L2/02-343

# Extensible Markup Language (XML) 1.1

## W3C Candidate Recommendation 15 October 2002

**This Version:**
http://www.w3.org/TR/2002/CR-xml11-20021015
**Latest Version:**
http://www.w3.org/TR/xml11/
**Previous Version:**
http://www.w3.org/TR/2002/WD-xml11-20020425/
**Editors:**
John Cowan, Reuters < jcowan@reutershealth.com >



---

## Abstract

This document describes XML 1.1, a deliverable of the XML Core Working Group as defined in the XML Blueberry Requirements. XML 1.1 was formerly known as XML Blueberry. This document takes the form of a series of alterations to the XML 1.0 Recommendation [XML1.0], and its numbered sections correspond to those of the XML 1.0 Recommendation. Sections of that Recommendation that do not appear in this document remain unchanged in XML 1.1.

## Status of this Document

*This section describes the status of this document at the time of its publication. Other documents may supersede this document. The latest status of this document series is maintained at the W3C.*

This specification is being put forth as a W3C Candidate Recommendation of XML 1.1. W3C publishes a technical report as a Candidate Recommendation to indicate that the document is believed to be stable, and to encourage implementation by the developer community. Candidate Recommendation status is described in section 5.2.3 of the Process Document.

The XML Core Working Group (part of the XML Activity, see summary) expects to request that the Director advance this specification to Proposed Recommendation after the XML Core Working Group documents at least two interoperable implementations. The two implementations must be produced by different organizations.

The current implementation report includes implementation feedback we have received to date.

Publication as a Candidate Recommendation does not imply endorsement by the W3C Membership. This is a draft document and may be updated, replaced or obsoleted by other documents at any time. It is inappropriate to cite this document as other than "work in progress."

While this and subsequent drafts of this specification will be written as a series of alterations to the

XML 1.0 Recommendation to facilitate editing and review, it is likely that the final XML 1.1 Recommendation will take the form of an integral revision of the XML 1.0 specification.

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

We explicitly invite comments on this draft. The Candidate Recommendation review period ends at 2359 UTC on 14 February 2003. Comments should be sent to www-xml-blueberry-comments@w3.org . This is the preferred method of providing feedback. Public comments and their responses can be accessed at http://lists.w3.org/Archives/Public/www-xml-blueberry-comments/.

Publication of this document does not imply endorsement by the W3C membership. This is a draft document and may be updated, replaced or obsoleted by other documents at any time. It is inappropriate to cite a W3C Candidate Recommendation as anything other than a "work in progress."A list of current W3C Recommendations and other technical documents can be found at http://www.w3.org/TR/.

---

## Table of Contents



---

# Introduction

The W3C's XML 1.0 Recommendation was first issued in 1998, and despite the issuance of many errata culminating in a Second Edition of 2000, has remained (by intention) unchanged with respect to what is well-formed XML and what is not. This stability has been extremely useful for interoperability. However, the Unicode Standard on which XML 1.0 relies for character specifications has not remained static, evolving from version 2.0 to version 3.1 and beyond. Characters not present in Unicode 2.0 may already be used in XML 1.0 character data. However, they are not allowed in XML names such as element type names, attribute names, enumerated attribute values, processing instruction targets, and so on. In addition, some characters that should have been permitted in XML names were not, due to oversights and inconsistencies in Unicode 2.0.

The overall philosophy of names has changed since XML 1.0. Whereas XML 1.0 provided a rigid definition of names, wherein everything that was not permitted was forbidden, XML 1.1 names are designed so that everything that is not forbidden (for a specific reason) is permitted. Since Unicode will continue to grow past version 3.1, further changes to XML can be avoided by allowing almost any character, including those not yet assigned, in names.

In addition, XML 1.0 attempts to adapt to the line-end conventions of various modern operating

systems, but discriminates against the conventions used on IBM and IBM-compatible mainframes. As a result, XML documents on mainframes are not plain text files according to the local conventions. XML 1.0 documents generated on mainframes must either violate the local line-end conventions, or employ otherwise unnecessary translation phases before parsing and after generation. Allowing straightforward interoperability is particularly important when data stores are shared between mainframe and non-mainframe systems (as opposed to being copied from one to the other). Therefore XML 1.1 adds NEL (#x85) to the list of line-end characters. For completeness, the Unicode line separator character, #x2028, is also supported.

Finally, there is considerable demand to define a standard representation of arbitrary Unicode characters in XML documents. Therefore, XML 1.1 allows the use of character references to the control characters #x1 through #x1F, most of which are forbidden in XML 1.0. For reasons of robustness, however, these characters still cannot be used directly in documents. In order to improve the robustness of character encoding detection, the additional control characters #x7F through #x9F, which were freely allowed in XML 1.0 documents, now must also appear only as character references. (Whitespace characters are of course exempt.) The minor sacrifice of backward compatibility is considered not significant. Due to potential problems with APIs, #x0 is still forbidden both directly and as a character reference.

A new XML version, rather than a set of errata to XML 1.0, is being created because the changes affect the definition of well-formed documents. XML 1.0 processors must continue to reject documents that contain new characters in XML names, new line-end conventions, and references to control characters. The distinction between XML 1.0 and XML 1.1 documents will be indicated by the version number information in the XML declaration at the start of each document.

## 2.2 Characters

*Change production [2]:*

```
[2]     Char    ::=    #x9 | #xA | #xD | [#x20-#x7E] | #x85 | [#xA0-#xD7FF]
                     | [#xE000-#xFFFD] | [#x10000-#x10FFFF]
```

*Change the associated comment to read:*

any Unicode character, excluding most ISO controls, the surrogate blocks, FFFE, and FFFF

## 2.3 Common Syntactic Constructs

*Change production [4], and add new production [4a]:*

```
[4]     NameStartChar := ":" | [A-Z] | "_" | [a-z] |
        [#xC0-#x2FF] | [#x370-#x37D] | [#x37F-#x1FFF] |
        [#x200C-#x200D] | [#x2070-#x218F] | [#x2C00-#x2FEF] |
        [#x3001-#xD7FF] | [#xF900-#xEFFFF]

[4a]    NameChar := NameStartChar | "-" | "." | [0-9] | #xB7 |
        [#x0300-#x036F] | [#x203F-#x2040]
```

*Change production [5] to:*

```
[5]     Name    ::=   NameStartChar NameChar*
```

*Insert the following three paragraphs after production 5:*

The first character of a Name must be a NameStartChar, but any other characters must be NameChars; this mechanism is used to prevent names from beginning with Latin (ASCII) digits or with basic combining characters. Almost all characters are permitted in names, except those which either are or reasonably could be used as delimiters. The intention is to be inclusive rather than exclusive, so that writing systems not yet encoded in Unicode can be used in XML names. See

Appendix B for suggestions on the creation of names.

Document authors are encouraged to use names which are meaningful words or combinations of words in natural languages, and to avoid symbolic or whitespace characters in names. Note that COLON, HYPHEN-MINUS, FULL STOP (period), LOW LINE (underscore), and MIDDLE DOT are explicitly permitted.

The ASCII symbols and punctuation marks, along with a fairly large group of Unicode symbol characters, are excluded from names because they are more useful as delimiters in contexts where XML names are used outside XML documents; providing this group gives those contexts hard guarantees about what *cannot* be part of an XML name. The character #x037E, GREEK QUESTION MARK, is excluded because when normalized it becomes a semicolon, which could change the meaning of entity references.

*Change production [7] to:*

```
[7]     Nmtoken    ::=   NameChar+
```

## 2.8 Prolog and Document Type Declaration

*Change "1.0" everywhere to "1.1"*

*Add the following paragraph:*

XML 1.1 processors should accept XML 1.0 documents as well. If a document is well-formed or valid XML 1.0, and provided it does not contain any characters in the range [#x7F-#x9F] other than as character escapes, it may be made well-formed or valid XML 1.1 respectively simply by changing the version number.

## 2.11 End-of-Line Handling

*Replace the second paragraph with:*

To simplify the tasks of applications, the characters passed to an application by the XML processor must be as if the XML processor normalized all line breaks in external parsed entities (including the document entity) on input, before parsing, by translating all of the following to a single #xA character:

- the two-character sequence #xD #xA
- the two-character sequence #xD #x85
- the single character #x85
- the single character #x2028
- any #xD character that is not immediately followed by #xA or #x85.

## 2.13 Normalization Checking [NEW]

All XML parsed entities (including document entities) should be fully normalized as per the definition of [Charmod] supplemented by the following definitions of *relevant constructs* for XML:

- The replacement text of all parsed entities
- All text matching, in context, one of the following productions:
    - CData
    - CharData
    - content

- Name
- Nmtoken

However, a document is still well-formed even if it is not fully normalized. XML processors should provide a user option to verify that the document being processed is in fully normalized form, and report to the application whether it is or not. The option to not verify should be chosen only when the input text is certified, as defined by [Charmod].

The verification of full normalization must be carried out as if by first verifying that the entity is in include-normalized form as defined by [Charmod] and by then verifying that none of the relevant constructs listed above begins (after character references are expanded) with a composing character as defined by [Charmod]. Non-validating processors must ignore possible denormalizations that would be caused by inclusion of external entities that they do not read.

**Note:** The composing characters are all Unicode characters of non-zero combining class, plus a small number of class-zero characters that nevertheless take part as a non-initial character in certain Unicode canonical decompositions. Since these characters are meant to follow base characters, restricting relevant constructs (including content) from beginning with a composing character does not meaningfully diminish the expressiveness of XML.

If, while verifying full normalization, a processor encounters characters for which it cannot determine the normalization properties (i.e., characters introduced in a version of [Unicode] later than the one used in the implementation of the processor), then the processor may, at user option, ignore any possible denormalizations caused by these characters. The option to ignore those denormalizations should not be chosen by applications when reliability or security are critical.

XML processors must not transform the input to be in fully normalized form. XML applications that create XML 1.1 output from either XML 1.1 or XML 1.0 input should ensure that the output is fully normalized; it is not necessary for internal processing forms to be fully normalized.

The purpose of this section is to strongly encourage XML processors to ensure that the creators of XML documents have properly normalized them, so that XML applications can make tests such as identity comparisons of strings without having to worry about the different possible "spellings" of strings which Unicode allows.

# 4.1 Character and Entity References

*Change the **Well-formedness constraint: Legal Character** to read:*

Characters referred to using character references must either match the production for Char, or be one of the ISO control characters in the ranges [#x1-#x1F] or [#x7F-#x9F].

# 4.3.4 Version Information in Entities [NEW]

Each entity, including the document entity, can be separately declared as XML 1.0 or XML 1.1. The version declaration appearing in the document entity determines the version of the document as a whole. An XML 1.1 document may invoke XML 1.0 external entities, so that otherwise duplicated versions of external entities, particularly DTD external subsets, need not be maintained. However, in such a case the rules of XML 1.1 are applied to the entire document.

If an entity (including the document entity) is not labeled with a version number, it is treated as if labeled as version 1.0.

# Appendix A References

*Add the following normative references:*

**[XML1.0]**
Tim Bray, Jean Paoli, C.M. Sperberg-McQueen, Eve Maler (editors), Extensible Markup Language (XML) 1.0 (Second Edition), 6 October 2000. (See http://www.w3.org/TR/REC-xml .)

**[Charmod]**
Martin J. Dürst, François Yergeau, Richard Ishida, Misha Wolf, Asmus Freytag, Tex Texin Character Model for the World Wide Web, W3C Working Draft, 30 April 2002. (See http://www.w3.org/TR/charmod/.)

# Appendix B Suggestions for XML Names (Non-Normative)

*Appendix B is to be changed from a normative appendix called "Character Classes" to a non-normative one called "Suggestions for XML Names", with the following content.*

The following suggestions define what is believed to be best practice in the construction of XML names used as element names, attribute names, processing instruction targets, entity names, notation names, and the values of attributes of type ID, and are intended as guidance for document authors and schema designers. All references to Unicode are understood with respect to a particular version of the Unicode Standard greater than or equal to 3.0; which version should be used is left to the discretion of the document author or schema designer.

The first two suggestions are directly derived from the rules given for identifiers in the Unicode Standard, version 3.0, and exclude all control characters, enclosing nonspacing marks, non-decimal numbers, private-use characters, punctuation characters (with the noted exceptions), symbol characters, unassigned codepoints, and whitespace characters. The other suggestions are mostly derived from XML 1.0 Appendix B.

- The first character of any name should have a Unicode General Category of Ll, Lu, Lo, Lm, Lt, or Nl, or else be '_' #x5F.
- Characters other than the first should have a Unicode General Category of Ll, Lu, Lo, Lm, Lt, Mc, Mn, Nl, Nd, Pc, or Cf, or else be one of the following: '-' #x2D, '.' #x2E, ':' #x3A or '·' #xB7 (middle dot). Since Cf characters are not directly visible, they should be employed with caution and only when necessary, to avoid creating names which are distinct to XML processors but look the same to human beings.
- Ideographic characters which have a canonical decomposition (including those in the ranges [#xF900-#xFAFF] and [#x2F800-#x2FFFD], with 12 exceptions) should not be used in names.
- Characters which have a compatibility decomposition (those with a "compatibility formatting tag" in field 5 of the Unicode Character Database -- marked by field 5 beginning with a "<") should not be used in names. This suggestion does not apply to #x0E33 THAI CHARACTER SARA AM or #x0EB3 LAO CHARACTER AM, which despite their compatibility decompositions are in regular use in those scripts.
- Combining characters meant for use with symbols only (including those in the ranges [#x20D0-#x20EF] and [#x1D165-#x1D1AD]) should not be used in names.
- The interlinear annotation characters ([#xFFF9-#xFFFB) should not be used in names.
- Variation selector characters should not be used in names.
- Names which are nonsensical, unpronounceable, hard to read, or easily confusable with other names should not be employed.