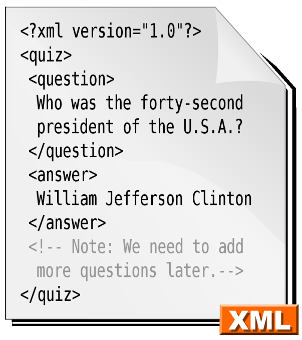
XML



```
<?xml version="1.0"?>
<quiz>
 <question>
  Who was the forty-second
  president of the U.S.A.?
 </question>
 <answer>
  William Jefferson Clinton
 </answer>
 <!-- Note: We need to add
  more questions later.-->
</quiz>
```

Filename extension	.xml
Internet media type	application/xml,[1] text/xml[2] (deprecated in an expired draft[3])
Uniform Type Identifier	public.xml
Developed by	World Wide Web Consortium
Type of format	Markup language
Extended from	SGML
Extended to	Numerous, including: XHTML, RSS, Atom
Standard(s)	1.0 (Fifth Edition) [4] November 26, 2008 1.1 (Second Edition) [5] August 16, 2006
Open format?	Yes

Current Status	Published
Year Started	1996
Editors	Tim Bray, Jean Paoli, C. M. Sperberg-McQueen, Eve Maler, François Yergeau, John Cowan
Related Standards	XML Schema
Domain	Data Serialization
Abbreviation	XML
Website	XML 1.0 [6]

Extensible Markup Language (**XML**) is a set of rules for encoding documents in machine-readable form. It is defined in the XML 1.0 Specification[7] produced by the W3C, and several other related specifications, all gratis open standards.[8]

XML's design goals emphasize simplicity, generality, and usability over the Internet.[9] It is a textual data format with strong support via Unicode for the languages of the world. Although the design of XML focuses on documents, it is widely used for the representation of arbitrary data structures, for example in web services.

Many application programming interfaces (APIs) have been developed that software developers use to process XML data, and several schema systems exist to aid in the definition of XML-based languages.

As of 2009, hundreds of XML-based languages have been developed,[10] including RSS, Atom, SOAP, and XHTML. XML-based formats have become the default for most office-productivity tools, including Microsoft Office (Office Open XML), OpenOffice.org (OpenDocument), and Apple's iWork.[11]

Key terminology

The material in this section is based on the XML Specification. This is not an exhaustive list of all the constructs which appear in XML; it provides an introduction to the key constructs most often encountered in day-to-day use.

(Unicode) Character

By definition, an XML document is a string of characters. Almost every legal Unicode character may appear in an XML document.

Processor and Application

The *processor* analyzes the markup and passes structured information to an *application*. The specification places requirements on what an XML processor must do and not do, but the application is outside its scope. The processor (as the specification calls it) is often referred to colloquially as an *XML parser*.

Markup and Content

The characters which make up an XML document are divided into *markup* and *content*. Markup and content may be distinguished by the application of simple syntactic rules. All strings which constitute markup either begin with the character "<" and end with a ">", or begin with the character "&" and end with a ";". Strings of characters which are not markup are content.

Tag

A markup construct that begins with "<" and ends with ">". Tags come in three flavors: *start-tags*, for example <section>, *end-tags*, for example </section>, and *empty-element tags*, for example <line-break/>.

Element

A logical component of a document which either begins with a start-tag and ends with a matching end-tag, or consists only of an empty-element tag. The characters between the start- and end-tags, if any, are the element's *content*, and may contain markup, including other elements, which are called *child elements*. An example of an element is <Greeting>Hello, world.</Greeting> (see hello world). Another is <line-break/>.

Attribute

A markup construct consisting of a name/value pair that exists within a start-tag or empty-element tag. In the example (below) the element *img* has two attributes, *src* and *alt*: <img&nbsp;src="madonna.jpg"&nbsp;alt='Foligno&nbsp;Madonna,&nbsp;by&nbsp;Raphael'/>. Another example would be <step number="3">Connect A to B.</step> where the name of the attribute is "number" and the value is "3".

XML Declaration

XML documents may begin by declaring some information about themselves, as in the following example.

```
<?xml version="1.0" encoding="UTF-8" ?>
```

Characters and escaping

XML documents consist entirely of characters from the Unicode repertoire. Except for a small number of specifically excluded control characters, any character defined by Unicode may appear within the content of an XML document. The selection of characters which may appear within markup is somewhat more limited but still large.

XML includes facilities for identifying the *encoding* of the Unicode characters which make up the document, and for expressing characters which, for one reason or another, cannot be used directly.

Valid characters

Unicode code points in the following ranges are valid in XML 1.0 documents:[12]

- U+0009, U+000A, U+000D: these are the only C0 controls accepted in XML 1.0;
- U+0020–U+D7FF, U+E000–U+FFFD: this excludes *some* (not all) non-characters in the BMP (all surrogates, U+FFFE and U+FFFF are forbidden);
- U+10000–U+10FFFF: this includes *all* code points in supplementary planes, including non-characters.

XML 1.1[13] extends the set of allowed characters to include all the above, plus the remaining characters in the range U+0001–U+001F. At the same time, however, it restricts the use of C0 and C1 control characters other than U+0009, U+000A, U+000D, and U+0085 by requiring them to be written in escaped form (for example U+0001 must be written as &#x01; or its equivalent). In the case of C1 characters, this restriction is a backwards incompatibility; it was introduced to allow common encoding errors to be detected.

The code point U+0000 is the only character that is not permitted in any XML 1.0 or 1.1 document.

Encoding detection

The Unicode character set can be encoded into bytes for storage or transmission in a variety of different ways, called "encodings". Unicode itself defines encodings which cover the entire repertoire; well-known ones include UTF-8 and UTF-16.[14] There are many other text encodings which pre-date Unicode, such as ASCII and ISO/IEC 8859; their character repertoires in almost every case are subsets of the Unicode character set.

XML allows the use of any of the Unicode-defined encodings, and any other encodings whose characters also appear in Unicode. XML also provides a mechanism whereby an XML processor can reliably, without any prior knowledge, determine which encoding is being used.[15] Encodings other than UTF-8 and UTF-16 will not necessarily be recognized by every XML parser.

Escaping

XML provides *escape* facilities for including characters which are problematic to include directly. For example:

- The characters "<" and "&" are key syntax markers and may *never* appear in content.[16]
- Some character encodings support only a subset of Unicode: for example, it is legal to encode an XML document in ASCII, but ASCII lacks code points for Unicode characters such as "é".
- It might not be possible to type the character on the author's machine.
- Some characters have glyphs that cannot be visually distinguished from other characters: examples are non-breaking-space (&#xa0;) and Cyrillic Capital Letter A (&#x410;).

There are five *predefined entities*: &lt; represents "<", &gt; represents ">", &amp; represents "&", &apos; represents ', and &quot; represents ". All permitted Unicode characters may be represented with a *numeric character reference*. Consider the Chinese character "中", whose numeric code in Unicode is hexadecimal 4E2D, or decimal 20,013. A user whose keyboard offered no method for entering this character could still insert it in an XML document encoded either as &#20013; or &#x4e2d;. Similarly, the string "I <3 Jörg" could be encoded for inclusion in an XML document as "I &lt;3 J&#xF6;rg".

"&#0;" is not permitted, however, as the null character is one of the control characters excluded from XML, even when using a numeric character reference.[17] An alternative encoding mechanism such as Base64 is needed to represent such characters.

Comments

Comments may appear anywhere in a document outside other markup. Comments cannot appear before the XML declaration. The string "--" (double-hyphen) is not allowed (as it is used to delimit comments). The ampersand has no special significance within comments, so entity and character references are not recognized as such, and there is no way to represent characters outside the character set of the document encoding.

An example of a valid comment: "<!-- no need to escape <code> & such in comments -->"

International use

XML supports the direct use of almost any Unicode character in element names, attributes, comments, character data, and processing instructions (other than the ones that have special symbolic meaning in XML itself, such as the less-than sign, "<"). Therefore, the following is a well-formed XML document, even though it includes both Chinese and Cyrillic characters:

```
<?xml version="1.0" encoding="UTF-8"?>
<外语>Китайська мова</外语>
```

Well-formedness and error-handling

The XML specification defines an XML document as a text which is well-formed, i.e. it satisfies a list of syntax rules provided in the specification. The list is fairly lengthy; some key points are:

- It contains only properly encoded legal Unicode characters.
- None of the special syntax characters such as "<" and "&" appear except when performing their markup-delineation roles.
- The begin, end, and empty-element tags which delimit the elements are correctly nested, with none missing and none overlapping.
- The element tags are case-sensitive; the beginning and end tags must match exactly.
- There is a single "root" element which contains all the other elements.

The definition of an *XML document* excludes texts which contain violations of well-formedness rules; they are simply not XML. An XML processor which encounters such a violation is required to report such errors and to cease normal processing. This policy, occasionally referred to as draconian, stands in notable contrast to the behavior of programs which process HTML, which are designed to produce a reasonable result even in the presence of severe markup errors. XML's policy in this area has been criticized as a violation of Postel's law.[18]

Schemas and validation

In addition to being well-formed, an XML document may be *valid*. This means that it contains a reference to a Document Type Definition (DTD), and that its elements and attributes are declared in that DTD and follow the grammatical rules for them that the DTD specifies.

XML processors are classified as *validating* or *non-validating* depending on whether or not they check XML documents for validity. A processor which discovers a validity error must be able to report it, but may continue normal processing.

A DTD is an example of a *schema* or *grammar*. Since the initial publication of XML 1.0, there has been substantial work in the area of schema languages for XML. Such schema languages typically constrain the set of elements that may be used in a document, which attributes may be applied to them, the order in which they may appear, and the allowable parent/child relationships.

DTD

The oldest schema language for XML is the Document Type Definition (DTD), inherited from SGML.

DTDs have the following benefits:

- DTD support is ubiquitous due to its inclusion in the XML 1.0 standard.
- DTDs are terse compared to element-based schema languages and consequently present more information in a single screen.
- DTDs allow the declaration of standard public entity sets for publishing characters.
- DTDs define a *document type* rather than the types used by a namespace, thus grouping all constraints for a document in a single collection.

DTDs have the following limitations:

- They have no explicit support for newer features of XML, most importantly namespaces.
- They lack expressiveness. XML DTDs are simpler than SGML DTDs and there are certain structures that cannot be expressed with regular grammars. DTDs only support rudimentary datatypes.
- They lack readability. DTD designers typically make heavy use of parameter entities (which behave essentially as textual macros), which make it easier to define complex grammars, but at the expense of clarity.
- They use a syntax based on regular expression syntax, inherited from SGML, to describe the schema. Typical XML APIs such as SAX do not attempt to offer applications a structured representation of the syntax, so it is less accessible to programmers than an element-based syntax may be.

Two peculiar features that distinguish DTDs from other schema types are the syntactic support for embedding a DTD within XML documents and for defining *entities*, which are arbitrary fragments of text and/or markup that the XML processor inserts in the DTD itself and in the XML document wherever they are referenced, like character escapes.

DTD technology is still used in many applications because of its ubiquity.

XML Schema

A newer schema language, described by the W3C as the successor of DTDs, is XML Schema, often referred to by the initialism for XML Schema instances, XSD (XML Schema Definition). XSDs are far more powerful than DTDs in describing XML languages. They use a rich datatyping system and allow for more detailed constraints on an XML document's logical structure. XSDs also use an XML-based format, which makes it possible to use ordinary XML tools to help process them.

RELAX NG

RELAX NG was initially specified by OASIS and is now also an ISO international standard (as part of DSDL). RELAX NG schemas may be written in either an XML based syntax or a more compact non-XML syntax; the two syntaxes are isomorphic and James Clark's Trang conversion tool can convert between them without loss of information. RELAX NG has a simpler definition and validation framework than XML Schema, making it easier to use and implement. It also has the ability to use datatype framework plug-ins; a RELAX NG schema author, for example, can require values in an XML document to conform to definitions in XML Schema Datatypes.

Schematron

Schematron is a language for making assertions about the presence or absence of patterns in an XML document. It typically uses XPath expressions.

ISO DSDL and other schema languages

The ISO DSDL (Document Schema Description Languages) standard brings together a comprehensive set of small schema languages, each targeted at specific problems. DSDL includes RELAX NG full and compact syntax, Schematron assertion language, and languages for defining datatypes, character repertoire constraints, renaming and entity expansion, and namespace-based routing of document fragments to different validators. DSDL schema languages do not have the vendor support of XML Schemas yet, and are to some extent a grassroots reaction of industrial publishers to the lack of utility of XML Schemas for publishing.

Some schema languages not only describe the structure of a particular XML format but also offer limited facilities to influence processing of individual XML files that conform to this format. DTDs and XSDs both have this ability; they can for instance provide the infoset augmentation facility and attribute defaults. RELAX NG and Schematron intentionally do not provide these.

Related specifications

A cluster of specifications closely related to XML have been developed, starting soon after the initial publication of XML 1.0. It is frequently the case that the term "XML" is used to refer to XML together with one or more of these other technologies which have come to be seen as part of the XML core.

- XML Namespaces enable the same document to contain XML elements and attributes taken from different vocabularies, without any naming collisions occurring. Although XML Namespaces are not part of the XML specification itself, virtually all XML software also supports XML Namespaces.
- XML Base defines the xml:base attribute, which may be used to set the base for resolution of relative URI references within the scope of a single XML element.
- The XML Information Set or *XML infoset* describes an abstract data model for XML documents in terms of *information items*. The infoset is commonly used in the specifications of XML languages, for convenience in describing constraints on the XML constructs those languages allow.
- xml:id Version 1.0 asserts that an attribute named xml:id functions as an "ID attribute" in the sense used in a DTD.
- XPath defines a syntax named *XPath expressions* which identifies one or more of the internal components (elements, attributes, and so on) included in an XML document. XPath is widely used in other core-XML specifications and in programming libraries for accessing XML-encoded data.
- XSLT is a language with an XML-based syntax that is used to transform XML documents into other XML documents, HTML, or other, unstructured formats such as plain text or RTF. XSLT is very tightly coupled with XPath, which it uses to address components of the input XML document, mainly elements and attributes.
- XSL Formatting Objects, or XSL-FO, is a markup language for XML document formatting which is most often used to generate PDFs.
- XQuery is an XML-oriented query language strongly rooted in XPath and XML Schema. It provides methods to access, manipulate and return XML.
- XML Signature defines syntax and processing rules for creating digital signatures on XML content.
- XML Encryption defines syntax and processing rules for encrypting XML content.

Some other specifications conceived as part of the "XML Core" have failed to find wide adoption, including XInclude, XLink, and XPointer.

Use on the Internet

It is common for XML to be used in interchanging data over the Internet. RFC 3023 gives rules for the construction of Internet Media Types for use when sending XML. It also defines the types "application/xml" and "text/xml", which say only that the data is in XML, and nothing about its semantics. The use of "text/xml" has been criticized as a potential source of encoding problems and is now in the process of being deprecated.[19] RFC 3023 also recommends that XML-based languages be given media types beginning in "application/" and ending in "+xml"; for example "application/svg+xml" for SVG.

Further guidelines for the use of XML in a networked context may be found in RFC 3470 [20], also known as IETF BCP 70; this document is very wide-ranging and covers many aspects of designing and deploying an XML-based language.

Programming interfaces

The design goals of XML include "It shall be easy to write programs which process XML documents."[9] Despite this fact, the XML specification contains almost no information about how programmers might go about doing such processing. The XML Infoset provides a vocabulary to refer to the constructs within an XML document, but once again does not provide any guidance on how to access this information. A variety of APIs for accessing XML have been developed and used, and some have been standardized.

Existing APIs for XML processing tend to fall into these categories:

- Stream-oriented APIs accessible from a programming language, for example SAX and StAX.
- Tree-traversal APIs accessible from a programming language, for example DOM.
- XML data binding, which provides an automated translation between an XML document and programming-language objects.
- Declarative transformation languages such as XSLT and XQuery.

Stream-oriented facilities require less memory and, for certain tasks which are based on a linear traversal of an XML document, are faster and simpler than other alternatives. Tree-traversal and data-binding APIs typically require the use of much more memory, but are often found more convenient for use by programmers; some include declarative retrieval of document components via the use of XPath expressions.

XSLT is designed for declarative description of XML document transformations, and has been widely implemented both in server-side packages and Web browsers. XQuery overlaps XSLT in its functionality, but is designed more for searching of large XML databases.

Simple API for XML (SAX)

SAX is a lexical, event-driven interface in which a document is read serially and its contents are reported as callbacks to various methods on a handler object of the user's design. SAX is fast and efficient to implement, but difficult to use for extracting information at random from the XML, since it tends to burden the application author with keeping track of what part of the document is being processed. It is better suited to situations in which certain types of information are always handled the same way, no matter where they occur in the document.

Pull parsing

Pull parsing[21] treats the document as a series of items which are read in sequence using the Iterator design pattern. This allows for writing of recursive-descent parsers in which the structure of the code performing the parsing mirrors the structure of the XML being parsed, and intermediate parsed results can be used and accessed as local variables within the methods performing the parsing, or passed down (as method parameters) into lower-level methods, or returned (as method return values) to higher-level methods. Examples of pull parsers include StAX in the Java programming language, XMLReader in PHP and System.Xml.XmlReader in the .NET Framework.

A pull parser creates an iterator that sequentially visits the various elements, attributes, and data in an XML document. Code which uses this iterator can test the current item (to tell, for example, whether it is a start or end element, or text), and inspect its attributes (local name, namespace, values of XML attributes, value of text, etc.), and can also move the iterator to the next item. The code can thus extract information from the document as it traverses it. The recursive-descent approach tends to lend itself to keeping data as typed local variables in the code doing the parsing, while SAX, for instance, typically requires a parser to manually maintain intermediate data within a stack of elements which are parent elements of the element being parsed. Pull-parsing code can be more straightforward to understand and maintain than SAX parsing code..

Document Object Model (DOM)

DOM (Document Object Model) is an interface-oriented Application Programming Interface that allows for navigation of the entire document as if it were a tree of "Node" objects representing the document's contents. A DOM document can be created by a parser, or can be generated manually by users (with limitations). Data types in DOM Nodes are abstract; implementations provide their own programming language-specific bindings. DOM implementations tend to be memory intensive, as they generally require the entire document to be loaded into memory and constructed as a tree of objects before access is allowed.

Data binding

Another form of XML processing API is XML data binding, where XML data is made available as a hierarchy of custom, strongly typed classes, in contrast to the generic objects created by a Document Object Model parser. This approach simplifies code development, and in many cases allows problems to be identified at compile time rather than run-time. Example data binding systems include the Java Architecture for XML Binding (JAXB), XML Serialization in .NET,[22] and CodeSynthesis XSD for C++.[23] [24]

XML as data type

XML is beginning to appear as a first-class data type in other languages. The ECMAScript for XML (E4X) extension to the ECMAScript/JavaScript language explicitly defines two specific objects (XML and XMLList) for JavaScript, which support XML document nodes and XML node lists as distinct objects and use a dot-notation specifying parent-child relationships.[25] E4X is supported by the Mozilla 2.5+ browsers and Adobe Actionscript, but has not been adopted more universally. Similar notations are used in Microsoft's LINQ implementation for Microsoft .NET 3.5 and above, and in Scala (which uses the Java VM). The open-source xmlsh application, which provides a Linux-like shell with special features for XML manipulation, similarly treats XML as a data type, using the <[ ]> notation.[26] The Resource Description Framework defines a data type rdf:XMLLiteral to hold wrapped, canonical XML.[27]

History

XML is an application profile of SGML (ISO 8879).[28]

The versatility of SGML for dynamic information display was understood by early digital media publishers in the late 1980s prior to the rise of the Internet.[29] [30] By the mid-1990s some practitioners of SGML had gained experience with the then-new World Wide Web, and believed that SGML offered solutions to some of the problems the Web was likely to face as it grew. Dan Connolly added SGML to the list of W3C's activities when he joined the staff in 1995; work began in mid-1996 when Sun Microsystems engineer Jon Bosak developed a charter and recruited collaborators. Bosak was well connected in the small community of people who had experience both in SGML and the Web.[31]

XML was compiled by a working group of eleven members,[32] supported by an (approximately) 150-member Interest Group. Technical debate took place on the Interest Group mailing list and issues were resolved by consensus or, when that failed, majority vote of the Working Group. A record of design decisions and their rationales was compiled by Michael Sperberg-McQueen on December 4, 1997.[33] James Clark served as Technical Lead of the Working Group, notably contributing the empty-element "<empty/>" syntax and the name "XML". Other names that had been put forward for consideration included "MAGMA" (Minimal Architecture for Generalized Markup Applications), "SLIM" (Structured Language for Internet Markup) and "MGML" (Minimal Generalized Markup Language). The co-editors of the specification were originally Tim Bray and Michael Sperberg-McQueen. Halfway through the project Bray accepted a consulting engagement with Netscape, provoking vociferous protests from Microsoft. Bray was temporarily asked to resign the editorship. This led to intense dispute in the Working Group, eventually solved by the appointment of Microsoft's Jean Paoli as a third co-editor.

The XML Working Group never met face-to-face; the design was accomplished using a combination of email and weekly teleconferences. The major design decisions were reached in twenty weeks of intense work between July and November 1996, when the first Working Draft of an XML specification was published.[34] Further design work continued through 1997, and XML 1.0 became a W3C Recommendation on February 10, 1998.

Sources

XML is a profile of an ISO standard SGML, and most of XML comes from SGML unchanged. From SGML comes the separation of logical and physical structures (elements and entities), the availability of grammar-based validation (DTDs), the separation of data and metadata (elements and attributes), mixed content, the separation of processing from representation (processing instructions), and the default angle-bracket syntax. Removed were the SGML Declaration (XML has a fixed delimiter set and adopts Unicode as the document character set).

Other sources of technology for XML were the Text Encoding Initiative (TEI), which defined a profile of SGML for use as a "transfer syntax"; and HTML, in which elements were synchronous with their resource, document character sets were separate from resource encoding, the xml:lang attribute was invented, and (like HTTP) metadata accompanied the resource rather than being needed at the declaration of a link. The Extended Reference Concrete Syntax (ERCS) project of the SPREAD (Standardization Project Regarding East Asian Documents) project of the ISO-related China/Japan/Korea Document Processing expert group was the basis of XML 1.0's naming rules; SPREAD also introduced hexadecimal numeric character references and the concept of references to make available all Unicode characters. To support ERCS, XML and HTML better, the SGML standard IS 8879 was revised in 1996 and 1998 with WebSGML Adaptations. The XML header followed that of ISO HyTime.

Ideas that developed during discussion which were novel in XML included the algorithm for encoding detection and the encoding header, the processing instruction target, the xml:space attribute, and the new close delimiter for empty-element tags. The notion of well-formedness as opposed to validity (which enables parsing without a schema) was first formalized in XML, although it had been implemented successfully in the Electronic Book Technology "Dynatext" software;[35] the software from the University of Waterloo New Oxford English Dictionary Project; the RISP LISP SGML text processor at Uniscope, Tokyo; the US Army Missile Command IADS hypertext system; Mentor Graphics Context; Interleaf and Xerox Publishing System.

Versions

There are two current versions of XML. The first (*XML 1.0*) was initially defined in 1998. It has undergone minor revisions since then, without being given a new version number, and is currently in its fifth edition, as published on November 26, 2008. It is widely implemented and still recommended for general use.

The second (*XML 1.1*) was initially published on February 4, 2004, the same day as XML 1.0 Third Edition,[36] and is currently in its second edition, as published on August 16, 2006. It contains features (some contentious) that are intended to make XML easier to use in certain cases.[37] The main changes are to enable the use of line-ending characters used on EBCDIC platforms, and the use of scripts and characters absent from Unicode 3.2. XML 1.1 is not very widely implemented and is recommended for use only by those who need its unique features.[38]

Prior to its fifth edition release, XML 1.0 differed from XML 1.1 in having stricter requirements for characters available for use in element and attribute names and unique identifiers: in the first four editions of XML 1.0 the characters were exclusively enumerated using a specific version of the Unicode standard (Unicode 2.0 to Unicode 3.2.) The fifth edition substitutes the mechanism of XML 1.1, which is more future-proof but reduces redundancy. The approach taken in the fifth edition of XML 1.0 and in all editions of XML 1.1 is that only certain characters are forbidden in names, and everything else is allowed, in order to accommodate the use of suitable name characters in future versions of Unicode. In the fifth edition, XML names may contain characters in the Balinese, Cham, or Phoenician scripts among many others which have been added to Unicode since Unicode 3.2.[37]

Almost any Unicode code point can be used in the character data and attribute values of an XML 1.0 or 1.1 document, even if the character corresponding to the code point is not defined in the current version of Unicode. In character data and attribute values, XML 1.1 allows the use of more control characters than XML 1.0, but, for "robustness", most of the control characters introduced in XML 1.1 must be expressed as numeric character references (and #x7F through #x9F, which had been allowed in XML 1.0, are in XML 1.1 even required to be expressed as numeric character references[39]). Among the supported control characters in XML 1.1 are two line break codes that must be treated as whitespace. Whitespace characters are the only control codes that can be written directly.

There has been discussion of an XML 2.0, although no organization has announced plans for work on such a project. XML-SW (SW for skunk works), written by one of the original developers of XML, contains some proposals for what an XML 2.0 might look like: elimination of DTDs from syntax, integration of namespaces, XML Base and XML Information Set (*infoset*) into the base standard.

The World Wide Web Consortium also has an XML Binary Characterization Working Group doing preliminary research into use cases and properties for a binary encoding of the XML infoset. The working group is not chartered to produce any official standards. Since XML is by definition text-based, ITU-T and ISO are using the name *Fast Infoset* for their own binary infoset to avoid confusion (see ITU-T Rec. X.891 | ISO/IEC 24824-1).

Criticism

XML has been criticized for verbosity and complexity, especially when it is used for exchanging highly structured data between applications, which was not its primary design goal.[40] Mapping the data structures of XML to the data types of programming languages such as Java, C#, or Javascript can be difficult. Other criticisms attempt to refute the claim that XML is a self-describing language[41] (though the XML specification itself makes no such claim). The most proposed alternatives include JSON and YAML — both focusing on representing structured data, rather than narrative documents.

References

[1] "XML Media Types, RFC 3023" (http://tools.ietf.org/html/rfc3023#section-3.2). IETF. 2001-01. pp. 9–11. . Retrieved 2010-01-04.
[2] "XML Media Types, RFC 3023" (http://tools.ietf.org/html/rfc3023#section-3.1). IETF. 2001-01. pp. 7–9. . Retrieved 2010-01-04.
[3] M. Murata, D. Kohn, and C. Lilley (2009-09-24). "Internet Drafts: XML Media Types" (http://tools.ietf.org/html/draft-murata-kohn-lilley-xml-03). IETF. . Retrieved 2010-06-10.
[4] http://www.w3.org/TR/2008/REC-xml-20081126/
[5] http://www.w3.org/TR/2006/REC-xml11-20060816/
[6] http://www.w3.org/TR/rec-xml
[7] "XML 1.0 Specification" (http://www.w3.org/TR/REC-xml). W3.org. . Retrieved 2010-08-22.
[8] "W3C DOCUMENT LICENSE" (http://www.w3.org/Consortium/Legal/2002/copyright-documents-20021231). .
[9] "XML 1.0 Origin and Goals" (http://www.w3.org/TR/REC-xml/#sec-origin-goals). . Retrieved July 2009.
[10] "XML Applications and Initiatives" (http://xml.coverpages.org/xmlApplications.html). .
[11] "Introduction to iWork Programming Guide. Mac OS X Reference Library" (http://developer.apple.com/mac/library/documentation/AppleApplications/Conceptual/iWork2-0_XML/Chapter01/Introduction.html). Apple. .
[12] "Extensible Markup Language (XML) 1.0 (Fourth Edition)" (http://www.w3.org/TR/2006/REC-xml-20060816/#charsets). W3.org. . Retrieved 2010-08-22.

[13] "Extensible Markup Language (XML) 1.1 (Second Edition)" (http://www.w3.org/TR/xml11/#charsets). W3.org. . Retrieved 2010-08-22.

[14] "Characters vs. Bytes" (http://www.tbray.org/ongoing/When/200x/2003/04/26/UTF). .

[15] "Autodetection of Character Encodings" (http://www.w3.org/TR/REC-xml/#sec-guessing). .

[16] It is allowed, but not recommended, to use "<" in XML entity values: Extensible Markup Language (XML) 1.0 (Fifth Edition): EntityValue definition (http://www.w3.org/TR/2008/REC-xml-20081126/#NT-AttValue)

[17] "W3C I18N FAQ: HTML, XHTML, XML and Control Codes" (http://www.w3.org/International/questions/qa-controls). .

[18] "Articles tagged with “draconian”" (http://diveintomark.org/tag/draconian). .

[19] M. Murata, D. Kohn, and C. Lilley (2009-09-24). "Internet Drafts: XML Media Types" (http://tools.ietf.org/html/draft-murata-kohn-lilley-xml-03). IETF. . Retrieved 2010-06-10.

[20] http://www.ietf.org/rfc/rfc3470.txt

[21] Push, Pull, Next! (http://www.xml.com/pub/a/2005/07/06/tr.html) by Bob DuCharme, at XML.com

[22] "XML Serialization in the .NET Framework" (http://msdn.microsoft.com/en-us/library/ms950721.aspx). Msdn.microsoft.com. . Retrieved 2009-07-31.

[23] "An Introduction to XML Data Binding in C" (http://www.artima.com/cppsource/xml_data_binding.html). Artima.com. 2007-05-04. . Retrieved 2009-07-31.

[24] "CodeSynthesis XSD — XML Data Binding for C" (http://www.codesynthesis.com/products/xsd/). Codesynthesis.com. . Retrieved 2009-07-31.

[25] "Processing XML with E4X" (https://developer.mozilla.org/en/core_javascript_1.5_guide/processing_xml_with_e4x). *Mozilla Developer Center*. Mozilla Foundation. .

[26] "XML Shell: Core Syntax" (http://www.xmlsh.org/CoreSyntax). xmlsh. 2010-05-13. . Retrieved 2010-08-22.

[27] "Resource Description Framework (RDF): Concepts and Abstract Syntax" (http://www.w3.org/TR/2003/WD-rdf-concepts-20030123/#dfn-rdf-XMLLiteral). W3.org. . Retrieved 2010-08-22.

[28] *ISO/IEC 19757-3*. ISO/IEC. 1 June 2006. p. vi

[29] Bray, Tim (February 2005). "A conversation with Tim Bray: Searching for ways to tame the world's vast stores of information" (http://www.acmqueue.com/modules.php?name=Content&pa=showpage&pid=282). Association for Computing Machinery's "Queue site". . Retrieved April 16, 2006.

[30] edited by Sueann Ambron and Kristina Hooper ; foreword by John Sculley. (1988). "Publishers, multimedia, and interactivity". *Interactive multimedia*. Cobb Group. ISBN 1-55615-124-1.

[31] Eliot Kimber (2006). "XML is 10" (http://drmacros-xml-rants.blogspot.com/#116460437782808906). .

[32] The working group was originally called the "Editorial Review Board." The original members and seven who were added before the first edition was complete, are listed at the end of the first edition of the XML Recommendation, at http://www.w3.org/TR/1998/REC-xml-19980210.

[33] "Reports From the W3C SGML ERB to the SGML WG And from the W3C XML ERB to the XML SIG" (http://www.w3.org/XML/9712-reports.html). W3.org. . Retrieved 2009-07-31.

[34] "Extensible Markup Language (XML)" (http://www.w3.org/TR/WD-xml-961114.html). W3.org. 1996-11-14. . Retrieved 2009-07-31.

[35] Jon Bosak, Sun Microsystems (2006-12-07). "Closing Keynote, XML 2006" (http://2006.xmlconference.org/proceedings/162/presentation.html). 2006.xmlconference.org. . Retrieved 2009-07-31.

[36] "Extensible Markup Language (XML) 1.0 (Third Edition)" (http://www.w3.org/TR/2004/REC-xml-20040204). W3.org. . Retrieved 2010-08-22.

[37] "Extensible Markup Language (XML) 1.1 (Second Edition) – Rationale and list of changes for XML 1.1" (http://www.w3.org/TR/xml11/#dt-name). W3C. . Retrieved 2009-12-11.

[38] Harold, Elliotte Rusty (2004). *Effective XML* (http://www.cafeconleche.org/books/effectivexml/). Addison-Wesley. pp. 10–19. ISBN 0321150406. .

[39] "Extensible Markup Language (XML) 1.1 (Second Edition)" (http://www.w3.org/TR/xml11/#sec-xml11). W3.org. . Retrieved 2010-08-22.

[40] Jeff Atwood (2009): XML: The Angle Bracket Tax (http://www.codinghorror.com/blog/2008/05/xml-the-angle-bracket-tax.html)

[41] Eric Brown (2003): The Myth of Self-Describing XML (http://workflow.healthbase.info/monographs/XML_myths_Browne.pdf)

Further reading

- Annex A of ISO 8879:1986 (SGML)
- Lawrence A. Cunningham (2005). "Language, Deals and Standards: The Future of XML Contracts". *Washington University Law Review*. SSRN 900616 (http://ssrn.com/abstract=900616).
- Bosak, Jon; Tim Bray (May 1999). "XML and the Second-Generation Web". *Scientific American*. Online at XML and the Second-Generation Web (http://www.scientificamerican.com/article.cfm?id=xml-and-the-second-genera).
- Kelly, Sean (February 6, 2006). "Making Mistakes with XML" (http://www.developer.com/xml/article.php/10929_3583081_1). *Developer.com*. Retrieved 2010-10-26.
- Ogbuji, Uche (14 Nov 2006). "Thinking XML: The XML decade" (http://www-128.ibm.com/developerworks/library/x-think38.html). *developerWorks*. IBM. Retrieved 2010-10-26.
- St. Laurent, Simon (February 12, 2003). "Five years later, XML..." (http://www.oreillynet.com/xml/blog/2003/02/five_years_later_xml.html). *O'Reilly XML Blog*. O'Reilly Media. Retrieved 2010-10-26.
- "W3C XML is Ten!" (http://www.w3.org/2008/02/xml10-pressrelease). World Wide Web Consortium. 12 February 2008. Retrieved 2010-10-26.

External links

- W3C XML homepage (http://www.w3.org/XML/)
- XML 1.0 Specification (http://www.w3.org/TR/REC-xml)
- Retrospective on Extended Reference Concrete Syntax (http://xml.ascc.net/en/utf-8/ercsretro.html) by Rick Jelliffe
- XML, Java and the Future of the Web (http://www.xml.com/pub/a/w3j/s3.bosak.html) by Jon Bosak
- XML.gov (http://xml.gov/)
- XML: Ten year anniversary (http://drmacros-xml-rants.blogspot.com/2006/11/xml-ten-year-aniversary.html) by Elliot Kimber
- 23 XML fallacies to watch out for (http://www.itworld.com/xml-fallacies-nlstipsm-080122) by Sean McGrath
- XML Injection (http://projects.webappsec.org/XML-Injection) - Web Application Security Consortium

Article Sources and Contributors

XML *Source*: http://en.wikipedia.org/w/index.php?oldid=410393632 *Contributors*: .:Ajvol:., 207.172.11.xxx, 213.253.39.xxx, 24ten, AHMartin, AThing, Aadaam, Actam, AdamCarden, Adeio, Ahabr, Ahkond, Ahoerstemeier, Aitias, Ajcumming, Aklauss, Aksi great, Alan Liefting, Alansohn, Alexbrn, AlistairMcMillan, Allkeyword, Amakuru, Amire80, AndersFeder, Andrisi, Angeltoribio, Ani td, Ankitasdeveloper, Anna Lincoln, Anon lynx, AnonMoos, Anti stupidity, Anu-43, Aomarks, Apollo1758, Aselaruwan123, Asqueella, Asteiner, Asymmetric, Atanveer9, AzaToth, B4hand, Barek, Barticus88, Bdesham, Beetstra, Belamp, Bernd in Japan, BertSen, Bevo, Bhadani, Biezl, BigFatBuddha, Bissinger, Bje2089, Blinklmc, Bluemoose, BlurTento, Bobdc, Bobianite, Boehm, Bonbayel, Bonethugnd, Booles, BorgQueen, Borgdylan, Boseko, BrianCully, Brick Thrower, Brighterorange, Brion VIBBER, Bryan Derksen, Brz7, Bunnyhop11, Burschik, Businessman332211, Bvajet, C.M.Sperberg-McQueen, CLD, CambridgeBayWeather, Cameltrader, Can't sleep, clown will eat me, CanadianLinuxUser, Caomhin, CapitalSasha, Carewolf, CarlHewitt, Cbdorsett, Cels2, Centrx, Charivari, Chininazu12, ChongDae, Chowbok, Chris 73, Chris Roy, Chrislk02, Chrisnewell, ChristopheS, Chzz, Cipherynx, Clayoquot, ClementSeveillac, CoSort2007, Coconut99 99, Cody5, Colonies Chris, Comesuntbob, Contraverse, Conversion script, CptAnonymous, Crosstowns, Cspan64, Cybercobra, D6, DARTH SIDIOUS 2, DKEdwards, Da monster under your bed, Dan100, DanConnolly, Daniel Olsen, Daniel.Cardenas, DanielVonEhren, DarkFalls, Darkfred, David spector, Davis685, Dcattell, Dcoetzee, DeadEyeArrow, Delcnsltmd, Deodar, Derek Ross, Derekread, Dicklyon, Dickpenn, DigitalEnthusiast, Dingbats, Dino72, Dionysia, Dkrms, Dlohcierekim, Dlrohrer2003, Dolcecars, DominiqueHazaelMassieux, Donmay12, DopefishJustin, DoriSmith, DougBarry, Dpattison2007, Dpbsmith, Dpm64, Dr Headgear, Dreftymac, Dthvt, Dullhunk, Dwheeler, Ebruchez, Edcolins, Edward Z. Yang, Efcavanaugh, Egandrews, Egil, Eisnel, ElBenevolente, Elharo, Elizarf, Ellmist, Elwikipedista, EngineerScotty, Eob, Eranb, Ericjs, Erik Zachte, Erikdw, Eritain, Etu, Evaluist, Ewsers, Fang.zheng, Fantasticfears, FatalError, Feline Hymnic, Ferdinand Pienaar, Figure, Fleminra, FloatingMind, Fnielsen, Folajimi, Fragglet, Fran Rogers, Francl, Frap, Freyr, Frisket, Fsolda, Furrykef, Fvw, GTBacchus, Gaius Cornelius, Gc9580, Gdrori, Geniac, Gennaro Prota, GentlemanGhost, GeoffPurchase, Ghettoblaster, Giftlite, Gjlubbertsen, Gjs238, Glass of water, Glenn, Gogo Dodo, Golwengaud, GrEp, GraemeL, Graham, Graham87, Greg Murray, Ground Zero, Grumpycraig, Gudeldar, Haakon, Hairy Dude, Hannes Hirzel, Harold f, Hashar, Hervegirod, Hicketyhicketyhack, Highwayman65251, Hirzel, Hogman500, Hu12, Hurricane111, Hypertrek, Hyuri, IMSoP, Ian Moody, IanBurrell, Iftikhar88hussaini, Ijmorlan, IlanaDavidi, Imars, Imjustmatthew, Int21h, Intgr, Iridescent, Isilanes, Itai, J.delanoy, JForget, JKing, JLaTondre, JPalonus, JRocketeer, Jackacon, Jacobko, Jacobolus, JakobVoss, JamesBrownJr, Jao, Jargon64, Jauerback, JavaWoman, Jaxad0127, Jaxsam1, Jay, Jeenuv, Jeff G., Jeff3000, Jehzlau, Jerazol, JeremySmyth, Jerome Charles Potts, Jesin, Jhannah, Jibjibjib, Jilplo Haggins, Jimthing, Jmlipton, Joachim Wuttke, Joanjoc, John Vandenberg, JohnSmith777, JohnWhitlock, Johnmarkh, Johnwcowan, Joku, Jonabbey, Jonkerz, Jonnyamazing, Jor, Jpbowen, Jshadias, Jzhang2007, Kai.Klesatschke, Kaldosh, Kamalakannanprogrammer, Kanags, Kapoing, Karderio, Karl Dickman, Katalaveno, Kbrose, Kc2idf, Keithgabryelski, Kenmccallum, Kensall, Kevinconroy, Kgaughan, Kha0sK1d, KickAssClown, Kl4m, Klangenfurt, Klaws, Koavf, Korval, Krauss, Kubigula, Kwiki, Kx1186, LDiracDelta, Lambiam, Larala, Lazynitwit, Lianmei, Liao, Lifefeed, Liftarn, Ligulem, Ling.Nut, LittleDan, Loveenatayal, Lumi71, Lycurgus, M.franceschet, M4gnum0n, MER-C, MK8, MaBoehm, Madir, Mah159, Mak Thorpe, Manishtomar, Maoj-wsu-sp, Mark Renier, MarkSweep, Martijn faassen, Martin451, Martinp23, MartynDavies, Mathmo, Matthäus Wander, MaxEnt, Maximaximax, Maximus06, Mayfare, Mbbradford, Mbell, Mcintyem, Mcorazao, Melab-1, Melon039, Meszigues, Mhkay, Michael Hardy, MichaelJanich, Miguelfms, MikeLynch, Minghong, Mion, Miss Dark, Mjb, Mjpieters, Mola8sses, Montgomery '39, Mp, Mr. Shoeless, Mr.Z-man, MrJones, MrOllie, Mrjmcneil, Ms2ger, Mthibault, Mvulpe, Mwtoews, Mww113, Mxn, NO ACMLM,AND XKEPPER SUCK !, Nannus, Nanshu, Natasha2006, Nations114, NawlinWiki, Neckro, Nemo bis, Netsnipe, Nicmila, Nigelj, Nikkimaria, Nile, Ninly, Niteowlneils, Nivaca, Nixeagle, Noldoaran, Nomediga, Norm mit, Nowa, Nsh, Nwbeeson, Ocaasi, Octane, Ogmios, Ohnoitsjamie, Okyea, OliD, OsamaK, Oscar-ja, Osquar F, OverlordQ, Oxblood, Oyster Flute, P3x984, PTSE, Patrick, Paul Foxworthy, PaulXemdli, Pavel Vozenilek, Paxsimius, Peashy, Pelle, Pengo, PeteVerdon, Peterl, Pgk, Philip Trueman, Phluid61, Phoenix-forgotten, Phyzome, Pianohacker, Pikiwyn, Pmberry, Pnm, Poccil, Porges, Pozcircuitboy, Prakash Nadkarni, Prodoc, Quarl, Quasipalm, Quiddity, Quilokos, Ramesses the Great, Rbonvall, Rbstimers, Rdmsoft, Red660, RedWolf, Redherring, Reinthal, Remy B, RenniePet, Rich Farmbrough, RichMorin, Richalex2010, Rick Block, Rick Jelliffe, RickBeton, Risi, Ritvikbhatnagar1, Rivecoder, Rje, Rjstott, Rjwilmsi, Rklawton, Robert K S, Robert Merkel, Robinjwest, Robomaeyhem, Rodney Boyd, Roger costello, Rory096, RoseParks, Rr2bwreain, Rror, Rvmolen, Ryanrs, Sam Hocevar, SamHathaway, SandiCastle, Sandius, Saqib, Saucepan, Sbvb, Schnolle, Scjessey, Scott MacLean, Scottielad, Sderose, Seanhan, Seidenstud, Semper discens, Sen Mon, ShaneCavanaugh, Shanes, Shibboleth, Shii, Shinkolobwe, Shizhao, Shlomital, SickTwist, Signsofstatic, Simetrical, SivaKumar, Sj, Sjc, Sleepyhead81, Smyth, Sosinfo, Sound effx, Spankman, Spe88, Spudstud, SqueakBox, Stefan.ciobaca, Stephen Gilbert, Steve R Barnes, Stevy76, StewartMH, Stf, Stijn Vermeeren, Stupiddestyredgasd, Stwalkerster, Superm401, Suruena, Suwayya, Svetovid, Syangtar, Sydius, TPK, Tagith, Tags123, Taknik, Talktovalentine, TastyPoutine, Technopilgrim, Teddyb, Terjen, Terrifictriffid, Terrycojones, Thadius856, The Thing That Should Not Be, TheMightyOrb, Thierryc, Think777, Thumperward, Thunderhead, TimBray, TimR, Timc, Timur.shemsedinov, Tobias Bergemann, Todd Vierling, Tony1, ToonArmy, Topbanana, Toussaint, Tpbradbury, Trade2tradewell, Trankin, Traroth, Treekids, Trovatore, Trscavo, Tsunaminoai, Turnstep, TwoOneTwo, Twocs, Typhoonhurricane, Typochimp, Tyrol5, UkPaolo, Uncle Milty, Unforgettableid, Unixxx, Unknown W. Brackets, Vaganyik, Varlaam, Verdy p, Versageek, Vespristiano, Vigilius, Violetriga, Vladkornea, Vojta, Volphy, Vrenator, WSU-AW-AK, Walk Up Trees, Waskage, Wavelength, Wellithy, Wereon, Whale plane, Whkoh, Wickorama, Wiki Leah, Wiki alf, Wiki0709, Wikilibrarian, Wmahan, WojPob, Woohookitty, Wrs1864, Wulfila, Ww, XJamRastafire, Xmltools, Xompanthy, Xpclient, Yaronf, Ygramul, Yonkie, Zhaolei, Zoeb, Zootm, Олександр Кравчук, 1241 anonymous edits

Image Sources, Licenses and Contributors

Image:XML.svg *Source*: http://en.wikipedia.org/w/index.php?title=File:XML.svg *License*: Creative Commons Attribution-Sharealike 2.5 *Contributors*: AutumnSnow, Fryed-peach, JeffyP, Jusjih, Karl Dickman, Latics, Platonides, SKvalen, Soeb, Verdy p, 3 anonymous edits

License