



XML Information Set (Second Edition)

W3C Recommendation 4 February 2004

This version:

<http://www.w3.org/TR/2004/REC-xml-infoset-20040204>

Latest version:

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

Previous version:

<http://www.w3.org/TR/2003/PER-xml-infoset-20031210>

Editors:

John Cowan, jcowan@reutershealth.com

Richard Tobin, richard@cogsci.ed.ac.uk

Please refer to the [errata](#) for this document, which may include some normative corrections.

See also [translations](#).

Copyright ©1999-2004 W3C® (MIT, ERCIM, Keio), All Rights Reserved. W3C [liability](#), [trademark](#), [document use](#) and [software licensing](#) rules apply.

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

This section describes the status of this document at the time of its publication. Other documents may supersede this document. A list of current W3C publications and the latest revision of this technical report can be found in the [W3C technical reports index](#) at <http://www.w3.org/TR/>.

This document is a [Recommendation](#) of the W3C. It has been reviewed by W3C Members and other interested parties, and has been endorsed by the Director as a W3C Recommendation. It is a stable document and may be used as reference material or cited as a normative reference from another document. W3C's role in making the Recommendation is to draw attention to the specification and to promote its widespread deployment. This enhances the

functionality and interoperability of the Web.

This document updates the Infoset to cover [XML 1.1](#) and [Namespaces 1.1](#), clarifies the consequences of certain kinds of invalidity, and corrects some typographical errors. It is a product of the [W3C XML Activity](#). The English version of this specification is the only normative version. However, for translations of this document, see <http://www.w3.org/2003/03/Translations/byTechnology?technology=xml-infoset>.

Documentation of intellectual property possibly relevant to this recommendation may be found at the Working Group's public [IPR disclosure page](#).

Please report errors in this document to www-xml-infoset-comments@w3.org (public [archives](#) are available). The errata list for this Recommendation is available at <http://www.w3.org/2001/10/02/xml-infoset-errata.html>.

Contents

- [1. Introduction](#)
 - [2. Information Items](#)
 - [2.1 The Document Information Item](#)
 - [2.2 Element Information Items](#)
 - [2.3 Attribute Information Items](#)
 - [2.4 Processing Instruction Information Items](#)
 - [2.5 Unexpanded Entity Reference Information Items](#)
 - [2.6 Character Information Items](#)
 - [2.7 Comment Information Items](#)
 - [2.8 The Document Type Declaration Information Item](#)
 - [2.9 Unparsed Entity Information Items](#)
 - [2.10 Notation Information Items](#)
 - [2.11 Namespace Information Items](#)
 - [3. Conformance](#)
 - [Appendix A: References](#)
 - [Appendix B: XML Reporting Requirements \(informative\)](#)
 - [Appendix C: Example \(informative\)](#)
 - [Appendix D: What is not in the Information Set](#)
 - [Appendix E: RDF Schema \(informative\)](#)
-

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. Nor does it constitute a minimum set of information that must be returned by an XML processor.

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.

Information sets may be created by methods (not described in this specification) other than parsing an XML document. See [Synthetic Infosets](#) below.

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 description of some part of an XML document: each information item has a set of associated named **properties**. In this specification, the property names are shown in square brackets, **[thus]**. The types of information item are listed in [section 2](#).

The XML Information Set does not require or favor 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 former terms are used 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.

In this specification, the words "must", "should", and "may" assume the meanings specified in [RFC2119](#), except that the words do not appear in uppercase.

XML Versions

Different versions of the XML specification may specify different parsing rules. The information set of an XML document is defined to be the one obtained by parsing it according to the rules of the specification whose version corresponds to that of the document. A document which does not specify a version number is considered to have version 1.0. If an XML processor accepts a document with a version number that it does not understand, it will not necessarily be able to produce the correct information set.

Namespaces

XML 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 [\[Relative Namespace URI References\]](#).

The value of a [namespace name] property is the normalized value of the corresponding namespace attribute; no additional URI escaping is applied to it by the processor.

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.

End-of-Line Handling

The values of all properties in the Infoset take account of the end-of-line normalization described in [\[XML\]](#), 2.11 "End-of-Line Handling".

Base URIs

Several information items have a [base URI] or [declaration base URI] property. These are 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.

The value of these properties does not reflect any URI escaping that may be required for retrieval of the resource, but it may include escaped characters if these were specified in the document, or returned by a server in the case of redirection.

In some cases (such as a document read from a string or a pipe) the rules in [\[XML Base\]](#) may result in a base URI being application dependent. In these cases this specification does not define the value of the [base URI] or [declaration base URI] property.

When resolving relative URIs the [base URI] property should be used in preference to the values of xml:base attributes; they may be inconsistent in the case of [Synthetic Infosets](#).

``Unknown" and ``No Value"

Some properties may sometimes have the value *unknown* or *no value*, and it is said that a property value is unknown or that a property has no value respectively. These values are distinct from each other and from all other values. In particular they are distinct from the empty string, the empty set, and the empty list, each of which simply has no members. This specification does not use the term **null** since in some communities it has particular connotations which may not match those intended here.

Inconsistencies Resulting from Invalidity

As noted above, an XML document need not be valid to have an information set. However, certain kinds of invalidity affect the values assigned to some properties. Entities, notations, elements and attributes may be undeclared. Notations and elements may be multiply declared (multiple declarations are valid for entities and attributes). An ID may be undefined or multiply defined. Such cases are noted where relevant in the Information Item definitions below.

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).

2. Information Items

An information set can contain up to eleven different types of information item, as explained in the following sections. Every information item has properties. For ease of reference, each property is given a name, indicated **[thus]**. Links to a definition and/or syntax in the XML 1.0 Recommendation [\[XML\]](#) are given for each information item.

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 also contains a [document type declaration](#) information item.
2. **[document element]** The [element](#) information item corresponding to the document element.
3. **[notations]** An unordered set of [notation](#) information items, one for each notation declared in the DTD. If any notation is multiply declared, this property has no value.
4. **[unparsed entities]** An unordered set of [unparsed entity](#) information items, one for each unparsed entity declared in the DTD.
5. **[base URI]** The base URI of the document entity.
6. **[character encoding scheme]** The name of the character encoding scheme in which the document entity is expressed.
7. **[standalone]** An indication of the standalone status of the document, either yes or no. This property is derived from the optional standalone document declaration in the XML declaration at the beginning of the document entity, and has no value 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 has no value 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 certain properties (indicated in their descriptions below) may be unknown. If it is true, those properties are never unknown.

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 is the value of the [document element] property of the document information item, corresponding to the root of the element tree, and all other element information items are accessible by recursively following its [children] property.

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 has no value.
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 name is unprefixed, this property has no value. Note that namespace-aware applications should use the namespace name rather than the prefix to identify elements.
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 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 namespace declarations (specified or defaulted from the DTD) of this element. Declarations of the form `xmlns=""` and `xmlns:name=""`, which undeclare the default namespace and prefixes respectively, count as namespace declarations. Prefix undeclaration was added in [Namespaces in XML 1.1](#). By definition, all namespace attributes (including those named `xmlns`, whose [prefix] property has no value) have a namespace URI of `http://www.w3.org/2000/xmlns/`. If the element has no namespace declarations, this set has no members.
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 namespaces), 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 declarations of the form `xmlns=""` or `xmlns:name=""`, which do not declare a namespace but rather undeclare the default namespace and prefixes. When resolving the prefixes of qualified names this property should be used in preference to the [namespace attributes] property; they may be inconsistent in the case of [Synthetic Infosets](#).
8. **[base URI]** The base URI of the element.
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 no default value and not specified in the element's start tag are not represented by attribute information items.

An attribute information item has the following properties:

1. **[namespace name]** The namespace name, if any, of the attribute. Otherwise, this property has no value.
2. **[local name]** The local part of the attribute name. This does not include any namespace prefix or following colon.
3. **[prefix]** The namespace prefix part of the attribute name. If the name is unprefixed, this property has no value. Note that namespace-aware applications should use the namespace name rather than the prefix to identify attributes.
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 there is no declaration for the attribute, this property has no value. If no declaration has been read, but the [all declarations processed] property of the document information item is false (so there may be an unread declaration), then the value of this property is unknown. Applications should treat no value and unknown as equivalent to a value of CDATA. The value of this property is not affected by the validity of the attribute value.
7. **[references]** If the attribute type is ID, NMTOKEN, NMTOKENS, CDATA, or ENUMERATION, this property has no value. If the attribute type is unknown, the value of this property is unknown. Otherwise (that is, if the attribute type is IDREF, IDREFS, ENTITY, ENTITIES, or NOTATION), the value of this property is an ordered list of the [element](#), [unparsed entity](#), or [notation](#) information items referred to in the attribute value, in the order that they appear there. In this case, if the attribute value is syntactically invalid, this property has no value. If the type is IDREF or IDREFS and

any of the IDs does not appear as the value of an ID attribute in the document, or if the type is ENTITY, ENTITIES or NOTATION and no declaration has been read for any of the entities or the notation, then this property has no value or is unknown, depending on whether the [all declarations processed] property of the document information item is true or false. If the type is IDREF or IDREFS and any of the IDs appears as the value of more than one ID attribute in the document, or if the type is NOTATION and there are multiple declarations for the notation, then this property has no value.

8. **[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 a **processing instruction information item** for each 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 white space 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. 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. **[notation]** The [notation](#) information item named by the target. If there is no declaration for a notation with that name, or there are multiple declarations, this property has no value. If no declaration has been read, but the [all declarations processed] property of the document information item is false (so there may be an unread declaration), then the value of this property is unknown.
5. **[parent]** The document, element, or document type declaration 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 placeholder by

which an XML processor can indicate that it has not expanded an external parsed entity. There is such an 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. **[system identifier]** The system identifier of the entity, as it appears in the declaration of the entity, without any additional URI escaping applied by the processor. If there is no declaration for the entity, this property has no value. If no declaration has been read, but the [all declarations processed] property of the document information item is false (so there may be an unread declaration), then the value of this property is unknown.
3. **[public identifier]** The public identifier of the entity, normalized as described in [4.2.2 External Entities \[XML\]](#). If there is no declaration for the entity, or the declaration does not include a public identifier, this property has no value. If no declaration has been read, but the [all declarations processed] property of the document information item is false (so there may be an unread declaration), then the value of this property is unknown.
4. **[declaration base URI]** The base URI relative to which the system identifier should be resolved (i.e. the base URI of the resource within which the entity declaration occurs). This is unknown or has no value in the same circumstances as the [system identifier] property.
5. **[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 a **character information item** for each data character that appears in the document, whether literally, as a character reference, or within a CDATA section.

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 white space appearing within element content (see [\[XML\]](#), 2.10 "White Space Handling"). Note that validating XML processors are

required to provide this information. If there is no declaration for the containing element, or there are multiple declarations, this property has no value for white space characters. If no declaration has been read, but the [all declarations processed] property of the document information item is false (so there may be an unread declaration), then the value of this property is unknown for white space characters. It is always false for characters that are not white space.

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)

There is a **comment information item** for each XML comment in the original document, except for those appearing in the DTD (which are not represented).

A comment information item has the following properties:

1. **[content]** A string representing the content of the comment.
2. **[parent]** The document or element 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, as it appears in the DOCTYPE declaration, without any additional URI escaping applied by the processor. If there is no external subset this property has no value.
2. **[public identifier]** The public identifier of the external subset, normalized as described in [4.2.2 External Entities \[XML\]](#). If there is no external subset or if it has no public identifier, this property has no value.
3. **[children]** An ordered list of [processing instruction](#) information items representing 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. 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, as it appears in the declaration of the entity, without any additional URI escaping applied by the processor.
3. **[public identifier]** The public identifier of the entity, normalized as described in [4.2.2 External Entities \[XML\]](#). If the entity has no public identifier, this property has no value.
4. **[declaration base URI]** The base URI relative to which the system identifier should be resolved (i.e. the base URI of the resource within which the entity declaration occurs).
5. **[notation name]** The notation name associated with the entity.
6. **[notation]** The [notation](#) information item named by the notation name. If there is no declaration for a notation with that name, or there are multiple declarations, this property has no value. If no declaration has been read, but the [all declarations processed] property of the document information item is false (so there may be an unread declaration), then the value of this property is unknown.

2.10. Notation Information Items

XML Definition: [notation](#) (section 4.7, Notations)

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

There is a **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, as it appears in the declaration of the notation, without any additional URI escaping applied by the processor. If no system identifier was specified, this property has no value.
3. **[public identifier]** The public identifier of the notation, normalized as

described in [4.2.2 External Entities \[XML\]](#). If the notation has no public identifier, this property has no value.

4. **[declaration base URI]** The base URI relative to which the system identifier should be resolved (i.e. the base URI of the resource within which the notation declaration occurs).

2.11. Namespace Information Items

Each element in the document has a **namespace information item** for each namespace that is in scope for that element.

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`, so that the declaration is of the default namespace, this property has no value.
2. **[namespace name]** The namespace name to which the prefix is bound.

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 10646-1:2000. Information technology — Universal Multiple-Octet Coded*

Character Set (UCS) — Part 1: Architecture and Basic Multilingual Plane and ISO/IEC 10646-2:2001. Information technology — Universal Multiple-Octet Coded Character Set (UCS) — Part 2: Supplementary Planes, as, from time to time, amended, replaced by a new edition or expanded by the addition of new parts. [Geneva]: International Organization for Standardization. (See <http://www.iso.ch> for the latest version.)

Namespaces

Namespaces in XML, W3C, eds. Tim Bray, Dave Hollander, Andrew Layman. 14 January 1999. Available at <http://www.w3.org/TR/REC-xml-names>.

Namespaces 1.1

Namespaces in XML 1.1, W3C, eds. Tim Bray, Dave Hollander, Andrew Layman, Richard Tobin. 4 February 2004. Available at <http://www.w3.org/TR/xml-names11>.

RFC2119

Key words for use in RFCs to Indicate Requirement Levels, ed. S. Bradner. March 1997. Available at <http://www.ietf.org/rfc/rfc2119.txt>.

XML

Extensible Markup Language (XML) 1.0 (Third Edition), W3C, eds. Tim Bray, Jean Paoli, C.M. Sperberg-McQueen, Eve Maler, François Yergeau. 4 February 2004. Available at <http://www.w3.org/TR/REC-xml>.

XML 1.1

Extensible Markup Language (XML) 1.1, W3C, eds. Tim Bray, Jean Paoli, C.M. Sperberg-McQueen, Eve Maler, John Cowan, François Yergeau. 4 February 2004. Available at <http://www.w3.org/TR/xml11>.

XML Base

XML Base, W3C, ed. Jonathan Marsh. February 2000. Available at <http://www.w3.org/TR/xmlbase>.

Informative References

DOM

Document Object Model (DOM) Level 1 Specification, W3C, 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, W3C, ed. Steven J. DeRose. 24 February 1999. Available at <http://www.w3.org/TR/NOTE-xptr-infoiset-liaison>.

Relative Namespace URI References

Results of W3C XML Plenary Ballot on relative URI References in namespace declarations, 3-17 July 2000, W3C, eds. Dave Hollander, C. M. Sperberg-McQueen. 6 September 2000. Available at <http://www.w3.org/2000/09/xppa>.

RDF Schema for the XML Information Set

RDF Schema for the XML Information Set, W3C, ed. Richard Tobin. 6 April 2001. Available at <http://www.w3.org/TR/xml-infoiset-rdfs>.

Appendix B: XML Reporting Requirements (informative)

Although the XML 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 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 white space 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 3.3.3 before passing them to the application.
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 (3.3.2).

Appendix C: Example (informative)

Consider the following example XML document:

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

<msg:message doc:date="19990421"
              xmlns:doc="http://doc.example.org/namespaces/doc"
              xmlns:msg="http://message.example.org/"
>Phone home!</msg:message>
```

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

- A [document](#) information item.
- An [element](#) information item with namespace name "http://message.example.org/", local part "message", and prefix "msg".
- An [attribute](#) information item with the namespace name "http://doc.example.org/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://doc.example.org/namespaces/doc`, and `http://message.example.org/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. White space outside the document element.
5. White space 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. White space within start-tags (other than significant white space 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. Comments in the DTD.
15. The location of declarations (whether in internal or external subset or parameter entities).
16. Any ignored declarations, including those within an IGNORE conditional section, as well as entity and attribute declarations ignored because previous declarations override them.
17. The kind of quotation marks (single or double) used to quote attribute values.
18. The boundaries of general parsed entities.
19. The boundaries of CDATA marked sections.
20. The default value of attributes declared in the DTD.

Appendix E: RDF Schema (informative)

See [RDF Schema for the XML Information Set](#) for a formal characterization of the Infoset.