XML Information Set

W3C Working Draft 2 February 2001

This version:
http://www.w3.org/TR/2001/WD-xml-infoset-20010202
Latest version:
http://www.w3.org/TR/xml-infoset
Previous version:
http://www.w3.org/TR/2000/WD-xml-infoset-20001220
Editors:
John Cowan
Richard Tobin

Abstract

This specification provides a set of definitions for use in other specifications that need to refer to the information in an XML document.

Status of this Document

Though this specification has already had a Last Call review on an earlier version, in light of the review and much discussion, the XML Core Working Group has reworked the specification. The WG invites public comment on this new Last Call draft. The Last Call period ends on 23 February 2001.

Comments on this specification are invited and should be sent to the public mailing list www-xml-infoset-comments@w3.org. An archive is available at http://lists.w3.org/Archives/Public/www-xml-infoset-comments/.

For background on this work, please see the XML Activity Statement. This specification is a product of the XML Core Working Group.

See the XML Information Set Requirements for the specific requirements that informed development of this specification.

It is inappropriate to use W3C Working Drafts as reference material or to cite them as other than "work in progress". A list of current W3C Recommendations and other technical documents can be found at http://www.w3.org/TR/ .

Contents

---

1. Introduction

This specification defines an abstract data set called the ***XML Information Set*** (***Infoset***). Its purpose is to provide a consistent set of definitions for use in other specifications that need to refer to the information in a well-formed XML document [XML].

It does not attempt to be exhaustive; the primary criterion for inclusion of an information item or property has been that of expected usefulness in future specifications.

An XML document has an information set if it is well-formed and satisfies the namespace constraints described below. There is no requirement for an XML document to be valid in order to have an information set.

An XML document's information set consists of a number of ***information items*** (the information set for any well-formed XML document will contain at least a document information item and several others). An information item is an abstract representation of some part of an XML document: each information item has a set of associated ***properties***. The types of information item are listed in section 2.

The XML Information Set does not require or favour a specific interface or class of interfaces. This specification presents the information set as a modified 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 XML Information Set.

The terms "information set" and "information item" are similar in meaning to the generic terms "tree" and "node", as they are used in computing. However, the latter terms were avoided in this specification to reduce possible confusion with other specific data models. Information items do *not* map one-to-one with the Nodes of the DOM or the "tree" and "nodes" of the XPath data model.

Namespaces

XML 1.0 documents that do not conform to [Namespaces], though technically well-formed, are not considered to have meaningful information sets. That is, this specification does not define an information set for documents that have element or attribute names containing colons that are used in other ways than as prescribed by [Namespaces].

Furthermore, this specification does not define an information set for documents which use relative URI references in namespace declarations. This is in accordance with the decision of the W3C XML Plenary Interest Group described in [Relative Namespace URI References]. Thus the value of a [namespace name] property is always an absolute URI with an optional fragment identifier.

Entities

An information set describes its XML document with entity references already expanded, that is, represented by the information items corresponding to their replacement text. However, there are various circumstances in which a processor may not perform this expansion. An entity may not be declared, or may not be retrievable. A non-validating processor may choose not to read all declarations, and even if it does may not expand all external entities. In these cases an unexpanded entity reference information item is used to represent the entity reference.

Base URIs

Several information items have a [base URI] property. This is computed according to [XML Base]. Note that retrieval of a resource may involve redirection at the

parser level (for example, in an entity resolver) or below; in this case the base URI is the final URI used to retrieve the resource after all redirection.

Null

Some properties may sometimes have the value ***null***, in which case it is said the the property is null. Null is a value distinct from all others. In particular it is distinct from the empty string, which is simply a string containing no characters.

Synthetic Infosets

This specification describes the information set resulting from parsing an XML document. Information sets may be constructed by other means, for example by use of an API such as the DOM or by transforming an existing information set.

An information set corresponding to a real document will necessarily be consistent in various ways; for example the [in-scope namespaces] property of an element will be consistent with the [namespace attributes] properties of the element and its ancestors. This may not be true of an information set constructed by other means; in such a case there will be no XML document corresponding to the information set, and to serialize it will require resolution of the inconsistencies (for example, by outputting namespace declarations that correspond to the namespaces in scope).

> ***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 editor's knowledge and belief, the information set scheme described in this document satisfies the requirements of the XPointer-Information Set Liaison Statement [XPointer-Liaison].*

2. Information Items

An information set can contain up to seventeen different types of information items, as explained in the following sections. Every information item has properties. For ease of reference, each property is given a name, indicated **[thus]** .

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 exactly one ***document information item*** in the information set, and all other information items are accessible from the properties of the document information item, either directly or indirectly through the properties of other

information items.

The document information item has the following properties:

1. **[children]** An ordered list of child information items, in document order. The list contains exactly one element information item. The list also contains one processing instruction information item for each processing instruction outside the document element, and one comment information item for each comment outside the document element. Processing instructions and comments within the DTD are excluded. If there is a document type declaration, the list will also contain a document type declaration information item.
2. **[document entity]** The entity information item corresponding to the document entity.
3. **[document element]** The element information item corresponding to the document element.
4. **[notations]** An unordered set of notation information items, one for each notation declaration in the DTD.
5. **[entities]** An unordered set of internal, external, and unparsed entity information items, one for each general entity declared in the DTD. There are always entity information items for the predefined entities (`lt`, `gt`, `amp`, `apos`, and `quot`) even if they are not declared. There is also an entity information item for the document entity.
6. **[base URI]** The base URI of the document entity, if that is known. If it is not known (because the document was parsed from a socket connection or from standard input, for example), this property is null. This property has the same value as the [base URI] property of the [document entity].
7. **[standalone]** An indication of the standalone status of the document, either "yes", "no". This property is derived from the optional standalone document declaration in the XML declaration at the beginning of the document entity, and is null if there is no standalone document declaration.
8. **[version]** A string representing the XML version of the document. This property is derived from the XML declaration optionally present at the beginning of the document entity, and is null if there is no XML declaration.
9. **[all declarations processed]** This property is not strictly speaking part of the infoset of the document. Rather it is an indication of whether the processor has read the complete DTD. Its value is a boolean. If it is false, then null values for certain properties (indicated in their descriptions below) may reflect the fact that a relevant declaration has not been read. If it is true, those null values mean that there is no such declaration.

2.2. Element Information Items

XML Definition: *element (Section 3, Logical Structures)*

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

There is an ***element information item*** for each element appearing in the XML document. One of the element information items corresponds to the document

element (the root of the element tree), and all other element information items are children of the document element, either directly or indirectly.

An element information item has the following properties:

1. **[namespace name]** The namespace name, if any, of the element type. If the element does not belong to a namespace, this property is null.
2. **[local name]** The local part of the element-type name. This does not include any namespace prefix or following colon.
3. **[prefix]** The namespace prefix part of the element-type name. If the element does not belong to a namespace, this property is an empty string.
4. **[children]** An ordered list of child information items, in document order. This list contains element, processing instruction, unexpanded entity reference, character, and comment information items, one for each element, processing instruction, reference to an unprocessed external entity, data character, and comment appearing immediately within the current element. If the element content includes any entity references, the list will also include pairs of entity start marker and entity end marker information items, one pair for each entity reference. If the element content includes any CDATA sections, the list will also include pairs of CDATA start marker and CDATA end marker information items, one pair for each CDATA section. If the element is empty, this list has no members.
5. **[attributes]** An unordered set of attribute information items, one for each of the attributes (specified or defaulted from the DTD) of this element. Namespace declarations do not appear in this set. If the element has no attributes, this set has no members.
6. **[namespace attributes]** An unordered set of attribute information items, one for each of the namespaces declared either in the start-tag of this element or provided in the DTD for this element type. If there are no such namespace declarations, this list has no members. By definition, all namespace attributes (including those named `xmlns`) have a namespace URI of `http://www.w3.org/2000/xmlns/`.
7. **[in-scope namespaces]** An unordered set of namespace information items, one for each of the namespaces in effect for this element. This set always contains an item with the prefix `xml` which is implicitly bound to the namespace name `http://www.w3.org/XML/1998/namespace`. It does not contain an item with the prefix `xmlns` (used for declaring prefixes), since an application can never encounter an element or attribute with that prefix. The set will include namespace items corresponding to all of the members of [namespace attributes], except for any representing a declaration in the form `xmlns=""`, which does not declare a namespace but rather undeclares the default namespace.
8. **[base URI]** The base URI of the element, as computed by the method of XML Base [XML Base]. If the element appears directly in the document entity, and the URI of the document entity is not known, this property may be null.
9. **[parent]** The document or element information item which contains this information item in its [children] property.

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 an ***attribute information item*** for each attribute (specified or defaulted) of each element in the document, including those which are namespace declarations. The latter however appear as members of an element's [namespace attributes] property rather than its [attributes] property.

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.

> ***Note:*** *The XML Information Set does not include a [children] property for attributes. The childNodes attribute provided by the Document Object Model, Level 1 Core Recommendation [DOM] is not completely and consistently specified; it cannot be the normalized value since it may contain entity references, but if it is the unnormalized value it does not contain the information about character references needed for normalization. Furthermore, DOM implementations vary in their interpretation of the attribute. The XML Core Working Group has decided that it is therefore not useful to specify a corresponding property in the XML Information Set.*

An attribute information item has the following properties:

1. **[namespace name]** The namespace name, if any, of the attribute. Otherwise, this property is null.
2. **[local name]** The local part of the attribute's name. This does not include any namespace prefix or following colon.
3. **[prefix]** The namespace prefix part of the element-type name. If the element does not belong to a namespace, this property is an empty string.
4. **[normalized value]** The normalized attribute value (see 3.3.3 Attribute-Value Normalization [XML]).
5. **[specified]** A flag indicating whether this attribute was actually specified in the start-tag of its element, or was defaulted from the DTD.
6. **[attribute type]** An indication of the type declared for this attribute in the DTD. Legitimate values are ID, IDREF, IDREFS, ENTITY, ENTITIES, NMTOKEN, NMTOKENS, NOTATION, CDATA, and ENUMERATION. If no declaration has been read for the attribute, this property is null; if the [all declarations processed] property of the document information item is true, this can only happen if no such declaration exists.
7. **[owner element]** The element information item which contains this information item in its [attributes] property.

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.

A processing instruction information item has the following properties:

1. **[target]** A string representing the target part of the processing instruction (an XML name).
2. **[content]** A string representing the content of the processing instruction, excluding the target and any whitespace immediately following it. If there is no such content, the value of this property will be an empty string.
3. **[base URI]** The base URI of the PI, as computed by the method of XML Base [XML Base]. If the PI appears directly in the document entity, and the URI of the document entity is not known, this property may be null. Note that if an infoset is serialized as an XML document, it will not be possible to preserve the base URI of any PI that originally appeared at the top level of an external entity, since there is no syntax for PIs corresponding to the `xml:base` attribute on elements.
4. **[parent]** The document, element, or document type definition information item which contains this information item in its [children] property.

2.5. Unexpanded Entity Reference Information Items

XML Definition: *Section 4.4.3, Included If Validating*

A ***unexpanded entity reference information item*** serves as a place-holder by which an XML processor can indicate that it has not expanded an external parsed entity. There is one such information item for each unexpanded reference to an external general entity within the content of an element. A validating XML processor, or a non-validating processor that reads all external general entities, will never generate unexpanded entity reference information items for a valid document.

An unexpanded entity reference information item has the following properties:

1. **[name]** The name of the entity referenced.
2. **[entity]** The internal or external entity information item for the unexpanded entity reference. If no declaration has been read for the entity, this property is null; if the [all declarations processed] property of the document is true, this can only happen if no such declaration exists.
3. **[parent]** The element information item which contains this information item in its [children] property.

2.6. Character Information Items

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

There is one ***character information item*** for each character that appears within the document element, either literally, as a character reference, or within a CDATA section.

Note, however, that the characters represented are those present after the end-of-line normalization described in [XML], 2.11 "End-of-Line Handling".

Each character is a logically separate information item, but XML applications are free to chunk characters into larger groups as necessary or desirable.

A character information item has the following properties:

1. **[character code]** The ISO 10646 character code (in the range 0 to #x10FFFF, though not every value in this range is a legal XML character code) of the character.
2. **[element content whitespace]** A boolean indicating whether the character is whitespace appearing within element content (see [XML], 2.10 "White Space Handling"). Note that validating XML processors are *required* by XML 1.0 to provide this information. If no declaration has been read for the containing element, this property is null; if the [all declarations processed] property of the document is true, this can only happen if no such declaration exists.
3. **[parent]** The element information item which contains this information item in its [children] property.

2.7. Comment Information Items

XML Definition: *comment (Section 2.5, Comments)*

XML Syntax: *[15] Comment (Section 2.5, Comments)*

A ***comment information item*** corresponds to each XML comment in the original document.

A comment information item has the following properties:

1. **[content]** A string representing the content of the comment.
2. **[parent]** The document, element, or document type declaration information item which contains this information item in its [children] property.

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 contains a single ***document type declaration information item***. Note that entities and notations are provided as properties of the document information item, not the document type declaration information item.

A document type declaration information item has the following properties:

1. **[system identifier]** The system identifier of the external subset, if it exists. Otherwise this property is null.
2. **[public identifier]** The public identifier of the external subset, if it exists. Otherwise this property is null.
3. **[children]** An ordered list of comment information items and processing instruction information items representing comments and processing instructions appearing in the DTD, in the original document order. Items from the internal DTD subset appear before those in the external subset.
4. **[parent]** The document information item.

2.9. Internal Entity Information Items

XML Definition: *entity (section 4, Physical Structures)*

XML Syntax: *[71] GEDecl (section 4.2, Entities)*

There is an ***internal entity information item*** for each internal general entity declared in the DTD. There are always entity information items for the predefined entities (`lt`, `gt`, `amp`, `apos`, and `quot`) even if they are not declared. If they are declared, their replacement texts are those declared, otherwise they are the values given by the example declarations in [XML].

An internal entity information item has the following properties:

1. **[name]** The name of the entity.
2. **[content]** The replacement text of the entity.

2.10. External Entity Information Items

XML Definition: *entity (section 4, Physical Structures)*

XML Syntax: *[71] GEDecl (section 4.2, Entities)*

There is an ***external entity information item*** for each external general entity declared in the DTD. There is also an external entity information item for the document entity.

An external entity information item has the following properties:

1. **[name]** The name of the entity. If the information item represents the document entity, this property is null.
2. **[system identifier]** The system identifier of the entity. If the information item represents the document entity this property *may* be null. In all other cases, the property must not be null.
3. **[public identifier]** The public identifier of the entity, if one is available. If no public identifier is available this property is null.
4. **[base URI]** The base URI of the entity. If the information item represents an entity which has not been read, this property is always null, and if it represents the document entity, the property *may* be null. In all other cases, the property must not be null.
5. **[charset]** The name of the character encoding in which the entity is expressed. This property is derived either from the encoding declaration optionally present at the beginning of the entity, or from a MIME header. This property is null in the case of an entity which has not been read.

2.11. Unparsed Entity Information Items

XML Definition: *entity (section 4, Physical Structures)*

XML Syntax: *[71] GEDecl (section 4.2, Entities)*

There is an ***unparsed entity information item*** for each unparsed general entity declared in the DTD.

An unparsed entity information item has the following properties:

1. **[name]** The name of the entity.
2. **[system identifier]** The system identifier of the entity.
3. **[public identifier]** The public identifier of the entity, if one is available. If no public identifier is available, this property is null.
4. **[notation]** The notation information item associated with the entity.

2.12. Notation Information Items

XML Definition: *notation (section 4.7, Notations)*

XML Syntax: *[82] NotationDecl (section 4.7, Notations)*

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

A notation information item has the following properties:

1. **[name]** The name of the notation.
2. **[system identifier]** The system identifier of the notation, if one was specified. If not, the property is null.
3. **[public identifier]** The public identifier of the notation, if one was specified. If not, the property is null.

2.13. Entity Start Marker Information Items

XML Definition: *entity reference (section 4.1, Character and Entity References)*

XML Syntax: *[68] EntityRef (section 4.1, Character and Entity References)*

Entity start marker information items are inserted just before the point where information items resulting from the inclusion of a general entity as a consequence of an entity reference begin.

Entity start marker information items are not used in connection with parameter entity references in the DTD.

An entity start marker information item has the following properties:

1. **[entity]** The entity information item referred to by the entity reference which triggered the insertion of this information item.
2. **[parent]** The element information item which contains this information item in its [children] property.

2.14. Entity End Marker Information Items

XML Definition: *entity reference (section 4.1, Character and Entity References)*

XML Syntax: *[68] EntityRef (section 4.1, Character and Entity References)*

Entity end marker information items are inserted just after the point where information items resulting from the inclusion of a general entity as a consequence of an entity reference end.

Entity end marker information items are not used in connection with parameter entity references in the DTD.

An entity end marker information item has the following properties:

1. **[entity]** The entity information item referred to by the entity reference which triggered the insertion of this information item.
2. **[parent]** The element information item which contains this information item in its [children] property.

2.15. Namespace Information Items

There is one ***namespace information item*** for each namespace in scope for each element in the document.

A namespace information item has the following properties:

1. **[prefix]** The prefix whose binding this item describes. Syntactically, this is the part of the attribute name following the `xmlns:` prefix. If the attribute name is simply `xmlns`, this property is an empty string.
2. **[namespace name]** The namespace name to which the prefix is bound.

2.16. CDATA Start Marker Information Items

XML Definition: *CDATA sections (section 2.7, CDATA sections)*

XML Syntax: *[18] CDSect (section 2.7, CDATA Sections)*

CDATA start marker information items are inserted just before the place where text embedded in a CDATA section begins.

A CDATA start marker information item has the following properties:

1. **[parent]** The element information item which contains this information item in its [children] property.

2.17. CDATA End Marker Information Items

XML Definition: *CDATA sections (section 2.7, CDATA sections)*

XML Syntax: *[18] CDSect (section 2.7, CDATA Sections)*

CDATA end marker information items are inserted just after the place where text embedded in a CDATA section ends.

A CDATA end marker information item has the following properties:

1. **[parent]** The element information item which contains this information item in its [children] property.

3. Conformance

Since the purpose of the Information Set is to provide a set of definitions, conformance is a property of specifications that use those definitions, rather than of implementations.

Specifications referring to the Infoset must:

- Indicate the information items and properties that are needed to implement the specification. (This indirectly imposes conformance requirements on processors used to implement the specification.)
- Specify how other information items and properties are treated (for example,

they might be passed through unchanged).
- Note any information required from an XML document that is not defined by the infoset.
- Note any difference in the use of terms defined by the Infoset (this should be avoided).

If a specification allows the construction of an infoset that has inconsistencies as described above under Synthetic Infosets it may describe how those inconsistencies are to be resolved, and should do so if it provides for serialization of the infoset.

Appendix A. References

Normative References

ISO/IEC 10646
ISO (International Organization for Standardization). *ISO/IEC 10646-1993 (E). Information technology -- Universal Multiple-Octet Coded Character Set (UCS) -- Part 1: Architecture and Basic Multilingual Plane.* [Geneva]: International Organization for Standardization, 1993 (plus amendments AM 1 through AM 7).

Namespaces
Namespaces in XML, eds. Tim Bray, Dave Hollander, 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 (Second Edition), eds. Tim Bray, Jean Paoli, C.M. Sperberg-McQueen, Eve Maler. 6 October 2000. Available at http://www.w3.org/TR/REC-xml.

XML Base
XML Base, ed. Jonathan Marsh. February 2000. Available at http://www.w3.org/TR/xmlbase.

Informative References

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

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

Relative Namespace URI References
Results of W3C XML Plenary Ballot on relative URI References in namespace declarations, 3-17 July 2000, eds. Dave Hollander, C. M. Sperberg-McQueen. 6 September 2000. Available at

http://www.w3.org/2000/09/xppa.

Appendix B: XML 1.0 Reporting Requirements (informative)

Although the XML 1.0 Recommendation [XML] is primarily concerned with XML syntax, it also includes some specific reporting requirements for XML 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 (numbers in parentheses refer to sections of the XML Recommendation):

1. An XML processor must always provide all characters in a document that are not part of markup to the application (2.10).
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 normalize line-ends to LF before passing them to the application (2.11).
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. An 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 (3.2.2).

Appendix C: Example (informative)

Consider the following example XML document:

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

<msg:message doc:date="19990421"
                xmlns:doc="http://www.doc.example/namespaces/doc"
                xmlns:msg="http://www.message.example/"
```

```
>Phone home!</msg:message>
```

The information set for this XML document contains the following information items:

- A document information item.
- An external entity information item for the document entity.
- Five internal entity information items for the built-in entities.
- An element information item with namespace name "`http://www.message.example/`", local part "`message`", and prefix "`msg`".
- An attribute information item with the namespace name "`http://www.doc.example/namespaces/doc`", local part "`date`", prefix "`doc`", and normalized value "`19990421`".
- Three namespace information items for the `http://www.w3.org/XML/1998/namespace`, `http://www.doc.example/namespaces/doc`, and `http://www.message.net/` namespaces.
- Two attribute information items for the namespace attributes.
- Eleven character information items for the character data.

Appendix D: What is not in the Information Set

The following information is not represented in the current version of the XML Information Set (this list is not intended to be exhaustive):

1. The content models of elements, from ELEMENT declarations in the DTD.
2. The grouping and ordering of attribute declarations in ATTLIST declarations.
3. The document type name.
4. Whitespace outside the document element.
5. Whitespace immediately following the target name of a PI.
6. Whether characters are represented by character references.
7. The difference between the two forms of an empty element: `<foo/>` and `<foo></foo>`.
8. Whitespace within start-tags (other than significant whitespace in attribute values) and end-tags.
9. The difference between CR, CR-LF, and LF line termination.
10. The order of attributes within a start-tag.
11. The order of declarations within the DTD.
12. The boundaries of conditional sections in the DTD.
13. The boundaries of parameter entities in the DTD.
14. The location of declarations (whether in internal or external subset or parameter entities).
15. Any ignored declarations, including those within an IGNORE conditional section, as well as entity and attribute declarations ignored because previous declarations override them.
16. The kind of quotation marks (single or double) used to quote attribute values.
17. Whether an external entity has a text declaration.

Appendix E: RDF Schema (informative)

The following RDF Schema provides a formal characterization of the Infoset. In case of disagreement between this schema and the prose in this document, the prose is normative.

```
<?xml version='1.0' encoding='utf-8' standalone='yes'?>
<!-- this can be decoded as US-ASCII or iso-8859-1 as well,
     since it contains no characters outside the US-ASCII repertoire -->
<!-- $Id$ -->
<rdf:RDF xmlns:rdf='http://www.w3.org/1999/02/22-rdf-syntax-ns#'
 xmlns:rdfs='http://www.w3.org/2000/01/rdf-schema#'
 xmlns='http://www.w3.org/2001/02/infoset#'>

<!--

 This RDF schema's namespace name (http://www.w3.org/2001/02/infoset#)
 will only be used to describe the infoitems and properties defined in
 the corresponding version of the XML Infoset specification.  Any new
 version of the specification that changes the infoitems or properties
 will have a new RDF schema with a different namespace name.

 This RDF schema for the XML Infoset is not a normative part of the
 XML Infoset Specification.  If this schema is found not to match
 the normative text of the specification, it will be corrected without
 changing the namespace name.

-->

<!--Enumeration classes and their members-->

<rdfs:Class ID='AttributeType'/>
<AttributeType ID='AttributeType.ID'/>
<AttributeType ID='AttributeType.IDREF'/>
<AttributeType ID='AttributeType.IDREFS'/>
<AttributeType ID='AttributeType.ENTITY'/>
<AttributeType ID='AttributeType.ENTITIES'/>
<AttributeType ID='AttributeType.NMTOKEN'/>
<AttributeType ID='AttributeType.NMTOKENS'/>
<AttributeType ID='AttributeType.NOTATION'/>
<AttributeType ID='AttributeType.CDATA'/>
<AttributeType ID='AttributeType.ENUMERATION'/>

<rdfs:Class ID='Boolean'/>
<Boolean ID='Boolean.true'/>
<Boolean ID='Boolean.false'/>

<rdfs:Class ID='Integer'
 rdfs:subClassOf='http://www.w3.org/2000/01/rdf-schema#Literal'/>

<rdfs:Class ID='Standalone'/>
<StandaloneType ID='Standalone.yes'/>
<StandaloneType ID='Standalone.no'/>

<!--Info item classes-->

<rdfs:Class ID='InfoItem'/>

<rdfs:Class ID='Document' rdfs:subClassOf='#InfoItem'/>
```

```
<rdfs:Class ID='Element' rdfs:subClassOf='#InfoItem'/>

<rdfs:Class ID='Attribute' rdfs:subClassOf='#InfoItem'/>

<rdfs:Class ID='ProcessingInstruction' rdfs:subClassOf='#InfoItem'/>

<rdfs:Class ID='Character' rdfs:subClassOf='#InfoItem'/>

<rdfs:Class ID='UnexpandedEntityReference' rdfs:subClassOf='#InfoItem'/>

<rdfs:Class ID='Comment' rdfs:subClassOf='#InfoItem'/>

<rdfs:Class ID='DocumentTypeDeclaration' rdfs:subClassOf='#InfoItem'/>

<rdfs:Class ID='Entity' rdfs:subClassOf='#InfoItem'/>

<rdfs:Class ID='InternalEntity' rdfs:subClassOf='#Entity'/>

<rdfs:Class ID='ExternalEntity' rdfs:subClassOf='#Entity'/>

<rdfs:Class ID='UnparsedEntity' rdfs:subClassOf='#Entity'/>

<rdfs:Class ID='Notation' rdfs:subClassOf='#InfoItem'/>

<rdfs:Class ID='EntityStartMarker' rdfs:subClassOf='#InfoItem'/>

<rdfs:Class ID='EntityEndMarker' rdfs:subClassOf='#InfoItem'/>

<rdfs:Class ID='Namespace' rdfs:subClassOf='#InfoItem'/>\

<rdfs:Class ID='CDATAStartMarker' rdfs:subClassOf='#InfoItem'/>

<rdfs:Class ID='CDATAEndMarker' rdfs:subClassOf='#InfoItem'/>

<!--Set containers-->

<rdfs:Class ID='InfoItemSet'
  rdfs:subClassOf='http://www.w3.org/1999/02/22-rdf-syntax-ns#Bag'/>

<rdfs:Class ID='AttributeSet' rdfs:subClassOf='#InfoItemSet'/>

<rdfs:Class ID='EntitySet' rdfs:subClassOf='#InfoItemSet'/>

<rdfs:Class ID='NamespaceSet' rdfs:subClassOf='#InfoItemSet'/>

<rdfs:Class ID='NotationSet' rdfs:subClassOf='#InfoItemSet'/>


<!--Sequence container-->

<rdfs:Class ID='InfoItemSeq'
 rdfs:subClassOf='http://www.w3.org/1999/02/22-rdf-syntax-ns#Seq'/>


<!--Info item properties-->

<rdfs:Property ID='allDeclarationsProcessed'>
  <rdfs:domain resource='#Document'/>
  <rdfs:range resource='#Boolean'/>
</rdfs:Property>

<rdfs:Property ID='attributes'>
  <rdfs:domain resource='#Element'/>
  <rdfs:range resource='#AttributeSet'/>
```

```
</rdfs:Property>

<rdfs:Property ID='attributeType'>
  <rdfs:domain resource='#Attribute'/>
  <rdfs:range resource='#AttributeType'/>
</rdfs:Property>

<rdfs:Property ID='baseURI'>
  <rdfs:domain resource='#Document'/>
  <rdfs:domain resource='#Element'/>
  <rdfs:domain resource='#ProcessingInstruction'/>
  <rdfs:domain resource='#ExternalEntity'/>
  <rdfs:range resource='http://www.w3.org/TR/1999/PR-rdf-schema-19990303
</rdfs:Property>

<rdfs:Property ID='characterCode'>
  <rdfs:domain resource='#Character'/>
  <rdfs:range resource='#Integer'/>
</rdfs:Property>

<rdfs:Property ID='charset'>
  <rdfs:domain resource='#ExternalEntity'/>
  <rdfs:range resource='http://www.w3.org/2000/01/rdf-schema#Literal'/>
</rdfs:Property>

<rdfs:Property ID='children'>
  <rdfs:domain resource='#Document'/>
  <rdfs:domain resource='#Element'/>
  <rdfs:domain resource='#DocumentTypeDeclaration'/>
  <rdfs:range resource='#InfoItemSeq'/>
</rdfs:Property>

<rdfs:Property ID='content'>
  <rdfs:domain resource='#ProcessingInstruction'/>
  <rdfs:domain resource='#Comment'/>
  <rdfs:domain resource='#InternalEntity'/>
  <rdfs:range resource='http://www.w3.org/2000/01/rdf-schema#Literal'/>
</rdfs:Property>

<rdfs:Property ID='namespaceAttributes'>
  <rdfs:domain resource='#Element'/>
  <rdfs:range resource='#AttributeSet'/>
</rdfs:Property>

<rdfs:Property ID='documentElement'>
  <rdfs:domain resource='#Document'/>
  <rdfs:range resource='#Element'/>
</rdfs:Property>

<rdfs:Property ID='documentEntity'>
  <rdfs:domain resource='#Document'/>
  <rdfs:range resource='#ExternalEntity'/>
</rdfs:Property>

<rdfs:Property ID='elementContentWhitespace'>
  <rdfs:domain resource='#Character'/>
  <rdfs:range resource='#Boolean'/>
</rdfs:Property>

<rdfs:Property ID='entity'>
  <rdfs:domain resource='#UnexpandedEntityReference'/>
  <rdfs:domain resource='#EntityStartMarker'/>
  <rdfs:domain resource='#EntityEndMarker'/>
  <rdfs:range resource='#Entity'/>
```

```
</rdfs:Property>

<rdfs:Property ID='entities'>
  <rdfs:domain resource='#Document'/>
  <rdfs:range resource='#EntitySet'/>
</rdfs:Property>

<rdfs:Property ID='inScopeNamespaces'>
  <rdfs:domain resource='#Element'/>
  <rdfs:range resource='#NamespaceSet'/>
</rdfs:Property>

<rdfs:Property ID='localName'>
  <rdfs:domain resource='#Element'/>
  <rdfs:domain resource='#Attribute'/>
  <rdfs:range resource='http://www.w3.org/2000/01/rdf-schema#Literal'/>
</rdfs:Property>

<rdfs:Property ID='name'>
  <rdfs:domain resource='#UnexpandedEntityReference'/>
  <rdfs:domain resource='#InternalEntity'/>
  <rdfs:domain resource='#ExternalEntity'/>
  <rdfs:domain resource='#Notation'/>
  <rdfs:range resource='http://www.w3.org/2000/01/rdf-schema#Literal'/>
</rdfs:Property>

<rdfs:Property ID='namespaceName'>
  <rdfs:domain resource='#Element'/>
  <rdfs:domain resource='#Attribute'/>
  <rdfs:domain resource='#Namespace'/>
  <rdfs:range resource='http://www.w3.org/2000/01/rdf-schema#Literal'/>
</rdfs:Property>

<rdfs:Property ID='normalizedValue'>
  <rdfs:domain resource='#Attribute'/>
  <rdfs:range resource='http://www.w3.org/2000/01/rdf-schema#Literal'/>
</rdfs:Property>

<rdfs:Property ID='notation'>
  <rdfs:domain resource='#UnparsedEntity'/>
  <rdfs:range resource='#Notation'/>
</rdfs:Property>

<rdfs:Property ID='notations'>
  <rdfs:domain resource='#Document'/>
  <rdfs:range resource='#NotationSet'/>
</rdfs:Property>

<rdfs:Property ID='ownerElement'>
  <rdfs:domain resource='#Attribute'/>
  <rdfs:range resource='#Element'/>
</rdfs:Property>

<rdfs:Property ID='parent'>
  <rdfs:domain resource='#Element'/>
  <rdfs:domain resource='#ProcessingInstruction'/>
  <rdfs:domain resource='#Character'/>
  <rdfs:domain resource='#UnexpandedEntityReference'/>
  <rdfs:domain resource='#Comment'/>
  <rdfs:domain resource='#DocumentTypeDeclaration'/>
  <rdfs:domain resource='#EntityStartMarker'/>
  <rdfs:domain resource='#EntityEndMarker'/>
  <rdfs:domain resource='#CDATAStartMarker'/>
  <rdfs:domain resource='#CDATAEndMarker'/>
```

```
  <rdfs:range resource='#InfoItem'/>
</rdfs:Property>

<rdfs:Property ID='prefix'>
  <rdfs:domain resource='#Namespace'/>
  <rdfs:domain resource='#Element'/>
  <rdfs:domain resource='#Attribute'/>
  <rdfs:range resource='http://www.w3.org/2000/01/rdf-schema#Literal'/>
</rdfs:Property>

<rdfs:Property ID='publicIdentifier'>
  <rdfs:domain resource='#ExternalEntity'/>
  <rdfs:domain resource='#UnparsedEntity'/>
  <rdfs:domain resource='#DocumentTypeDeclaration'/>
  <rdfs:domain resource='#Notation'/>
  <rdfs:range resource='http://www.w3.org/2000/01/rdf-schema#Literal'/>
</rdfs:Property>

<rdfs:Property ID='specified'>
  <rdfs:domain resource='#Attribute'/>
  <rdfs:range resource='#Boolean'/>
</rdfs:Property>

<rdfs:Property ID='standalone'>
  <rdfs:domain resource='#Document'/>
  <rdfs:range resource='#Standalone'/>
</rdfs:Property>

<rdfs:Property ID='systemIdentifier'>
  <rdfs:domain resource='#ExternalEntity'/>
  <rdfs:domain resource='#UnparsedEntity'/>
  <rdfs:domain resource='#DocumentTypeDeclaration'/>
  <rdfs:domain resource='#Notation'/>
  <rdfs:range resource='http://www.w3.org/2000/01/rdf-schema#Literal'/>
</rdfs:Property>

<rdfs:Property ID='target'>
  <rdfs:domain resource='#ProcessingInstruction'/>
  <rdfs:range resource='http://www.w3.org/2000/01/rdf-schema#Literal'/>
</rdfs:Property>

<rdfs:Property ID='version'>
  <rdfs:domain resource='#Document'/>
  <rdfs:range resource='http://www.w3.org/2000/01/rdf-schema#Literal'/>
</rdfs:Property>

</rdf:RDF>
```