INTERNATIONAL STANDARD

ISO/IEC
13250-3

First edition
2007-03-15

Information technology — Topic Maps —

Part 3:
XML syntax

Technologies de l'information — Plans relatifs à des sujets —

Partie 3: Syntaxe XML

© ISO/IEC 2007

PDF disclaimer

This PDF file may contain embedded typefaces. In accordance with Adobe's licensing policy, this file may be printed or viewed but shall not be edited unless the typefaces which are embedded are licensed to and installed on the computer performing the editing. In downloading this file, parties accept therein the responsibility of not infringing Adobe's licensing policy. The ISO Central Secretariat accepts no liability in this area.

Adobe is a trademark of Adobe Systems Incorporated.

Details of the software products used to create this PDF file can be found in the General Info relative to the file; the PDF-creation parameters were optimized for printing. Every care has been taken to ensure that the file is suitable for use by ISO member bodies. In the unlikely event that a problem relating to it is found, please inform the Central Secretariat at the address given below.

iTeh STANDARD PREVIEW
(standards.iteh.ai)

Contents

Page

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of the joint technical committee is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

ISO/IEC 13250-3 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 34, *Document description and processing languages.*

This first edition of ISO/IEC 13250-3 is part of a multi-part standard. The complete series will cancel and replace ISO/IEC 13250:2003.

ISO/IEC 13250 consists of the following parts, under the general title *Information technology — Topic Maps*:

— *Part 2: Data model*

— *Part 3: XML syntax*

— *Part 4: Canonicalization*

Introduction

XTM (XML Topic Maps) 2.0 is a syntax for the interchange of Topic Maps. The syntax is not designed to be extended or modified. Ease of human authoring was not prioritized during the design of XTM, and consequently it is not recommended to edit the syntax directly.

This part of ISO/IEC 13250 should be read in conjunction with *[ISO/IEC 13250-2]* since the interpretation of the XTM syntax is defined through a mapping from the syntax to the data model there defined. Informative guidance on how to serialize instances of the data model to the XTM syntax is also provided.

XTM 2.0 is a revision of the XTM 1.0 syntax defined in *[ISO/IEC 13250:2003]* and *[XTM1.0]*. A description of the differences between the two versions can be found in Annex D.

iTeh STANDARD PREVIEW
(standards.iteh.ai)

ISO/IEC 13250-3:2007
https://standards.iteh.ai/catalog/standards/sist/4c5c75f5-9614-460c-8b87-
23937b7dcf80/iso-iec-13250-3-2007

iTeh STANDARD PREVIEW

(standards.iteh.ai)

Information technology — Topic Maps —

Part 3:
XML syntax

1 Scope

This part of ISO/IEC 13250 defines an XML-based interchange syntax for Topic Maps, which can be used to interchange instances of the data model defined in *[ISO/IEC 13250-2]*. It also defines a mapping from the interchange syntax to the data model. The syntax is defined with a RELAX-NG schema, and more precision is provided through the mapping to the data model, which effectively also defines the interpretation of the syntax.

2 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

NOTE   Each of the following documents has a unique identifier that is used to cite the document in the text. The unique identifier consists of the part of the reference up to the first comma.

ISO/IEC 13250-2, *Information technology — Topic Maps — Data model*

W3C XML, *Extensible Markup Language (XML) 1.0 (Third Edition)*, W3C Recommendation, 4 February 2004, available at <http://www.w3.org/TR/2004/REC-xml-20040204>

W3C XML-Names, *Namespaces in XML*, W3C Recommendation, 14 January 1999, available at <http://www.w3.org/TR/1999/REC-xml-names-19990114/>

W3C XPointer, *XPointer Framework Version 1.0*, W3C Recommendation, 25 March 2003, available at <http://www.w3.org/TR/2003/REC-xptr-framework-20030325/>

W3C XML-Infoset, *XML Information Set (Second Edition)*, W3C Recommendation, 4 February 2004, available at <http://www.w3.org/TR/2004/REC-xml-infoset-20040204>

W3C Canonical XML, *Canonical XML Version 1.0*, W3C Recommendation, 15 March 2001, available at <http://www.w3.org/TR/2001/REC-xml-c14n-20010315>

ISO/IEC 19757-2, *Information technology — Document Schema Definition Languages (DSDL) — Part 2: Regular-grammar-based validation — RELAX NG*

IETF RFC 3986, *Uniform Resource Identifier (URI): Generic Syntax*, Internet Standards Track Specification, January 2005, available at <http://www.ietf.org/rfc/rfc3986.txt>

IETF RFC 3987, *Internationalized Resource Identifiers (IRIs)*, Internet Standards Track Specification, January 2005, available at <http://www.ietf.org/rfc/rfc3987.txt>

3 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

3.1
XTM
the syntax defined in this part of ISO/IEC13250

4   Syntax definition

4.1   About the syntax

The acronym *XTM* is often used to refer to the syntax defined in this part of ISO/IEC13250. Its full name is XML Topic Maps. The namespace for the XTM syntax is `http://www.topicmaps.org/xtm/`.

An XTM document is an XML document that conforms to the XTM syntax. This clause defines the syntax of XTM documents using a RELAX-NG schema in compact syntax *[ISO/IEC 19757-2]*, and their semantics using prose describing the mapping from XTM documents to *[ISO/IEC 13250-2]*. The full schema can be found in Annex A, a DTD in Annex B, and a W3C XML Schema in Annex C.

4.2   Deserialization

The process of exporting Topic Maps from an implementation's internal representation of the data model to an instance of a Topic Maps syntax is known as serialization. The opposite process, deserialization, is the process of building an instance of an implementation's internal representation of the data model from an instance of a Topic Maps syntax.

This clause defines how instances of the XTM syntax are deserialized into instances of the data model defined in *[ISO/IEC 13250-2]*. Serialization is only implicitly defined, but implementations should guarantee that for any data model instance the XTM serialization produced by the implementation should when deserialized to a new data model instance produce one that has the same canonicalization as the original data model instance, according to *[ISO 13250-4]*.

The input to the deserialization process is:

— A document item as defined by *[W3C XML-Infoset]*, representing an XTM document. (Information item properties from *[W3C XML-Infoset]* are referred to using [[property name]], in order to distinguish them from properties from *[ISO/IEC 13250-2]*.)

— An absolute IRI. This is the IRI from which the XTM document was retrieved, known as the document IRI. This IRI shall always be provided, as it is necessary in order to assign the item identifiers of the topic items created during deserialization. If the XTM document was not read from any particular IRI the application is responsible for providing an IRI considered suitable.

Deserialization is done by processing each element item in the document item in document order. For each element item encountered the operations specified in the clause for that element type are performed. An input element item matches a clause in this document when the [[namespace uri]] property is set to `"http://www.topicmaps.org/xtm/"`, and the [[local name]] matches the element type name given in the title of that clause.

Whenever a new information item is created, those of its properties which have set values are initialized to the empty set; all other properties are initialized to null.

NOTE   This part of ISO/IEC13250 requires an instance of the XML Information Set as input to the deserialization process, but in most cases the actual input will be an XML document. This part of ISO/IEC13250 does not constrain how XML Information Set instances are built from XML documents, but assumes that in most cases this will be done by simply using an XML processor.

XML processors conformant to the XML Recommendation may produce different results given the same XML document, depending on whether they are validating or non-validating, and depending on which optional features they support. Reliance on any particular behaviour in the XML processors used by recipients is strongly discouraged.

4.3   Common syntactical constructs

4.3.1   Common declarations

The following declarations are used throughout the schema for brevity.

```
default namespace = "http://www.topicmaps.org/xtm/"
```

```
namespace xtm = "http://www.topicmaps.org/xtm/"

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

start = topicMap

reifiable = attribute reifier { xsd:anyURI }?, itemIdentity*
href = attribute href { xsd:anyURI }

any-markup = (text | element * - xtm:* { attribute * { text }*, any-markup* })*
```

4.3.2 The reifier attribute

The reifier attribute is used to refer from the topic map construct on which it appears to the topic reifying that construct. The reference is an IRI matching one of the topic's item identifiers.

During deserialization the value in the [[normalized value]] property of the attribute node representing the reifier attribute is resolved into an absolute IRI following the procedure in 4.3.4. If there is a topic item with that IRI in its [item identifiers] property, that topic item is set as the value of the [reifier] property of the topic map construct being processed. If no such topic item exists, a new topic item is created, the IRI added to its [item identifiers] property, and that topic item is set as the value of the [reifier] property of the topic map construct being processed.

4.3.3 The href attribute

The href attribute always references an information resource using a relative or absolute IRI valid according to *[IETF RFC 3986]* and *[IETF RFC 3987]*, but the meaning of the reference depends on context.

During deserialization the value in the [[normalized value]] property of the attribute node representing the href attribute is turned into an IRI following the procedure in 4.3.4.

4.3.4 Creating IRIs from strings

To create an IRI from a string unescape the string by replacing %HH escape sequences with the characters they represent, and decode the resulting character sequence from UTF-8 to a sequence of abstract Unicode characters. The resulting string is turned into an absolute IRI by resolving it against the document IRI.

4.4 The `topicMap` element

The `topicMap` element type is the document element of all XTM documents. It acts as a container for the topic map, and can be used to reify it, but has no further significance. It is declared as follows:

```
topicMap = element topicMap { reifiable, version, mergeMap*,
                   (topic | association)* }

version = attribute version { "2.0" }
```

The version attribute is used to specify which version of XTM the document conforms to. For XTM 2.0 documents this shall be "2.0".

During deserialization the `topicMap` element causes a topic map item to be created.

4.5 The `topic` element

The `topic` element type is used to represent topics, and acts as a container and point of reference for topic information. The child elements of the `topic` element provide identification as well as names and occurrences, while association roles played by the topic are specified outside the `topic` element.

The `topic` element type is declared as follows:

```
topic = element topic { id,
                        (itemIdentity | subjectLocator | subjectIdentifier)*,
                        instanceOf?, (name | occurrence)* }
id = attribute id { xsd:ID }
```

The id attribute provides a unique identifier within the document for the topic, which is used to refer to it.

During deserialization the `topic` element causes a topic item to be created and inserted into the [topics] property of the topic map item.

A locator is created by concatenating the document IRI, a `"#"` character, and the value of the [[normalized value]] property of the attribute item in the [[attributes]] property of that element item whose [[local name]] property is `"id"`. This locator is added to the [item identifiers] property of the topic item. If this makes the topic item equal to another topic item the two topic items are merged according to the procedure given in *[ISO/IEC 13250-2]*.

4.6 The `itemIdentity` element

The `itemIdentity` element is used to assign an item identifier to the topic map construct represented by its parent element. It is declared as follows:

```
itemIdentity = element itemIdentity { href }
```

During deserialization the `itemIdentity` element causes a locator to be created from its href attribute as specified in 4.3.3. This locator is added to the [item identifiers] property of the information item created by the parent element. If the parent element is a `topic` element and this makes the topic item equal to another topic item the two topics are merged according to the procedure given in *[ISO/IEC 13250-2]*.

4.7 The `subjectLocator` element

The `subjectLocator` element is used to assign a subject locator to the topic that is represented by its parent `topic` element. It is declared as follows:

```
subjectLocator = element subjectLocator { href }
```

During deserialization the `subjectLocator` element causes a locator to be created from its href attribute as specified in 4.3.3 and added to the [subject locators] property of the topic item created by the parent `topic` element. If this makes the topic item equal to another topic item the two topic items are merged according to the procedure given in *[ISO/IEC 13250-2]*.

4.8 The `subjectIdentifier` element

The `subjectIdentifier` element is used to assign a subject identifier to the topic that is represented by its parent `topic` element.

The `subjectIdentifier` element is declared as follows:

```
subjectIdentifier = element subjectIdentifier { href }
```

During deserialization the `subjectIdentifier` element causes a locator to be created from its href attribute as specified in 4.3.3. This locator is added to the [subject identifiers] property of the topic item created by the parent `topic` element. If this makes the topic item equal to another topic item the two topic items are merged according to the procedure given in *[ISO/IEC 13250-2]*.

4.9 The `instanceOf` element

The `instanceOf` element type is used to assign one or more types to the topic represented by its parent element. The types are always topics, indicated by the `instanceOf` element's child elements. The `instanceOf` element type is declared as follows:

```
instanceOf = element instanceOf { topicRef+ }
```

For each child element of the `instanceOf` element a topic item is produced following the procedure in 4.20. For each topic item the following steps are then taken:

— A new association item is created, with two association role items in its [roles] property, and a topic item representing the type-instance association type (described in *[ISO/IEC 13250-2:2006]*, 7.2) in its [type] property. If no such topic item exists already, one is created, and the subject identifier added to its [subjectidentifiers] property.

— The first association role item has its [type] property set to the topic item representing the `type` role in the same association (see the section referenced above), while the [player] property is set to the topic produced by the child `topicRef` element.

— The second association role item has its [type] property set to the topic item representing the `instance` role in the same association (see the section referenced above), while the [player] property is set to the topic produced by the parent element (that is, the current topic).

4.10 The `name` element

The `name` element type is used to add topic names to the topic represented by the parent `topic` element. The child elements of the `name` element provide the property values of the topic name item.

The `name` element type is declared as follows:

```
name = element name { reifiable, type?, scope?, value, variant* }
```

During deserialization the `name` element causes a topic name item to be created, and added to the [topic names] property of the topic item created by the parent `topic` element.

If the `name` element has an `type` child element it is processed according to the procedure in 4.14. Otherwise the [type] property of the topic name item is set to the topic item whose [subject identifiers] property contains "`http://psi.topicmaps.org/iso13250/model/topic-name`"; if no such topic item exists, one is created.

4.11 The `value` element

The `value` element type is used to provide the value of the topic name. It is declared as follows:

```
value = element value { text }
```

During deserialization the information items in the [[children]] property of the `value` element are traversed, and for each character information item the Unicode character specified by the [[character code]] property is appended to the [value] property of the topic name item created by the parent `name` element.

4.12 The `variant` element

The `variant` element type is used to add a variant name to a topic name. It is declared as follows:

```
variant = element variant { reifiable, scope, (resourceRef | resourceData) }
```

During deserialization the `variant` element causes a variant item to be created and added to the [variants] property of the topic name item created by the `name` parent element. After the `scope` child element has been processed, the topics in the [scope] property of the topic name item created by the `name` parent element are added to the [scope] property of the variant name item.

4.13 The `scope` element

The `scope` element type is used to assign a scope to the statement represented by the parent element. It is declared as follows:

```
scope = element scope { topicRef+ }
```

During deserialization the `scope` element has no direct effect on the information set being produced, but changes the interpretation of its child elements. Each `topicRef` child element is processed according to the procedure in 4.20 to produce a topic item. These topic items are gathered into a set that is assigned as the value of the [scope] property of the information item produced by the parent element.