XML Information Set

Working Draft of May 17, 1999

This version:

http://www.w3.org/TR/1999/WD-xml-infoset-19990517

Latest version:

http://www.w3.org/TR/xml-infoset

Editors:

John Cowan

David Megginson

Abstract

This specification describes an abstract data set containing the information available from an XML document.

Status of this Document

This is a W3C Working Draft for review by members of the W3C and other interested parties in the general public. Because it is the first public release, it contains many queries and open issues, all of which are clearly indicated in the document.

While it is a Working Draft or a Proposed Recommendation it is subject to change. It may be updated, replaced or rendered obsolete by other W3C documents at any time. It is inappropriate to use W3C Working Drafts as reference material or to cite them as other than "work in progress."

This work is part of the W3C XML Activity.

Please review and send comments to `www-xml-infoset-comments@w3.org`, which is a publicly-archived mailing list.

Contents

1. Introduction

This document specifies an abstract data set called the **XML information set** (**Infoset**), a description of the information available in a well-formed XML document [XML].

An XML document's information set consists of two or more **information items** (the information set for any well-formed XML document will contain at least the document information item and one element information item). An information item is an abstract representation of some component of an XML document: each information item has a set of associated properties, some of which are required to be available through the information set, and some of which are optionally available.

The XML information set does not require or favor a specific interface or class of interfaces. This specification presents the information set as a tree for the sake of clarity and simplicity, but there is no requirement that the XML information set be made available through a tree structure; other types of interfaces, including (but not limited to) event-based and query-based interfaces are also capable of providing information conforming to the information set. As long as the information in the information set is made available to XML applications in one way or another, the requirements of this document are satisfied.

> ***Note:*** *In this document, the words "must", "should", and "may" assume the meanings specified in RFC 2119 [RFC2119], except that the words do not appear in upper case.*

> ***Note:*** *To the best of the editors' knowledge and belief, the information set scheme described in this document satisfies the requirements of the XPointer-Information Set Liaison Statement [XPointer-Liason].*

> ***Note:*** *To the best of the editors' knowledge and belief, the interface specified by the Document Object Model, level one core Recommendation [DOM] conforms to the XML Information Set as currently specified.*

2. Information Items

The XML information set can contain eleven different types of information items (in the following list, read "required" as "required if present in the original XML document"; see also Processor Limitations, below):

1. a document information item (*required*)
2. element information items (*required*)
3. attribute information items (*required*)
4. processing instruction information items (*required*)
5. character information items (*required*)
6. reference to unknown entity information items (*required*)
7. comment information items (*optional*)
8. a document type declaration information item (*optional*)
9. entity information items (*required for unparsed entities, optional for others*)
10. notation information items (*required*)
11. attribute declaration information items (*optional*)

2.1. The Document Information Item

> ***XML Definition:*** *document (Section 2, Documents)*

> ***XML Syntax:*** *[1] Document (Section 2.1, Well-Formed XML Documents)*

There is always one **document information item** in the information set, and all other information items are related to the document information item, either directly or indirectly.

2.1.1. Document: Required Properties

The document information item must have the following properties available in some form:

> ***Query:*** *Should comments and the document type declaration be required rather than optional?*

1. An ordered list of child information items, in the original document order. The list must contain exactly one element information item, together with one processing instruction information item for each processing instruction preceding the document element (either

in the document entity or in a lower-level entity) or following the document entity; the list may optionally contain other information items as well (see below).

2. An unordered set of notation information items, one for each notation declaration that the XML processor has read.
3. An unordered set of entity information items, one for each unparsed entity (NDATA) declaration that the XML processor has read.

2.1.2. Document: Optional Properties

The document information item may also have the following properties available in some form:

4. One comment information item for each comment outside the document element, added to the ordered list of child information items. The relative position of each comment information item in the list must reflect its position in the original document. Comments within the internal or external DTD subset are considered to follow the document type declaration information item and to precede the document element.
5. Exactly one document type declaration information item, added to the ordered list of child information items. The relative position of the document type declaration information item in the list must reflect its position in the original document.
6. One entity information item for each parsed entity declaration read by the processor, added to the unordered set of entities. There can also be an entity information item for the document entity and for the external DTD subset.
7. An unordered set of attribute declaration information items, one for each attribute declaration read by the processor.

2.2. Element Information Items

> ***XML Definition:*** *element (Section 3, Logical Structures)*
>
> ***XML Syntax:*** *[39] Element (Section 3, Logical Structures)*

There is one **element information item** for each element appearing in the XML document. Exactly one of the element information items correspond to the document element (the root of the element tree), and all other element information items are contained within the document element, either directly or indirectly.

2.2.1. Elements: Required Properties

An element information item must have the following properties available in some form:

> ***Query:*** *Should comments be required rather than optional?*
>
> ***Query:*** *When Namespace processing is being performed, should the original prefix also be available?*

***Query:** Should attribute starting with "xmlns" be included even when performing Namespace processing?*

1. The URI part, if any, of the element's name. If Namespace processing is not being performed, the URI part will always be null.
2. The local part of the element's name. If Namespace processing is being performed, the prefix and colon (if any) will have been removed from the beginning of the name; if Namespace processing is not being performed, the local part will contain the entire name from the original document, including any colons.
3. An ordered list of element, processing instruction, reference to unknown entity and character information items, one for each element, processing instruction, reference to an unknown entity, and character appearing immediately within the current element, in the original document order. If the element is empty, this list will have zero members.
4. An unordered set of attribute information items, one for each of the attributes (specified or defaulted) for this element. If Namespace processing is being performed, attributes with names beginning with `xmlns` will be excluded from the set; if Namespace processing is not being performed, attribute with names beginning with `xmlns` are included in the set. If there are no non-#IMPLIED attributes specified or defaulted for the element, this set will be empty.

2.2.2. Elements: Optional Properties

An element information item may also have the following properties available in some form:

5. One comment information item for each comment appearing immediately within the current element, added to the ordered list of elements, processing instructions, references to unknown entities, and characters appearing immediately within the current element. The relative position of each comment information item in the list must reflect its position in the original document.
6. A reference to the entity information item for the entity in which this element begins and ends.

2.3. Attribute Information Items

***XML Definition:** attribute (Section 3.1, Start-Tags, End-Tags, and Empty-Element Tags)*

***XML Syntax:** [41] Attribute (Section 3.1, Start-Tags, End-Tags, and Empty-Element Tags)*

There is one **attribute information item** for each attribute (specified or defaulted) for each element in the document instance; when Namespace processing is being performed, attributes with names beginning with "xmlns" will not have corresponding information items.

> ***Query:*** *Should xml:lang and xml:space also be excluded and modeled as character properties instead?*

Attributes declared in the DTD with a default value of `#IMPLIED` and not specified in the element's start tag are not represented by attribute information items.

2.3.1. Attributes: Required Properties

An attribute information item must have the following properties available in some form:

> ***Query:*** *When Namespace processing is being performed, should the original prefix also be available?*

1. The URI part, if any, of the attribute's name. If Namespace processing is not being performed, the URI part will always be null.
2. The local part of the attribute's name. If Namespace processing is being performed, the prefix and colon (if any) will have been removed from the beginning of the name; if Namespace processing is not being performed, the local part will contain the entire name including any colons.
3. An ordered list of character information items, one for each character appearing in the normalized attribute value.

2.3.2. Attributes: Optional Properties

In addition, for each attribute information item, the following property may **optionally** be available in some form:

4. A reference to the attribute declaration information item corresponding to this attribute.

2.4. Processing Instruction Information Items

> ***XML Definition:*** *processing instruction (Section 2.6, Processing Instructions)*

> ***XML Syntax:*** *[16] PI (Section 2.6, Processing Instructions)*

There is one **processing instruction information item** for every processing instruction in the document. The XML declaration and text declarations for external parsed entities are not considered processing instructions.

2.4.1. Processing Instructions: Required Properties

A processing instruction information item must have the following properties available in some form:

1. The target part of the processing instruction's content (an XML name).
2. The content of the processing instruction, excluding the target and any whitespace immediately following it. The content may be the empty string.

2.4.2. Processing Instructions: Optional Properties

A processing instruction information item may also have the following properties available in some form:

3. A reference to the entity information item for the entity in which this processing instruction appears.

2.5. Reference to Unknown Entity Information Items

> ***XML Definition:*** *Section 4.4.3, Included If Validating*

There is one **reference to unknown entity information item** for each reference to an entity not included by a non-validating XML processor, either because the processor has not read the declaration or because the processor does not include external parsed entities.

A validating XML processor will never generate reference to unknown entity information items for a valid XML document.

2.5.1. Reference to Unknown Entity: Required Properties

A reference to unknown entity information item must have the following information available in some form:

1. The name of the entity referenced.

2.5.2. Reference to Unknown Entity: Optional Properties

A reference to unknown entity information item may also have the following properties available in some form:

2. A reference to the entity information item for an external parsed entity, if the processor has read the declaration.
3. A reference to the entity information item for the entity in which this reference appears.

2.6. Character Information Items

> ***XML Definition:*** *characters (Section 2.2, Characters)*
>
> ***XML Syntax:*** *[2] Char (Section 2.2, Characters)*

There is one **character information item** for each non-markup character that appears within the document element, either literally, as a character reference, or within a CDATA section. There is also one character information item for each character that appears in a normalized attribute value.

Note, however, that a CR (#xD) character that is followed by a LF (#xA) character is not represented by any information item. Furthermore, a CR character that is *not* followed by a LF character is treated as a LF character. This rule does not apply to CR characters created by character references such as `&#xD;` or `&#13;`.

Each character is a logically-separate information item, but processing software is free to chunk characters into larger groups as necessary.

2.6.1. Characters: Required Properties

A character information item must have the following properties available in some form:

1. The ISO 10646 character code (in the range 0 to hex 0010FFFF) of the character.
2. A flag indicating whether the character is whitespace appearing within element content (see [XML], 2.10 "White Space Handling"). Validating processors are required by XML 1.0 to provide this information; non-validating processors may always set this flag to false.

2.6.2. Characters: Optional Properties

A character information item may also have the following properties available in some form:

> ***Query:*** *Should the inherited values of xml:lang and xml:space also be modeled as optional character properties?*

3. An indication of whether the character was included literally, as a character reference, as part of a CDATA section, or through one of the predefined XML entities.
4. A reference to the entity information item for the entity in which this character appears.

2.7. Comment Information Items

> ***XML Definition:*** *comment (Section 2.5, Comments)*
>
> ***XML Syntax:*** *[15] Comment (Section 2.5, Comments)*
>
> ***Query:*** *Should comment information items be required?*

The optional **comment information item** corresponds to a single XML comment in the original document.

2.7.1. Comments: Required Properties

> ***Query:*** *Should the contents of the comment be optional, so that only its position may be reported?*

If a comment information item is included, the following properties must be available:

1. The content of the comment.

2.7.2. Comments: Optional Properties

A comment information item may also have the following properties available in some form:

1. A reference to the entity information item for the entity in which this comment appears.

2.8. The Document Type Declaration Information Item

> ***XML Definition:*** *document type declaration (section 2.8, Prolog and Document Type Declaration)*
>
> ***XML Syntax:*** *[28] doctypedecl (section 2.8, Prolog and Document Type Declaration)*

If the XML document has a document type declaration, then the information set may optionally contain a single **document type declaration information item**.

2.8.1. Document Type Declaration: Optional Properties

A document type declaration information item may have the following properties available in some form:

1. A reference to the entity information item for the external DTD subset. The public and system identifiers for the external DTD subset are available through this information item.

2.9. Entity Information Items

> ***XML Definition:*** *entity (section 4, Physical Structures)*
>
> ***XML Syntax:*** *[70] EntityDecl (section 4.2, Entity Declarations)*

Entity information items are optional, except for information items representing unparsed external (NDATA) entities, which are required to appear in the information set.

There is at most one entity information item for each entity, internal or external, declared in the DTD: when the same entity is declared more than once, only the first declaration is used. There is also at most one entity information item for the document instance, and at most one for the DTD external subset (if there is one).

> ***Query:*** *Is it confusing to represent the external DTD subset with an entity information item? (The XML Recommendation treats the external subset essentially as an external parameter entity, except that it does not have an entity name.)*

2.9.1. Entities: Required Properties

The entity information item, if included, must have the following information available in some form:

1. An indication of the type of the entity (internal parameter entity, external parameter entity, internal general entity, external general entity, unparsed entity, document entity, or external DTD subset).
2. The name of the entity. If the information item represents the document entity or the external DTD subset, the name is null.
3. The system identifier of the entity. If the information item represents an internal entity, the system identifier is always null, and if it represents the document entity, the value *may* be null; otherwise, it must have a non-null value.
4. The public identifier of the entity, if one is available. For internal entities, the value is always null.
5. A reference to the notation information item associated with the entity, if the entity is an unparsed (NDATA) entity. For entities other than unparsed entities, the value is always null.

2.9.2. Entities: Optional Properties

An entity information item may also have the following information available in some form:

> ***Query:*** *Should the information from the XML declaration or text declaration also be optionally available?*

6. The text of the entity, if it is an internal entity.
7. A reference to the entity information item for the entity in which the entity was declared.

2.10. Notation Information Items

> ***XML Definition:*** *notation (section 4.7, Notation Declarations)*
>
> ***XML Syntax:*** *[82] NotationDecl (section 4.7, Notation Declarations)*

There is one **notation information item** for each notation declared in the DTD.

2.10.1. Notations: Required Properties

A notation information item must have the following properties available:

1. The name of the notation.
2. The system identifier of the notation, if one was specified.
3. The public identifier of the notation, if one was specified.

2.10.2. Notations: Optional Properties

4. A reference to the entity information item for the entity in which the notation was declared.

2.11. Attribute Declaration Information Items

> ***XML Definition:*** *attribute declaration (section 3.3, Attribute-List Declarations)*
>
> ***XML Syntax:*** *[53] AttDef (section 3.3, Attribute-List Declarations)*

Attribute declaration information items are an optional part of the information set. There is at most one attribute declaration information item for each attribute declared in an ATTLIST declaration within the DTD: if an attribute is declared more than once for the same element, only the first declaration is used.

2.11.1. Attribute Declarations: Required Properties

An attribute declaration information item, if present, must have the following properties available in some form:

1. The name of the attribute as declared, including any Namespace prefix.
2. The name of the element type for which the attribute was declared.
3. The default value of the attribute. If the attribute was declared with the default value #IMPLIED or #REQUIRED, this value will be null.

2.11.2. Attribute Declarations: Optional Properties

If an attribute declaration information item is provided for an XML document, the following properties may be available in some form:

> ***Query:*** *Should any of this information be required?*

4. The XML type of the attribute (ID, IDREF, IDREFS, NMTOKEN, NMTOKENS, ENTITY, ENTITIES, NOTATION, or ENUMERATION).
5. A list of allowed values for the attribute, if it has the type NOTATION or ENUMERATION.
6. The default declaration type of the attribute (REQUIRED, IMPLIED, FIXED, or DEFAULTED).
7. A reference to the entity information item for the entity in which the attribute was declared.

3. Namespace Processing

Namespace processing [Namespaces] represents a virtual transformation of an XML document, where elements and attributes acquire new, two-part names based on declarations made with specially-named attributes. As a result, for any single XML document there are two possible instantiations of the XML Information Set: one without Namespace processing, and one with.

> ***Query:*** *Is it best for the Information Set to explicitly allow for a document without Namespace processing?*

The XML Information Set provides a single model that is capable of describing a document either without or with Namespace processing, **at user option**: element and attribute names have both URI parts and local parts, and the URI parts will simply be null when Namespace processing is not in force.

Consider the following example:

```
<?xml version="1.0"?>

<msg:message dc:date="19990421"
             xmlns:dc="http://purl.org/metadata/dublin_core#"
             xmlns:msg="http://www.message.net/"
>Phone home!</msg:message>
```

Without Namespace processing, the Information Set for this document will contain the following items in some form (for simplicity's sake, some properties have been omitted):

- A Document information item.
- An Element information item with the URI part null and the local part "`msg:message`".
- An Attribute information item with the URI part null and the local part "`dc:date`".
- An Attribute information item with the URI part null and the local part "`xmlns:dc`".
- An Attribute information item with the URI part null and the local part "`xmlns:msg`".
- Eleven Character information items for the character data, and an additional 68 Character information items for the attribute values.

With Namespace processing, the Information Set for the *same* XML document will contain the following items in some form:

- A Document information item.
- An Element information item with the URI part "`http://www.message.net/`" and the local part "`message`".
- An Attribute information item with the URI part "http://purl.org/metadata/dublin_core#" and the local part "`date`".
- Eleven Character information items for the character data, and an additional 8 Character information items for the attribute value.

If an XML document contains no names that include colons and no attribute names that begin with the letters "xmlns", then the XML Information Set will be instantiated identically with or without Namespace processing.

4. Conformance

An XML processor conforms to the XML Information Set if it provides all the required information items and all required associated information. For instance, attributes are required information items, and an XML processor that does not report the existence of attributes, as well as their names (and URI parts if Namespace processing is being performed) and values, does not conform to the XML Information Set.

Some information items are optional, and some required information items have optional information associated with them. If a processor is required to or chooses to report an information item, then it is required to supply at least what the XML Information Set defines for that item in order to conform. For instance, if a processor chooses to supply entity information items, which are optional, then it is required to supply names for the entities, since the XML Information Set specifies that entity information items are required to make knowledge available about entity names. However, since entity information items are optional, a processor which does not supply them at all also conforms to the XML Information Set.

XML Processors may optionally provide additional information not found in the XML Information Set; for instance, the XML Information Set excludes whitespace that occurs between attributes from the information set, but an XML Processor that provides this information conforms as long as it provides the information that is required by the XML Information Set.

5. Processor Limitations

The information set for an XML document can contain only information that a processor has actually read.

The XML 1.0 Recommendation [XML] explicitly allows non-validating XML processors to omit parsing the external DTD subset and external entities (both parsed general entities and parameter entities). As a result, it is possible that a non-validating processor will omit reading attribute and entity declarations or actual markup that will affect the quantity and quality of information included in the information set.

Wherever this specification designates information as *required*, it is important to note that the information is required only if the processor actually reads the part of the XML document in which the information appears. Validating processors must report all required information; non-validating processors may omit information that appears outside of the top-level document entity (either in the external DTD subset or in an external text entity) if they do not read the other entities.

6. XML 1.0 Reporting Requirements

Although the XML 1.0 Recommendation [XML] is primarily concerned with XML syntax, it also includes some specific reporting requirements for processors.

The reporting requirements include errors, which are outside the scope of this specification, and document information; all of the XML 1.0 requirements for document information reporting have been integrated into the XML information set specification (numbers in parentheses refer to sections of the Recommendation):

1. An XML processor must always provide all characters in a document that are not part of markup to the application (2.10). We have interpreted this requirement to refer only to characters within the document element.
2. A validating XML processor must inform the application which of the character data in a document is whitespace appearing within element content (2.10).
3. An XML processor must pass a single LF character in place of CR or CR-LF characters appearing in its input.
4. An XML processor must normalize the value of attributes according to the rules in clause 3.3 before passing them to the application. This implies that the value of attributes after normalization are passed to the application (3.3).
5. An XML processor must pass the names and external identifiers (system identifiers, public identifiers or both) of declared notations to the application (4.7).
6. When the name of an unparsed entity appears as the explicit or default value of an ENTITY or ENTITIES attribute, an XML processor must provide the names, system identifiers, and (if present) public identifiers of both the entity and its notation to the application (4.6, 4.7).
7. An XML processor must pass processing instructions to the application. (2.6)
8. An XML processor (necessarily a non-validating one) that does not include the replacement text of an external parsed entity in place of an entity reference must notify the application that it recognized but did not read the entity (4.4.3).
9. A validating XML processor must include the replacement text of an entity in place of an entity reference. (5.2)
10. A validating XML processor must supply the default value of attributes declared in the DTD for a given element type but not appearing in the element's start tag (5.2).

7. What is not in the Information Set

The following information is not represented in the current version of the XML Information Set:

> ***Query:*** *Should any of this information be included?*

1. The information in the XML declaration and text declarations.
2. Element content models from ELEMENT declarations.
3. The grouping and ordering of attribute declarations in ATTLIST declarations.
4. Whitespace outside the document element.
5. Whitespace within start-tags (other than significant whitespace in attribute values) and end-tags.
6. The difference between CR, CR-LF, and LF line termination.
7. The unnormalized form of attribute values (see 3.3.3 Attribute-Value Normalization [XML]).
8. The order of attributes within a start-tag.
9. The order of declarations within the DTD.
10. The boundaries of conditional sections.
11. Any ignored declarations, including those within an IGNORE conditional section, as well as entity and attribute declarations ignored because previous entity declarations overrode them.

Furthermore, the XML Infoset does not provide any method of assigning a single series of numbers to all child nodes of an element or of the document that is guaranteed to be reliable regardless of the underlying XML processor. Although such a method would be desirable, it is considered unachievable, due to the difficulties produced by references to unknown entities and optional information items.

In other words, there is no reliable way to specify something like "the second child of this element" without restricting both the type of processor and the types of children being counted. For more information, see the section on processor limitations.

8. Other Open Issues

1. [3a, 3c] Links are not currently represented in the information set. Should they be? In particular, should ID-IDREF connections be represented?
2. [3b] Should there be a specification of partial information sets?
3. [7] How do we interoperate with DOM?
4. [22] Should the Infoset model different degrees of ordering for different application domains?
5. [18] Are there remaining problems with hierarchically-scoped information?
6. Should this document specify what information (if any) a validating processor must make available for a well-formed but invalid document?

9. References

DOM

Document Object Model (DOM) Level 1 Specification, eds. Vidur Apparao, Steve Byrne, Mike Champion, et alii. 1 October 1998. Available at http://www.w3.org/TR/REC-DOM-Level-1/.

Namespaces

Namespaces in XML, eds. Tim Bray, Dave Hollander, and Andrew Layman. 14 January 1999. Available at http://www.w3.org/TR/REC-xml-names.

RFC2119

Key words for use in RFCs to Indicate Requirement Levels, ed. S. Bradner. March 1997. Available at http://www.isi.edu/in-notes/rfc2119.txt.

XML

Extensible Markup Language (XML) 1.0, eds. Tim Bray, Jean Paoli, and Michael Sperberg-McQueen. 10 February 1998. Available at http://www.w3.org/TR/REC-xml.

XPointer-Liason

XPointer-Information Set Liason Statement, ed. Steven J. DeRose. 24 February 1999. Available at http://www.w3.org/TR/NOTE-xptr-infoset-liaison.