illustration</text></title>
  <contract-front>
    <date-block>Agreement dated:<field class="date" type="blank" name="contract_date" length=""/>
  </date-block>
  <body>
  </body>
</contract>
```

Remember that this does NOT match the picture above.

**party**

**Synopsis**

**Content Model**

party ::=  
  • Zero or More of  
    • text  
    • person-record  
    • term

**Attributes**

Common Attributes and Class Attributes

**Description**

This is a place holder for information about a party to the contract.

**Attributes**

Common Attributes Class Attributes

**Parents**

party appears inside: date-block parties text
Children

The following elements occur inside `party: text, person-record and term`

See Also

name address

Example

See `party.xml` in `parties`

**party-signature**

`party-signature` -- Place where a signature or seal will be applied to the contract document.

**Synopsis**

**Content model**

`party-signature` ::= 

- Sequence of
  - Zero or more block
  - One of
    - signatory-group
    - signatory-record
  
Zero or more block

**Attributes**

Common Attributes

Additional attributes:

- `layout` (enumeration)
  - "from-left"

- `party-id`

**Description**

This is the place for markup for what would be considered a signature. In the case, where this is rendered to a physical document, this is where a person would apply a "pen-and-ink" signature and/or affix a seal. There will be sets of signatures, for example, a signatory and witness (see `SIG3frag.xml` and `SIG4frag.xml` below), or two partners or officers of the same company (see `SIG2frag.xml` below).
**Processing Expectations**

If `align-records is horizontal`, then the signatures will be rendered in multiple columns, usually two. If set to `vertical`, then they will be rendered vertically.

**Attributes**

Common Attributes

<table>
<thead>
<tr>
<th><code>align-records</code></th>
<th>Table 8. (enumeration)</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;horizontal&quot;</td>
<td>render signatures horizontally along the page</td>
</tr>
<tr>
<td>&quot;vertical&quot;</td>
<td>render signatures vertically along the page</td>
</tr>
</tbody>
</table>

**Parents**

back

**Children**

The following elements occur in `party-signature`: block, signatory-group, signatory-record

**Examples**

Our first example consists of a single `signatory-record` containing both a `signatory-record` and a `witness` within the `party-signature` record:

**Example 51. signature.xml**

```xml
<?xml version="1.0" encoding="utf-8"?>
<contract xmlns="urn:oasis:names:tc:eContracts:1:0"
          xmlns:dc="http://purl.org/dc/elements/1.1/"
          xmlns:xi="http://www.w3.org/2001/XInclude">
  <title><text>Sample of :
          - back
          - party-signature
          - signatory-group
          - signatory-record
          - signatory
          - signatory-line
  </text></title>
  <body></body>
  <back>
    <party-signature>
      <signatory-group>
        <block> </block>  
    </party-signature>
  </back>
</contract>
```
A simple party signature by John Doe is here. Observe the initial block to give the text "Signed by John Doe" A signatory-record, signatory and signature-line

Signed by **John W. Doe**

**John W. Doe**

**Example 52. SIG1frag.xml**

```xml
<party-signature> <block><text>Signed by <name class="person">John W. Doe</name></text></block><signatory-record><signatory><signature-line><field type="blank" value="John W. Doe"></field></signature-line></signatory> </signatory-record></party-signature>
```

When there are two persons signing for one party, there are two signature-line tags within one signature-group:
Signed by the Limited Partners

John A Doe  
Partner  

John W. Smith  
Partner

Example 53. SIG2frag.xml

<party-signature>  
<block><text>Signed by the Limited Partners</text></block>  
<signatory-group align-records="horizontal">  
<signatory-record>  
<signatory> 
<signature-line> 
<field type="blank" length="50"/> 
<text>John A Doe</text> 
</signature-line> 
</signatory> 
</signatory-record>  
<signatory-record>  
<signatory> 
<signature-line> 
<field type="blank" length="50"/> 
<text>John W. Smith</text> 
</signature-line> 
</signatory> 
</signatory-group> 
</party-signature>

Observe the align-record=horizontal which forces both Mr. Doe and Mr. Smith's signature lines to be side by side.

Similarly, one can include a witness element in a signatory-record:

Signed by John W. Smith in the presence of:

........................................................................................................  
Witness  
........................................................................................................  
John W. Smith

Example 54. SIG3frag.xml

<party-signature>  
<block><text>Signed by John W. Smith in the presence of:</text></block>  
<signatory-record align-signatory-witness="horizontal">  
<witness> 
<signature-line> 
<field type="dotleader" length="50"/> 
<text>Witness</text> 
</signature-line> 
</witness> 
<signatory> 
<signature-line> 
<field type="dotleader" length="50"/> 
<text>John W. Smith</text> 
</signature-line> 
</signatory> 
</signatory-record> 
</party-signature>

Observe the align-signatory-witness attribute which appears on the signatory-record. Here is a similar example, but where the align-signatory-witness attribute is specified to be vertical:
Signed by John W. Smith:

........................................

John W. Smith

........................................

Witness

Example 55. SIG4frag.xml


The user may put the attribute, layout="right-column-only" on the party-signature to have several lines of text as well as the place where the user will sign lined up on the right side of the page:

Accepted and agreed:

Signed on behalf of XYZ Corporation Limited, by John W. Smith, its authorized director:

........................................

John W. Smith
Director

Example 56. SIG5frag.xml

<party-signature layout="right-column-only"><block><text>Accepted and agreed:</text></block><block><text>Signed on behalf of XYZ Corporation Limited, by John W. Smith, its authorized director:</text></block><signatory-record><signatory><signature-line><field type="dotleader" length="50"/><text>John W. Smith</text><text>Director</text></signature-line></signatory></signatory-record></party-signature>

This example shows the use of the brace attribute on the signatory-record:
Signed on behalf of XYZ Corporation Limited by John W. Smith, its managing director in the presence of:


Signature of witness

John W. Smith

Name of witness

Example 57. SIG9frag.xml

signatory-record's can be nested to show delegation. This is rendered by a small indentation for the information within the nested signatory-record.
"Lender"
Roadway Finance Company,
a Delaware limited company

By: XYZ Limited Partnership,
a Delaware limited partnership, its sole member

By: XYZ Office Properties Trust,
a Delaware real estate investment trust, its general partner

By: /s/ John A. Doe

Name: John A. Doe
Title: Senior Vice President

1000 North Riverside Plaza
Chicago, IL 60606
Attention: General Counsel

**person-record**

**Synopsis**

**Content Model**

```
person-record ::= 
  • Zero or more
    • address
    • field
    • name
    • term
    • text
```
Attributes

Common Attributes

Class Attributes

Additional attributes:

• party-id

Description

This is used to contain information about a person or other entity. It may be contained in a party record. It also may be intermixed in text in a text or date-block.

This is used to contain information about a person or other entity. It may be contained in a party record or it may be in such elements as text or phrase. Here, it would often have a link to a corresponding party element.

Attributes

Common Attributes

Class Attributes

Additional Attributes:

party-id  This would be a link to the corresponding party element.

Parents

person-record appears inside: date-block and party text

Children

person-record contains: address and field name term text

See Also

party

Examples

See party.xml and parties.XML in parties

phrase

Synopsis

Content Model

phrase ::=  

•  Zero or More of  

•  conditional  

•  em  

•  field
• reference
• statutory-em
• strike
• sub
• sup
• text

Attributes
Common Attributes and Class Attributes

Description
This might be used on something for which emphasis is placed in rendering. It may be used to explain something outside the normal flow of text. It may be a technical or foreign word.

Processing Expectations
Typically, would be rendered differently so as to give emphasis, possibly as indicated by class element.

Attributes
Common Attributes Class Attributes

Parents
address, citation, conditional, date, date-block, em, fallback, name, note-in-line, num, phrase, reference, statutory-em, strike, sub, subtitle, sup, term, terms, and text xi:fallback

Children
The following elements occur inside: phrase
em, field, reference, statutory-em, strike, sub, sup, text

See Also
note and -in-linenote reference phrase

Examples

Example 58. note.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<contract xmlns="urn:oasis:names:tc:eContracts:1:0"
    xmlns:dc="http://purl.org/dc/elements/1.1/"
    xmlns:xi="http://www.w3.org/2001/XInclude">
```
Sample of note and note-in-line.

Send me a shirt.

Send me desks as listed below.

1

2
**Attributes**

Common Attributes and Class Attributes

Additional attributes:

- href
- print-url

**Description**

This word is used to refer to other words, e.g. a printed document or a location on the world wide web.

**Processing Expectations**

If the contract document is printed, the external web site should be displayed next to the enclosed text, appropriately punctuated. Example: `<reference href="http://www.oasis-open.org/committees/documents.php?wg_abbrev=legalxml-econtracts">This specification</reference>` i.C MAY be rendered:

This specification (www.oasis-open.org/committees/documents.php?wg_abbrev=legalxml-econtracts)

**Attributes**

Common Attributes

Additional Attributes:

- href: This specifies the URL or other way of finding the object, as one would write it in HTML
- print-url: The text that should be printed near the enclosed text, when the contract document is printed.

**Parents**

`reference` appears inside: `address`, `citation`, `conditional`, `date`, `date-block`, `em`, `fallback`, `name`, `note-in-line`, `num`, `phrase`, `reference`, `statutory-em`, `strike`, `sub`, `subtitle`, `sup`, `term`, `terms`, and text `xi:fallback`

**Children**

The following elements occur inside `reference`:

- `citation`
- `em`
- `field`
- `reference`
- `statutory-em`
- `strike`
- `sub`
- `subtitle`
- `sup`
- `text`

**See Also**

term
Examples

Example 59. attachments.XML

```xml
<?xml version="1.0" encoding="utf-8"?>
<contract xmlns="urn:oasis:names:tc:eContracts:1:0"
  xmlns:dc="http://purl.org/dc/elements/1.1/
  xmlns:xi="http://www.w3.org/2001/XInclude">
  <title><text>Sample of
    - attachments
    - attachment
    - num
    - title
    - block
    - text</text></title>
  <body/>
  <attachments>
    <attachment class="appendix" id="a1">
      <num>Appendix A</num>
      <title><text>Form of notice in Schema files</text></title>
      <block><text>They must be sent regular mail.</text></block>
      <block><text>Standard Schema</text>
        <reference href="http://www.elkera.com" print-url="http://www.elkera.com/FindThis">can be read on the web easily!</reference>.</block>
    </attachment>
  </attachments>
</contract>
```

row

Synopsis

A row of a table

Content Model

trow ::= 
  • One or more
    • entry

Attributes

Common Attributes

Additonal Attributes:
• rowsep
• valign

**Additional Constraints**

There SHALL be no more **entries** within this **row** than indicated by the **cols** attribute in the enclosing **tgroup**.

**Description**

Reminder: **table** is taken from the CALS Exchange Table Model [table1],[table2].

**Attributes**

Common Attributes

Additional Attributes:

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>rowsep</td>
<td>If true, put a line below each cell.</td>
</tr>
<tr>
<td>valign</td>
<td>This is the default for all <strong>row</strong> and <strong>entry</strong> within the <strong>tbody</strong> element.</td>
</tr>
</tbody>
</table>

**Table 9. (enumeration)**

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>top</td>
<td></td>
</tr>
<tr>
<td>middle</td>
<td>approximately vertically centered</td>
</tr>
<tr>
<td>bottom</td>
<td></td>
</tr>
</tbody>
</table>

**Parents**

**row** appears inside **thead** and **tbody**.

**Children**

The following element occur inside **row: entry**

**Example:**

Please see **table**.

**signatory**

signatory - markup for the person who signs a document

**Synopsis**

**Content Model**

signatory ::= 

• Zero or more of
  • **signature-line**
• block

Attributes

Common Attributes

Description

This provides the blank line and printed information for a person who signs a contract.

Attributes

Common Attributes

Parents

signatory appears inside signatory-record

Children

The following elements occur inside: signatory

signature-line and block

See Also

witness

Examples

Please see party-signature for extensive examples.

signatory-group

Synopsis

Content Model

signatory-group ::= 
  • Sequence of
    • block
    • One or more signatory-record

Attributes

Common Attributes

Additional attributes:

• align-records(enumeration)
Description

This represents a group of related signatures, e.g., several officers for the same corporation or a witness and a signature. These are represented as the signature and witness tags, respectively.

Processing Expectations

The signature information will be aligned across the page if align-records is horizontal or vertically if specified as vertical. A brace or thick line will be to the right of the signatures if the brace is specified.

Attributes

Common Attributes Additional Attributes:

- **align-records**. Controls whether the enclosed signatures are laid out side-by-side or vertically.

<table>
<thead>
<tr>
<th>Table 10. (enumeration)</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;horizontal&quot;</td>
</tr>
<tr>
<td>&quot;vertical&quot;</td>
</tr>
</tbody>
</table>

- **brace** If block-only and the layout on the enclosing party-signature is NOT right-column-only, a brace or thick column separator is rendered to the right of the content in the left column.

<table>
<thead>
<tr>
<th>Table 11. (enumeration)</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;block-only&quot;</td>
</tr>
</tbody>
</table>

Parents

signatory-group appears in party-signature.

Children

The following elements occur in this special redefinition of signatory-group: block and signatory-record

See Also

signatory-record

Examples

See party-signature
signatory-record

Synopsis

Content Model

signatory-record ::= 

- Sequence of
  - block
- Choice of
  - signatory-record
    - Choice of
      - Sequence of
        - One or more signatory
      - Zero or more witness
  - Sequence of
    - One or more witness
    - One signatory
  - Zero or more block

Attributes

Common Attributes

Additional attributes:

- align-signatory-witness (enumeration)
  - "horizontal"
  - "vertical"

- brace (enumeration)
  - "block-only"
Additional Constraints

brace is ignored when contained the enclosing party-signature specified layout as right-column-only or if enclosed within a signatory-group element.

Description

This is used to bind one or more signatories as well as a possible witness. Its content can be in three forms:

- a single signatory-record This is used for sub-delegations. (See SIG12frag.xml under party-signature for an example.)
- One or more signatory elements. These may be followed by witness. The multiple signatory records reflect several people signing on behalf of one party. (See SIG2frag.xml under party-signature.
- One or more witness records followed by a signatory record. (See SIG3frag.xml under party-signature.

Processing Expectations

If one has signatory-record nested, then the inner one, representing a subdelegation, would be nested. See the screen shot next to SIG12frag.xml.

If a align-signatory-witness is horizontal the information in the signatory-record is laid out horizontally. If vertical then the information is laid out one under another. Please look at SIG3frag.xml and SIG4frag.xml for examples of align-signatory-witness being vertical and horizontal (under party-signature.

If brace is block-only, then there will be a brace or thick column separator to the right of the left column's content. See SIG9frag.xml for an example.

Attributes

Common Attributes

Additional Attributes:

align-signatory-witness

Indicates whether the witness and signatory are output horizontally or vertically.

Table 12. (enumeration)

<table>
<thead>
<tr>
<th>horizontal</th>
<th>The signature information is laid out across the page.</th>
</tr>
</thead>
<tbody>
<tr>
<td>vertical</td>
<td>The signature information is laid out with one item underneath another</td>
</tr>
</tbody>
</table>

brace

If set to block, a brace or thick column is applied to the right>

Table 13. (enumeration)

<table>
<thead>
<tr>
<th>&quot;horizontal&quot;</th>
<th>The signature information is laid out across the page.</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;vertical&quot;</td>
<td>The signature information is laid out with one item underneath another</td>
</tr>
</tbody>
</table>
Parents
The signatory-record appears in party-signature, signatory-group, and signatory-record

Children
The following elements occur in signatory-record: block, signatory, signatory-record, and witness

See Also
signatory

Examples
See party-signature

signature-line
signature-line creates the line on which a pen-and-ink signature is applied

Synopsis

Content Model
signature-line ::
    • Zero or more interleaves of
      • text
      • field

Attributes

Description
signature-line This creates the line on which a pen-and-ink signature is applied

Attributes

Parents
signature-line appears inside: entry signatory and witness

Children
The following elements occur inside signature-line: text and field
Examples

Please see party-signature for extensive examples.

statutory-em

Synopsis

Content Model

statutory-em ::= 
  • Zero or More of 
    • conditional 
    • em 
    • field 
    • reference 
    • statutory-em 
    • strike 
    • sub 
    • sup 
    • text 

Attributes

Common Attributes and Class Attributes

Description

This is used to mark up content that must be emphasized as per a particular statute. Presumably, the application program will render the contract emphasizing the text as required by that statute. The TC recognized this need in [Min0216].

Attributes

Common Attributes and Class Attributes

Parents

address, citation, conditional, date, date-block, em, fallback, name, note-in-line, num, phrase, reference, statutory-em, strike, sub, subtitle, sup, term, terms, and text xi:fallback

Children

The following elements occur inside: statutory-em
See Also

Examples

Example 60. em.XML

<?xml version="1.0" encoding="utf-8"?>
<contract xmlns="urn:oasis:names:tc:eContracts:1:0"
         xmlns:dc="http://purl.org/dc/elements/1.1/"
         xmlns:xi="http://www.w3.org/2001/XInclude">
   <title><text>Sample of inline elements</text></title>
   <body>
     <block>
       <text>The location to deliver <em>the item</em>
       <strike>has to be one of <statutory-em>designated terminals.</statutory-em></strike> will be any places.</text>
     </block>
   </body>
</contract>

strike

Synopsis

Content Model

strike ::= 

- Zero or More of
  - conditional
  - em
  - field
  - reference
  - statutory-em
  - strike
  - sub
  - sup
Attributes

Common Attributes and Class Attributes

Description

This is used to markup text that is to be striked through. This is commonly used to indicate changes between versions of documents.

Attributes

Common Attributes and Class Attributes

Parents

text

Children

The following elements occur inside: strike, em, field, reference, statutory-em, strike, sub, sup, text

See Also

sub sup

Examples

Example 61. EM.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<contract xmlns="urn:oasis:names:tc:eContracts:1:0"
    xmlns:dc="http://purl.org/dc/elements/1.1/"
    xmlns:xi="http://www.w3.org/2001/XInclude">
    <title><text>Sample of inline elements</text></title>
    <body>
        <block>
            <text>The location to deliver <em>the item</em> <strike>has to be one of <statutory-em>designated terminals.</statutory-em></strike> will be any places.</text>
        </block>
    </body>
</contract>
```
**sub**

**Synopsis**

**Content Model**

\[
\text{sub ::=}
\]

- Zero or More of
  - conditional
  - em
  - field
  - reference
  - statutory-em
  - strike
  - sub
  - sup
  - text

**Attributes**

Common Attributes and Class Attributes

**Description**

This is used to mark up content to be subscripted.

**Attributes**

Common Attributes and Class Attributes

**Parents**

address, citation, conditional, date, date-block, em, fallback, name, note-in-line, num, phrase, reference, statutory-em, strike, sub, subtitle, sup, term, terms, and text xi:fallback

**Children**

The following elements occur inside: sub

em, field, reference, statutory-em, strike, sub, sup, text

**See Also**

sup strike
Examples

Example 62. subsup.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<contract xmlns="urn:oasis:names:tc:eContracts:1:0"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:xi="http://www.w3.org/2001/XInclude">
  <title><text>Sample of sub and sup</text></title>
  <body>
    <block>
      <text>The Elkera<sup>®</sup> Business Narrative Markup Language (BNML<sup>â€”</sup>)</text>
    </block>
    <block>
      <text>A<sub>10</sub></text>
    </block>
  </body>
</contract>
```

subtitle

Synopsis

Content Model

subtitle ::= 
  • Zero or More of
    • conditional
    • citation
    • em
    • field
    • reference
    • statutory-em
    • strike
    • sub
    • sup
    • text
Attributes

Common Attributes

Description

This gives a subtitle for the element within it is contained.

Attributes

Common Attributes

Parents

subtitle appears inside: attachment contract

Children

The following elements occur inside subtitle: text

See Also:

title

Examples

Example 63. subtitle.XML

```xml
<?xml version="1.0" encoding="utf-8"?>
<contract xmlns="urn:oasis:names:tc:eContracts:1:0"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:xi="http://www.w3.org/2001/XInclude">
  <title><text>Subtitle Example</text></title>
  <subtitle>Here is a <em>subtitle</em> for this subcontract eContracts specification draft</subtitle>
  <body></body>
  <back></back>
  <attachments>
    <attachment><title><text>Attachment Title</text></title>
    <subtitle>Attachment Subtitle</subtitle>
    <subtitle>More than One Subtitle Allowed</subtitle>
  </attachments>
</contract>
```
**sup**

**Synopsis**

**Content Model**

sup ::=  
  • Zero or More of  
    • conditional  
    • em  
    • field  
    • reference  
    • statutory-em  
    • strike  
    • sub  
    • sup  
    • text

**Attributes**

Common Attributes and Class Attributes

**Description**

This is used to markup text that is to be superscripted.

**Attributes**

Common Attributes and Class Attributes

**Parents**

address, citation, conditional, date, date-block, em, fallback, name, note-in-line, num, phrase, reference, statutory-em, strike, sub, subtitle, sup, term, terms, and text xi:fallback

**Children**

The following elements occur inside: sup

em, field, reference, statutory-em, strike, sub, sup, text
See Also
sub strike

Examples

Example 64. subsup.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<contract xmlns="urn:oasis:names:tc:eContracts:1:0"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:xi="http://www.w3.org/2001/XInclude">
  <title><text>Sample of sub and sup</text></title>
  <body>
    <block>
      <text>The Elkera<sup>®</sup> Business Narrative Markup Language (BNML<sup>â€¢</sup>)</text>
    </block>
    <block>
      <text>A<sub>10</sub></text>
    </block>
  </body>
</contract>
```

table

Synopsis
table is taken from the CALS Exchange Table Model [table1], [table2], [table3] was also used in preparing this description of table and the elements that appear underneath it.

Content Model
table ::=  
  • Sequence of  
    • title  
    • One or more tgroup

Attributes
Common Attributes and orient.attribute

Additional Attributes
  • frame
Description

This is the opening element of a table. The information will be in the `tgroup`.

Processing Expectations

This allows the user to create a table and specify certain defaults that will apply to all elements.

Attributes

Common Attributes and `orient.attribute`

Additional Attributes:

<table>
<thead>
<tr>
<th>frame</th>
<th><code>Table 14. (enumeration)</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>top</td>
<td>below title</td>
</tr>
<tr>
<td>sides</td>
<td>(left and right sides)</td>
</tr>
<tr>
<td>bottom</td>
<td>after information for last row</td>
</tr>
<tr>
<td>topbot</td>
<td>both top and bottom</td>
</tr>
<tr>
<td>none</td>
<td>none of the above</td>
</tr>
<tr>
<td>all</td>
<td>all of the above</td>
</tr>
</tbody>
</table>

colsep If true, then display a line to the right of each cell (corresponding to an `entry`)

rowsep If true draw a roling below each row

pgwide If true, then the table spans the entire page.

Parents

table appears inside: `block`
Children

The following elements occur inside table: tgroup and title

See also:

In [Min1226], the TC noted the possibility of using markup for table for signatures. However, the TC thought that it would be more preferable to use markup specifically for signatures such as that provided by party-signature and signatory-group

Example

Example 65. table.XML

```xml
<?xml version="1.0" encoding="utf-8"?>
<contract xmlns="urn:oasis:names:tc:eContracts:1:0"
         xmlns:dc="http://purl.org/dc/elements/1.1/"
         xmlns:xi="http://www.w3.org/2001/XInclude">
  <title><text>Sample of
    - table
    - tgroup
  </text></title>
  <body>
    <block>
      <table>
        <tgroup cols="1">
          <colspec colnum="1" colname="C1" colwidth="10" colsep="true" rowsep="false" align="justify"/>
          <thead>
            <row><entry><block></block></entry></row>
          </thead>
          <tbody>
            <row><entry><block></block></entry></row>
          </tbody>
        </tgroup>
      </table>
    </block>
  </body>
</contract>
```

tbody

Synopsis

Content Model

tbody ::=  
  • One or more
• **row**

**Attributes**

Common Attributes

Additional Attributes

• **valign**

**Description**

Reminder: table is taken from the CALS Exchange Table Model [table1],[table2].

**Attributes**

Common Attributes

valign This is the default for all row and entry within the tbody element.

**Table 15. (enumeration)**

<table>
<thead>
<tr>
<th>Position</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>top</td>
<td></td>
</tr>
<tr>
<td>middle</td>
<td>approximately vertically centered</td>
</tr>
<tr>
<td>bottom</td>
<td></td>
</tr>
</tbody>
</table>

**Parents**

tbody appears inside: tgroup

**Children**

The following elements occur inside tbody: row

**Example:**

Please see table.

**term**

**Synopsis**

**Content Model**

term ::= 

• Zero or More of

  • conditional
  • em
• field
• reference
• statutory-em
• strike
• sub
• sup
• text

Attributes
Common Attributes and Class Attributes

Description
This contains a single term, as one can tell from the name, and is used in such places as definition, information relating to people such as party the text containers such as date-block and text

Attributes
Common Attributes and Class Attributes

Parents
date-block party person-record terms text

Children
The following elements occur inside: date
em, field, reference, statutory-em, strike, sub, sup, text

See Also
phrase and field

Examples

Example 66. body01.xml

<?xml version="1.0" encoding="utf-8"?>
<contract xmlns="urn:oasis:names:tc:eContracts:1:0"
xmlns:dc="http://purl.org/dc/elements/1.1/
xmlns:xi="http://www.w3.org/2001/XInclude">
<title><text>Sample of
- body
- item
In this licence:

**BNML Standard Schema**

means the XML Schema called

BNML Standard and all explanatory or reference documentation provided, or associated, with that Schema, described in the notice required by

Appendix A

Also, please see parties.XML under parties to see how this is used inside of a list of parties to indicate how each party would be referred to in a contract, e. g. John Doe will hereinafter be known as "General Partner."

**terms**

**Synopsis**

**Content Model**

**terms ::=**

- Zero or More of
  - conditional
  - em
  - field
• reference
• statutory-em
• strike
• sub
• sup
• text

Attributes

Common Attributes

Description

This is used in a definition when there are more than one terms associated with the same text, i.e., having the same definition.

It contains several term, possibly separated by some punctuation.

Attributes

Common Attributes

Parents

terms appears inside: definition

Children

The following elements occur inside terms:

em, field, reference, statutory-em, strike, sub, sup, text term

Examples

Example 67. terms.XML

```xml
<?xml version="1.0" encoding="utf-8"?>
<contract xmlns="urn:oasis:names:tc:eContracts:1:0"
         xmlns:dc="http://purl.org/dc/elements/1.1/"
         xmlns:xi="http://www.w3.org/2001/XInclude">
  <title><text>Sample of
         - terms and -term </text></title>
  <body>
    <block>
      <definition>
```

Specification
<p><a>text</a></p>

text - contains text and elements that markup parts of text, particularly for formatting purposes.

<h3>Synopsis</h3>

<h4>Content Model</h4>

text ::=  
  • Choice of:  
    • address  
    • conditional  
    • date  
    • em  
    • field  
    • note  
    • note-in-line  
    • object  
    • party  
    • person-record  
    • phrase  
    • reference  
    • statutory-em  
    • strike  
    • sub  
    • sup
Attributes

Common Attributes and Class Attributes

Additional attributes:

- **textflow** (enumeration)
  - "runon"

- **xml:space** (enumeration)
  - "default"
  - "preserve"

Description

As one can expect from the name, it contains the text or string of characters to be rendered in a document.

Attributes

Common Attributes and Class Attributes

Additional Attributes:

- **textflow** Tells whether this text element should be rendered in the same line as the previous element. Normally, if a text element occurs after a block, there will be a line break. If this attribute is runon, then it will be on the same line.

  **Table 16. (enumeration)**

<table>
<thead>
<tr>
<th>&quot;runon&quot;</th>
<th>If this text follows a block, the enclosed characters will appear on the same line as the text from the information from block.</th>
</tr>
</thead>
</table>

- **xml:space** If set to preserve, then white space (tabs, spaces or carriage returns) inside the element will be preserved.

  **Table 17. (enumeration)**

<table>
<thead>
<tr>
<th>&quot;default&quot;</th>
<th>Do not necessarily preserve white space</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;preserve&quot;</td>
<td>Preserve white space</td>
</tr>
</tbody>
</table>
Parents
text appears inside: address, block, citation, conditional, date, date-block, em, fallback, name, note-in-line, num, party, person-record, reference, sub, signature-line, statutory-em, sup, subtitle, strike, term, terms, text and title

Children
The following elements occur inside text: address, date, em, field, note, note-in-line, object, party, person-record, phrase, reference, statutory-em, strike, sub, sup, term, and text

See Also
term, note-in-line

Examples
See the BNML Schema Public License The two examples below show the text in its most typical home, a block element as well as some of the inline content such as em and sub that it often contains:

Example 68. subsup.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<contract xmlns="urn:oasis:names:tc:eContracts:1:0"
xmlns:dc="http://purl.org/dc/elements/1.1/
xmlns:xi="http://www.w3.org/2001/XInclude">
<title><text>Sample of sub and sup</text></title>
<body>
<block>
 <text>The Elkera<sup>®</sup> Business Narrative</text>
</block>
<block>
 <text>A<sub>10</sub></text>
</block>
</body>
</contract>
```

Example 69. em.XML

```xml
<?xml version="1.0" encoding="utf-8"?>
<contract xmlns="urn:oasis:names:tc:eContracts:1:0"
xmlns:dc="http://purl.org/dc/elements/1.1/
xmlns:xi="http://www.w3.org/2001/XInclude">
<title><text>Sample of inline elements</text></title>
<body>
```
**Synopsis**

In a table, a set of rows and possibly a header.

**Content Model**

tgroup ::=  
  • Sequence of  
    • Zero or more colspec  
    • Zero or one thead  
    • Exactly one tbody

**Attributes**

Common Attributes

Additional Attributes:

- cols
- colsep
- rowsep
- align

**Description**

Reminder: table is taken from the CALS Exchange Table Model [table1],[table2].

**Attributes**

Common Attributes

Additional Attributes:

align  If a cell within this tgroup does not have an enclosed formatting style, then align it as indicated:
### Table 18. (enumeration)

<table>
<thead>
<tr>
<th>left</th>
<th>align to the left (default)</th>
</tr>
</thead>
<tbody>
<tr>
<td>right</td>
<td>align to the right (default)</td>
</tr>
<tr>
<td>center</td>
<td>center the text</td>
</tr>
<tr>
<td>justify</td>
<td>justify the text</td>
</tr>
</tbody>
</table>

- **cols**: Number of columns in the `tgroup`
- **colsep**: If true, display a line to the right of each entry
- **rowsep**: If true, display a line below each cell.

#### Parents

- `tgroup` appears inside: `table`

#### Children

The following elements occur inside `tgroup`: `colspec`, `thead` and `tbody`

#### Example:

Please see `table`.

### thead

#### Synopsis

In a `table`, defines the header.

#### Content Model

thead ::= 

- One or more

  - **row**

#### Attributes

Common Attributes

Additional Attributes

- **valign**

#### Description

Reminder: `table` is taken from the CALS Exchange Table Model [table1],[table2].
Processing Expectations

These rows will be displayed at the beginning of the table, of course, and at new pages (or other physical specification)

Attributes

Common Attributes

valign This is the default for all row and entry within the thead element.

Table 19. (enumeration)

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>top</td>
<td></td>
</tr>
<tr>
<td>middle</td>
<td>approximately vertically centered</td>
</tr>
<tr>
<td>bottom</td>
<td></td>
</tr>
</tbody>
</table>

Parents

thead appears inside: tgroup

Children

The following elements occur inside thead: row

Example:

Please see table.

title

Synopsis

Content Model

title ::=  
  • Zero or more occurrences of  
    • text

Attributes

Common Attributes

Description

This is used to indicate the title for the contract itself, attachments, elements in lists, and many other elements, as listed below.
Parents

title appears inside: attachment background back body contract inclusion item block.item
parties table

Children

The following elements occur inside title: text

See Also:

subtitle

Examples

See party.xml under parties

witness

witness (a person who witnesses the signing of a contract or document)

Synopsis

Content Model

witness ::=  
  • Zero or more of  
    • signature-line  
    • block

Attributes

Common Attributes

Description

This provides the blank line and printed information for a witness who also signs a contract.

Attributes

Common Attributes

Parents

witness appears inside signatory-record

Children

The following elements appear inside witness: signature-line and block.
**See Also**

signatory

**Examples**

Please see party-signature

**xi:fallback**

**Synopsis**

It can only be used inside `xi:include` It will only be used when the look up the resource specified in the enclosing `xi:include href` attribute fails.

**Content Model**

`xi:fallback` ::= 

- Zero or More of
  
  - Conditional
  
  - `em`
  
  - `field`
  
  - `reference`
  
  - `statutory-em`
  
  - `strike`
  
  - `sub`
  
  - `sup`
  
  - `text`
  
  - `xi:include`

**Attributes**

None.

**Description**

As stated above, this provides the text or XML to be included when the `xi:include` that contains it fails.

This provides a recursive series starting with `xi:include`. That can contain a `xi:fallback` to specify what should be done if the look up for the top level fails. In turn, that `xi:fallback` may contain a `xi:include` And that `xi:include` may contain another `xi:fallback` should it fail and so on indefinitely.
Parents

`xi:fallback` appears inside `xi:include`

Children

The following elements occur inside `xi:fallback`:

`em, field, reference, statutory-em, strike, sub, sup, text` `xi:include`

See Also:

Please do not confuse this with `fallback` `xi:fallback` can only be used inside `xi:include`

Example

Please see the example for `xi:include`

`xi:include`

Synopsis

This allow including or bringing in content from other files. It uses the World Wide Web Consortium XML Inclusions recommendation:

http://www.w3.org/TR/2004/PR-xinclude-20040930

Content Model

`xi:include::=`

- Zero or one `xi:fallback`

Attributes

`href`

Description

As mentioned above, this is taken from from other files. It uses the World Wide Web Consortium XML Inclusions recommendation:

http://www.w3.org/TR/2004/PR-xinclude-20040930

It specifies an external reference from which to load the content.

Attributes

`href` This specifies the URI (possibly after escaping characters) from which to load the content.

Parents

`xi:include` appears inside `xi:fallback attachment back background body entry inclusion itemregular`
**Children**

The following elements occur inside `xi:include`, `xi:fallback`

**Example:**

This will bring in `Inventry01.xml`. If that is not available, it will bring in `Acct.xml`. If both of these files are not available, it will bring in the text:

This is the fallback of Acct.xml

It will also include `Terminal.xml`. Should this not be available, this will be a fatal error.

**Example 70. xiInclude.xml**

```xml
<?xml version="1.0" encoding="utf-8"?>
<contract xmlns="urn:oasis:names:tc:eContracts:1:0"
   xmlns:dc="http://purl.org/dc/elements/1.1/"
   xmlns:xi="http://www.w3.org/2001/XInclude">
  <title><text>Sample of xi:include.</text></title>
  <body>
    <xi:include href="Inventry01.xml">
      <xi:fallback>
        <xi:include href="Acct.xml">
          <xi:fallback>This is the fallback of Acct.xml.</xi:fallback>
        </xi:include>
      </xi:fallback>
    </xi:include>
    <xi:include href="Terminal.xml"/>
  </body>
</contract>
```

**Acknowledgments**

The following individuals have participated in the creation of this specification and are gratefully acknowledged:

- Rolly Chambers, Smith, Curry, and Hancock
- Daniel Greenwood, Electronic Commerce Architecture Project, Massachussets Institute of Technology
- Dr. Laurence Leff, Western Illinois University
- Mr. Dave Marvit, Fujitsu Corporation
- Mr. Peter Meyer, Elkera Pty Limited
- Dr. Zoran Milosovic, Deontic.com
References

Normative


[xinclude] *XML Inclusions (XInclude) Version 1.0* W3C Proposed Recommendation 30 September 2004


Non-normative


open.org/committees/documents.php?wg_abbrev=legalxml-econtracts

[MIN0119] Dr. Laurence Leff, Minutes Draft from the OASIS Legal XML Member Section Secretary (File id: MIN0119), Minutes Teleconference of January 19th, January 22 2006.

[MIN1226] Dr. Laurence Leff, Minutes Draft from the OASIS Legal XML Member Section Electronic Contracts Technical Committee Secretary (min1226) http://lists.oasis-open.org/archives/legalxml-econtracts

[MIN0216] Dr. Laurence Leff, Minutes Draft from the OASIS Legal XML Member Section Secretary (File id: MIN0216), Minutes of the February Sixteenth Teleconference, March Third, 2006. http://lists.oasis-open.org/archives/legalxml-econtracts


[meyer05a] Meyer, Peter, Development Steps for TC Specification Electronic Mail forwarded on September 21 2005 to the TC email list by civicsdotcom-oasis@yahoo.com http://lists.oasis-open.org/archives/legalxml-econtracts


[table1] CALS "XML Exchange Table Model" www.oasis-open.org/specs/soextblx.dtd

