xml:id Version 1.0

W3C Recommendation 9 September 2005

This version:
http://www.w3.org/TR/2005/REC-xml-id-20050909/
Latest version:
http://www.w3.org/TR/xml-id/
Previous version:
http://www.w3.org/TR/2005/PR-xml-id-20050712/
Editors:
Jonathan Marsh, Microsoft <jmarsh@microsoft.com>
Daniel Veillard, Invited Expert <daniel@veillard.com>
Norman Walsh, Sun Microsystems <Norman.Walsh@Sun.COM>

Please refer to the **errata** for this document, which may include normative corrections.

See also **translations**.

This document is also available in these non-normative formats: XML.

Copyright © 2005 W3C® (MIT, ERCIM, Keio), All Rights Reserved. W3C liability, trademark and document use rules apply.

Abstract

This document defines the meaning of the attribute `xml:id` as an ID attribute in XML documents and defines processing of this attribute to identify IDs in the absence of validation, without fetching external resources, and without relying on an internal subset.

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. This document has been developed by the W3C XML Core Working Group as part of the XML Activity. It has been reviewed by W3C Members and other interested parties and has been endorsed by the Director. 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.

The English version of this specification is the only normative version. Translations of this document may be available.

If you have any comments on this document, send them to public-xml-id@w3.org, a mailing list with a public archive. An errata list for this edition is available.

This document defines the meaning of the attribute `xml:id` as an ID attribute in XML documents and defines processing of this attribute to identify IDs in the absence of validation, without fetching external resources, and without relying on an internal DTD subset.

This document is based upon the *xml:id Version 1.0 Proposed Recommendation* of 12 July 2005. Feedback received during that review resulted in minor editorial changes. Evidence of interoperation between at least two implementations of this specification are documented in the Implementation Report. A test suite is also available.

This document was produced under the 5 February 2004 W3C Patent Policy. The Working Group maintains a public list of patent disclosures relevant to this document; that page also includes instructions for disclosing [and excluding] a patent. An individual who has actual knowledge of a patent which the individual believes contains Essential Claim(s) with respect to this specification should disclose the information in accordance with section 6 of the W3C Patent Policy.

Table of Contents

Appendices

---

1 Introduction

[XML 1.0] and [XML 1.1] provide a mechanism for annotating elements with unique identifiers. This mechanism consists of declaring the type of an attribute as "ID", after which the parser will validate that

- the ID value matches the allowed lexical form,
- the value is unique within the XML document, and that
- each element has at most one single unique identifier

Declarations in either the internal or external DTD subset of an XML document can declare attributes to be of type ID. However, processing the external DTD subset is optional for conformant XML processors; and some specifications, notably [SOAP], forbid DTDs completely, leaving no guarantee that all consumers of the XML document will be able to successfully recognize the identifiers.

Identifiers can be declared through external mechanisms as well. Of particular interest is [XML Schemas] which provides a type "xs:ID" with the same uniqueness and validity constraints as XML. However, there are no guarantees that consumers will have the "correct" schema available, nor that they will process it if they do.

A mechanism allowing unique element identifiers to be recognized by all conformant XML processors, whether they validate or not, is desirable in making XML sub-resource linking robust. This specification allows authors to identify elements with IDs that can be recognized by any processor without regard to how, or if, any internal or external declarations are available.

An additional problem is that DTD-based and XML Schema-based identifiers are exposed through different conceptual mechanisms - the *attribute type* infoset property, and the *type definition* family of properties respectively. A uniform mechanism for recognizing identifiers is desirable.

This specification provides such a mechanism: it describes the semantics of `xml:id` attributes. This specification has been designed to be a separate layer in processing and to be compatible with existing validation technologies.

Implementors are encouraged to support `xml:id` processing and to make ID type assignment the default behavior of their processors.

It has been a guiding principle in the design of this specification that the result of `xml:id` processing should be the same as if an appropriate declaration has been seen and used by the processor.

2 Terminology

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

[Definition: An **xml:id processor** is a software module that works in conjunction with an XML processor to provide access to the IDs in an XML document.]

[Definition: An **xml:id error** is a non-fatal error that occurs when an xml:id processor finds that a document has violated the constraints of this specification.]

Validation is the process of comparing an XML document (or part of an XML document) against a grammar or set of rules to determine if the actual structure of the document satisfies the constraints of the grammar or the rules. Some validation technologies also perform type assignment, determining not only if the document satisfies the specified constraints but also determining, for example, which (elements and/or) attributes are of type “ID”.

Although often performed together, validation and type assignment are not the same process. A non-validating XML 1.0 processor, for example, can perform type assignment using only declarations from the internal subset, without ever having any information about the structural validity of the document.

[Definition: The process of **ID type assignment** causes an `xml:id` attribute value to be an ID.] This is often achieved by making the type of the attribute "ID" in the infoset or post-schema-validation infoset (PSVI), but that is not the only possible mechanism.

> **Note:**
>
> Application-level processing of IDs, including which elements can actually be addressed by which ID values, is beyond the scope of this specification.

3 Syntax

Per [Namespaces in XML] (and [Namespaces in XML 1.1]), prefixes beginning “xml” are reserved for use by XML and XML-related specifications. This specification licenses the use of the attribute “xml:id” for use as a common syntax for identifiers in XML with the semantics specified herein.

Authors of XML documents are encouraged to name their ID attributes "xml:id" to increase the interoperability of these identifiers on the Web.

In namespace-aware XML processors, the "xml" prefix is bound to the namespace name `http://www.w3.org/XML/1998/namespace` as described in Namespaces in XML [Namespaces in XML] (and [Namespaces in XML 1.1]). Note that `xml:id` can be still used by non-namespace-aware XML processors.

4 Processing `xml:id` Attributes

Each `xml:id` attribute is processed in the following way:

1. The attribute's value is normalized according to the rules for attribute-value normalization on attributes of type ID. For more details, see **E Attribute Value Normalization on IDs**.

   The infoset *normalized value* property is updated with the normalized value.

2. ID type assignment is performed with the normalized value.

An `xml:id` processor must assure that the following constraints hold for all `xml:id` attributes:

- The normalized value of the attribute is an `NCName` according to the *Namespaces in XML* Recommendation which has the same version as the document in which this attribute occurs (NCName for XML 1.0, or NCName for XML 1.1).

- The declared type of the attribute, if it has one, is “ID”. All declarations for `xml:id` attributes must specify “ID” as the type of the attribute.

An `xml:id` processor should assure that the following constraint holds:

- The values of all attributes of type “ID” (which includes all `xml:id` attributes) within a document are unique.

An xml:id error occurs for any `xml:id` attribute that does not satisfy the constraints.

The `xml:id` processor performs ID type assignment on all `xml:id` attributes, even those that do not satisfy the constraints.

An `xml:id` processor should update the infoset *references* property, as described in Section 2.3 of [XML Information Set], and update any implementation-specific structures used for cross-referencing to reflect the results of ID type assignment.

Many validation technologies impose the constraint that an XML element can have at most one attribute of type ID. That constraint is *not* imposed by `xml:id` processing.

This specification defines xml:id processing, but it is up to the application to determine when such processing occurs. Users of applications that provide facilities for modifying XML documents may reasonably expect xml:id processing to occur whenever a change is made to an ID value.

5 Informing the Application

When ID type assignment occurs, the xml:id processor must report the assigned `xml:id` attributes to the application. How this is reported is implementation dependent.

- For applications that operate conceptually on the Infoset, an `xml:id` processor can use the *attribute type* Infoset property:

  The `xml:id` processor may report the results of ID type assignment in a DTD compatible manner by setting the *attribute type* infoset property of the attribute to ID.

- For applications that operate conceptually on the PSVI, an `xml:id` processor can use the *type definition* family of PSVI properties:

  The `xml:id` processor may report the results of ID type assignment in an XML Schema compatible manner by setting the PSVI *type definition* property of the attribute to `xs:ID`.

- For applications that operate on data models defined in other ways, the mechanisms are implementation dependent:

  The `xml:id` processor may report the results of ID type assignment in some other way.

The key requirement is that the application be made aware of the results of ID type assignment.

6 Errors

A violation of the constraints in this specification results in an xml:id error. Such errors are not fatal, but should be reported by the xml:id processor. In the interest of interoperability, it is strongly recommended that `xml:id` errors not be silently ignored.

7 Conformance

7.1 Conformance to xml:id

Conformance to `xml:id` for applications that rely on XML processors using validation technologies consists in the use of the `xml:id` construct as explained in **4**

Processing xml:id Attributes and by conformance to both the constraints of this specification and the rules of the validation technology.

Conformance to `xml:id` for applications that rely on non-validating XML processors is defined by the recognition of `xml:id` attributes as explained in **4 Processing xml:id Attributes** and by conformance to the constraints of this specification.

Conformance to constraints that "must" be assured is mandatory. It is recommended that applications assure the other constraints as well. This specification defines no simply optional constraints.

A document is conformant to this specification if it generates no xml:id errors.

7.2 XML Information Set Conformance

This specification conforms to the [XML Information Set]. The following information items must be present in the input infosets to enable correct processing:

- *Element Information Items* with *attributes* property.
- *Attribute Information Items* with *namespace name*, *local name* and *normalized value* properties.

In addition, the following properties might be present in the output infoset:

- *attribute type* properties on *Attribute Information Items*.

8 Extensibility

This specification is not extensible. There are no provisions for application designers to alter the name of the `xml:id` attribute, the set of attribute values that are considered IDs, the location(s) where they can occur, or make any other extensions.

A References

IETF RFC 2119
RFC 2119: Key words for use in RFCs to Indicate Requirement Levels. Internet Engineering Task Force, 1997. (See http://www.ietf.org/rfc/rfc2119.txt.)

XML 1.0
Tim Bray, Jean Paoli, C.M. Sperberg-McQueen, et. al., editors. *Extensible Markup Language (XML) 1.0 (Third Edition).* World Wide Web Consortium, 2004. (See http://www.w3.org/TR/REC-xml/.)

XML 1.1
Tim Bray, Jean Paoli, C.M. Sperberg-McQueen, et. al., editors. *Extensible Markup Language (XML) 1.1.* World Wide Web Consortium, 2004. (See http://www.w3.org/TR/xml11/.)

XML Information Set

John Cowan and Richard Tobin, editors. *XML Information Set (Second Edition).* World Wide Web Consortium, 2004. (See http://www.w3.org/TR/xml-infoset/.)

Namespaces in XML

Tim Bray, Dave Hollander, and Andrew Layman, editors. *Namespaces in XML*. World Wide Web Consortium, 1999. (See http://www.w3.org/TR/REC-xml-names/.)

Namespaces in XML 1.1

Tim Bray, Dave Hollander, Andrew Layman, and Richard Tobin, editors. *Namespaces in XML 1.1*. World Wide Web Consortium, 2004. (See http://www.w3.org/TR/xml-names11/.)

B References (Non-Normative)

XML Schemas

Henry S. Thompson, David Beech, Murray Maloney, Noah Mendelsohn, editors. *XML Schema Part 1: Structures.* World Wide Web Consortium, 2001. (See http://www.w3.org/TR/xmlschema-1/.)

SOAP

Martin Gudgin, Marc Hadley, Noah Mendelsohn, et. al., editors. *SOAP Version 1.2 Part 1: Messaging Framework*. World Wide Web Consortium, 2003. (See http://www.w3.org/TR/soap12-part1/.)

C Impact on Canonicalization (Non-Normative)

The Canonical XML Version 1.0 specification describes a process whereby attributes in the `xml:` namespace are inherited in a canonicalized document. While this produces a reasonable result with `xml:lang` or `xml:space` attributes, processing `xml:id` attributes in this way is likely to produce documents that contain xml:id errors, specifically `xml:id` attribute values that are not unique.

This is an apparent flaw in the design of Canonical XML. The Exclusive XML Canonicalization Version 1.0 specification does not have this feature and may be more appropriate for documents containing IDs.

D Validation Technologies (Non-Normative)

This appendix describes how `xml:id` processing interacts with selected validation technologies.

D.1 With DTD Validation

DTD authors are encouraged to use `xml:id` attributes when providing identifiers for elements declared in their DTDS.

The following DTD fragment illustrates a sample declaration for the `xml:id` attribute:

```
<!ATTLIST someElement
    xml:id     ID          #IMPLIED
>
```

DTD authors are encouraged to declare attributes named `xml:id` with the type `ID`. A document that uses `xml:id` attributes that have a declared type other than `ID` will always generate xml:id errors.

Consumers of documents validated using properly declared `xml:id` attributes can recognize IDs through the *attribute type* property.

D.2 With XML Schema Validation

XML Schema authors are encouraged to use `xml:id` attributes when providing identifiers for elements declared in their schemas. Note that this can most easily be accomplished by importing the schema for the XML namespace and using the attribute declaration it contains.

The following XML Schema fragment for the XML namespace illustrates a sample declaration for the `xml:id` attribute:

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
           targetNamespace="http://www.w3.org/XML/1998/namespace">

    <xs:attribute name="id" type="xs:ID"/>

</xs:schema>
```

XML Schema authors are encouraged to declare attributes named `xml:id` with the type `xs:ID`. A document that uses `xml:id` attributes that have a declared type other than `xs:ID` will always generate xml:id errors.

Consumers of documents validating the `xml:id` attributes against an appropriate schema for the XML namespace can recognize IDs through the *type definition* family of PSVI properties.

Applications can recognize `xml:id` attributes as IDs by conceptually using a Minimally Conforming Schema Processor and the schema above.

Note that the effects of a Minimally Conforming Schema Processor, processing the above schema, are approximated by simply looking for attributes named `xml:id`, ensuring the value of such attributes has the correct lexical form (NCName), and the value is unique within the document.

D.3 With RELAX NG Validation

RELAX NG Grammar authors are encouraged to use `xml:id` attributes when providing identifiers for elements declared in their schemas.

The following RELAX NG fragment illustrates a sample declaration for the `xml:id` attribute:

```
<optional xmlns="http://relaxng.org/ns/structure/1.0"
              datatypeLibrary="http://www.w3.org/2001/XMLSchema-datatypes">
  <attribute name="xml:id">
    <data type="ID"/>
  </attribute>
</optional>
```

RELAX NG Grammar authors are encouraged to declare attributes named `xml:id` with the type `xs:ID`. A document that uses `xml:id` attributes that have a declared type other than `xs:ID` will always generate xml:id errors.

E Attribute Value Normalization on IDs (Non-Normative)

[XML 1.0] requires parsers to normalize all attribute values. Normalization expands character references, expands entity references, and cleans up line end characters. Attributes of type ID are subject to additional normalization rules: removing leading and trailing space characters and replacing sequences of spaces with a single space.

The xml:id processor has to assure that both steps of normalization are performed all attributes named `xml:id`. In particular, the parser may not have performed the additional normalization steps required for attributes of type ID because the attribute may not be declared or may be declared as an ID.

Consider the following document:

```
<!DOCTYPE doc [
<!ATTLIST doc xml:id ID #IMPLIED>
]>
<doc xml:id="  one
">
<para xml:id="  two
"></para>
</doc>
```

The initial value of `xml:id` on `doc` will be “one” because the parser knew that it was an ID. The initial value on `para` will be “ two ”. Because the parser didn't know it was an ID, it will not have performed the additional normalizations required.

After xml:id processing, the value of the `xml:id` attributes on `doc` and `para` will be “one” and “two”, respectively. These properly normalized values will be stored in the *normalized value* property in the infoset. Performing xml:id processing changes the infoset if there are incompletely normalized `xml:id` attributes.

Note:

For interoperability, document producers should use fully normalized values that are legal `NCNames` in `xml:id` attributes.