



Associating Style Sheets with XML documents

1.0 (Second Edition)

W3C Recommendation 28 October 2010

This version:

Latest version:

Previous version:

Authors and Contributors:

James Clark (First Edition) <jjc@jclark.com>

Simon Pieters (Second Edition) (Opera Software) <simonp@opera.com>

Henry S. Thompson (Second Edition) (University of Edinburgh) <ht@inf.ed.ac.uk>

Copyright © 2010 W3C[®] (MIT, INRIA, Keio), All Rights Reserved.
W3C [liability](#), [trademark](#), [document use](#), and [software licensing](#) rules apply.

Abstract

This document allows style sheets to be associated with an XML document by including one or more processing instructions with a target of `xml-stylesheet` in the document's prolog.

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](http://www.w3.org/TR/) at <http://www.w3.org/TR/>.

This document is a product of the [XML Core Working Group](#) as part 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-stylesheet>.

This second edition incorporates all known errata as of the publication date, clarifies several areas left unspecified in the earlier edition, and has been restructured to allow other specifications to reuse the rules for parsing pseudo-attributes from a string. A more detailed description of the changes from the first edition is in [Appendix B – Changes since the first edition](#) on page 6. This edition supersedes the previous [edition of 29 June 1999](#).

Comments on this document may be sent to www-xml-stylesheet-comments@w3.org; public [archives](#) are available. The errata list for this document is available at <http://www.w3.org/1999/06/REC-xml-stylesheet-19990629/errata>

This document has been reviewed by W3C Members, by software developers, and by other W3C groups and interested parties, and is endorsed by the Director as a W3C Recommendation. It is a stable document and may be used as reference material or cited 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.

W3C maintains a [public list of any patent disclosures](#) made in connection with the deliverables of the group; that page also includes instructions for disclosing a patent. An individual who has actual knowledge of a patent which the individual believes contains [Essential Claim\(s\)](#) must disclose the information in accordance with [section 6 of the W3C Patent Policy](#).

Table of Contents

1. Introduction	1
2. Conformance requirements	1
3. Pseudo-attributes	2
4. The xml-stylesheet processing instruction	3

Appendices

A. References	5
B. Changes since the first edition	6

This page is intentionally left blank.

1. Introduction

(This section is non-normative.)

Authors might have particular intentions as to how user agents are to present the information contained in their XML documents. This specification provides a non-intrusive mechanism, using a processing instruction, to provide links to one or more style sheets, i.e. resources specifying the desired rendering in a designated language. User agents will use these resources to control presentation of XML.

Consider an XHTML document with style sheet associations using the `link` element:

```
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title>Example with link elements</title>
    <link rel="stylesheet" href="common.css"/>
    <link rel="stylesheet" href="default.css" title="Default style"/>
    <link rel="alternate stylesheet" href="alt.css" title="Alternative style"/>
    <link rel="stylesheet" href="single-col.css" media="all and (max-width: 30em)"/>
  </head>
  <body>
    ...
  </body>
</html>
```

This document could be written as follows, using [xml-stylesheet processing instructions](#):

```
<?xml-stylesheet href="common.css"?>
<?xml-stylesheet href="default.css" title="Default style"?>
<?xml-stylesheet alternate="yes" href="alt.css" title="Alternative style"?>
<?xml-stylesheet href="single-col.css" media="all and (max-width: 30em)"?>
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title>Example with xml-stylesheet processing instructions</title>
  </head>
  <body>
    ...
  </body>
</html>
```

2. Conformance requirements

All diagrams, examples, and notes in this specification are non-normative, as are all sections explicitly marked non-normative. Everything else in this specification is normative.

The key words *must*, *must not*, *should*, *should not* and *may* in the normative parts of this document are to be interpreted as described in RFC 2119. These words do not appear in all uppercase letters in this specification. [RFC2119]

The following conformance classes are defined by this specification:

Documents

A document is considered to be a conforming document if it satisfies all must-level criteria in this specification that apply to documents.

[xml-stylesheet processing instructions](#) and their pseudo-attributes are intended for use by document authors to provide links to and information about one or more stylesheets, as described in § 4 – [The xml-stylesheet processing instruction](#) on page 3, which are available to be applied to the containing document.

xml-stylesheet processors

XML defines an application as a software module which receives the information content of an XML document from an XML processor. An *xml-stylesheet processor* is such an application which processes XML processing instructions in accordance with this specification. [XML]

An *xml-stylesheet processor* is considered to be a conforming *xml-stylesheet processor* if it satisfies all must-level criteria in this specification that apply to *xml-stylesheet processors*. *xml-stylesheet processors* do not have to check or enforce any of the constraints on documents.

Applications

An *xml-stylesheet processor* may be part of a larger XML application, or may function independently. In either case, an *application* is the consumer of the pseudo-attribute analysis defined in this specification.

A detailed specification of how *applications* exploit the information contained in *xml-stylesheet processing instructions* is out of scope for this document, as this may reasonably vary from application to application. Accordingly this specification places no conformance requirements on applications.

It is however expected that specifications governing the use of stylesheet languages will reference this specification in describing how applications may identify stylesheets for use. Such specifications are expected to specify interpretations for the values (or absence of value) of those *pseudo-attributes* which they make use of, doing so in a way that is as consistent as possible with those *pseudo-attributes'* descriptions in § 4 – [The xml-stylesheet processing instruction](#) on page 3, insofar as this makes sense for the application and stylesheet language(s) in question.

This specification is defined with reference to the vocabulary for XML provided by the XML Information Set. [INFOSET]

The productions in this specification use the same notation as used in the XML specification. Tokens in the grammar that are not defined in this specification are defined in the XML specification. [XML]

3. Pseudo-attributes

The *rules for parsing pseudo-attributes from a string* are given in this section.

The *parsing result* is either a set of *pseudo-attributes* or an error.

If the given string is not matched by the **PseudoAtts** production, the *parsing result* is an error.

Productions for strings with pseudo-attributes

[1] `PseudoAtts ::= PseudoAtt? (S PseudoAtt)* S?`

- [2] PseudoAtt ::= [Name](#) S? "=" S? **PseudoAttValue**
- [3] PseudoAttValue ::= (" ([^"<&] | [CharRef](#) | **PredefEntityRef**)* "" | "" ([^"<&] | [CharRef](#) | **PredefEntityRef**)* "")
- [4] PredefEntityRef ::= "&"; "<"; ">"; """; "'";

Each string matched by **PseudoAtt** in the **PseudoAtts** production represents a *pseudo-attribute*. A *pseudo-attribute* has a [name](#) and a [value](#).

The string matched by [Name](#) in the **PseudoAtt** production constitutes the *name* of the corresponding *pseudo-attribute*.

The string matched by **PseudoAttValue** in the **PseudoAtt** production — with the three modifications specified below — constitutes the *value* of the corresponding *pseudo-attribute*.

1. Each [CharRef](#) is replaced with the character it represents according to XML [[XML](#)].
2. Each **PredefEntityRef** in **PseudoAttValue** is replaced with U+0026 (&) if it is "&"; U+003C (<) if it is "<"; U+003E (>) if it is ">"; U+0022 (") if it is """; and U+0027 (') if it is "'".
3. The first and last character (the start and end quotes) are removed.

The [parsing result](#) is an error if the XML Legal Character well-formedness constraint is violated for any [CharRef](#). [[XML](#)]

The [parsing result](#) is an error if there are more than one *pseudo-attribute* with the same [name](#).

If the [parsing result](#) is not an error, then it is the set of *pseudo-attributes* represented when the given string is matched by the **PseudoAtts** production.

4. The xml-stylesheet processing instruction

A processing instruction information item is said to be a *potential xml-stylesheet processing instruction* if it has the [target] property `xml-stylesheet` and it either is in the [children] property of a document information item and appears before the element information item of the document information item's [children] property, or it appears in the [children] property of a document type declaration information item.

For [potential xml-stylesheet processing instructions](#) that are in the [children] property of a document information item, [xml-stylesheet processors](#) must report to the [application](#) the [parsing result](#) of invoking the [rules for parsing pseudo-attributes from a string](#), using the processing instruction information item's [content] property as the string.

For [potential xml-stylesheet processing instructions](#) that are in the [children] property of a document type declaration information item, [xml-stylesheet processors](#) may report to the [application](#) the [parsing result](#) of invoking the [rules for parsing pseudo-attributes from a string](#), using the processing instruction information item's [content] property as the string. If it is not reported to the [application](#), the processing instruction information item is said to be *ignored*.

 Since non-validating XML processors are not required to read parameter entities or the external subset, it is possible that processing instructions that appear in parameter entities or the external subset will not be present in the document type declaration information item's [children] property. [[XML](#)]

A [potential `xml-stylesheet` processing instruction](#) is said to be an *xml-stylesheet processing instruction* if the [parsing result](#) is not an error when invoking the [rules for parsing pseudo-attributes from a string](#), using the processing instruction information item's `[content]` property as the string.

Documents must not use processing instruction information items with the `[target]` property `xml-stylesheet` if they are not [xml-stylesheet processing instructions](#).

Documents should not use [xml-stylesheet processing instructions](#) that are in the `[children]` property of a document type declaration information item.

An [xml-stylesheet processing instruction](#) will match the following production:

Production for `xml-stylesheet` processing instruction

```
[5]      StyleSheetPI ::= "<?xml-stylesheet" ( (S PseudoAtts)? - (Char* "?>" Char*) ) ">"
```

Documents may specify the following [pseudo-attributes](#) on `xml-stylesheet` processing instructions, unless otherwise stated:

href

Gives the address of the referenced style sheet. Documents must specify this [pseudo-attribute](#). Documents must set the [value](#) to a string that matches the grammar for `<IRI-reference>` given in RFC 3987. [RFC3987]



It is possible to associate a style sheet that is in the same document as the [xml-stylesheet processing instruction](#) by using just a fragment identifier in the [pseudo-attribute](#).

For details on how this is managed in XSLT, see [3.11 Embedded Stylesheet Modules](#) in [XSLT].

type

Gives an advisory media type for the referenced style sheet. If specified, documents must set the [value](#) to a string that matches the `media-type` production defined in section 3.7 "Media Types" of RFC 2616. [RFC2616]



Syntactic conformance to the production in RFC 2616 is all that is required by the above. This specification imposes no requirements with respect to the status of the types used. Some unregistered (as of the publication of this specification) media types, such as `text/xsl`, are well-supported by existing implementations, whereas some registered media types, such as `application/xslt+xml`, are not.

The [value](#) is advisory in that it is intended to be used by an application only when no other source of media type information becomes available during retrieval of the stylesheet itself.

title

Gives the title of the referenced style sheet in a style sheet set. No constraints beyond those of the [PseudoAttValue](#) production are placed on the [value](#).

media

Gives the media for which the referenced style sheet applies. If specified, documents must set the [value](#) to a string that matches the `media_query_list` production of the Media Queries specification. [MQ]

charset

Gives an advisory character encoding for the referenced style sheet. If specified, documents must set the **value** to a valid character encoding name, which must be the name or alias labeled as "preferred MIME name" in the IANA Character Sets registry, if there is one, or the encoding's name, if none of the aliases are so labeled. [[IANACHARSET](#)]

The **value** is advisory in that it is intended to be used by an application only when no other source of encoding information becomes available during retrieval of the stylesheet itself, either explicitly via an HTTP header, or implicitly via its media type, as in the case of XML documents.

alternate

If the **value** is "yes", it indicates that the referenced style sheet is an alternative style sheet, and documents must also specify the **pseudo-attribute** with a non-empty **value**. If specified, documents must set the **value** to either "yes" or "no".

Documents must not specify other **pseudo-attributes** on [xml-stylesheet processing instructions](#).

Any links to style sheets that are specified externally to the document (e.g. Link headers in some versions of HTTP [[RFC2068](#)]) are considered to create associations that occur before the associations specified by the [xml-stylesheet processing instructions](#). The **application** is responsible for taking all associations and determining how, if at all, their order affects its processing.

Appendix A. References

References are normative unless marked "non-normative".

IANACHARSET

[Character Sets. IANA, May 2007](#). Available at <http://www.iana.org/assignments/character-sets>.

INFOSET

[XML Information Set, J. Cowan, R. Tobin. W3C, February 2004](#). Available at <http://www.w3.org/TR/xml-infoset/>.

MQ

[Media Queries, H. Lie, T. Çelik, D. Glazman, A. van Kesteren. W3C, April 2009](#). Available at <http://www.w3.org/TR/css3-mediaqueries/>.

RFC2068

[\(non-normative\) Hypertext Transfer Protocol — HTTP/1.1, R. Fielding, J. Gettys, J. Mogul, H. Frystyk, T. Berners-Lee. IETF, June 1997](#). Available at <http://www.ietf.org/rfc/rfc2068.txt>.

RFC2119

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

RFC2616

[Hypertext Transfer Protocol — HTTP/1.1](#), R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, T. Berners-Lee. IETF, June 1999. Available at <http://www.ietf.org/rfc/rfc2616.txt>.

RFC3987

[Internationalized Resource Identifiers \(IRIs\)](#), M. Dürst, M. Suignard. IETF, January 2005. Available at <http://www.ietf.org/rfc/rfc3987.txt>.

XML

[Extensible Markup Language](#), T. Bray, J. Paoli, C. Sperberg-McQueen, E. Maler, F. Yergeau. W3C, November 2008. Available at <http://www.w3.org/TR/xml/>.

XSLT

[\(non-normative\) XSL Transformations \(XSLT\) Version 2.0](#), M. Kay. W3C, January 2007. Available at <http://www.w3.org/TR/xslt20/>.

Appendix B. Changes since the first edition

This appendix is non-normative

The first edition of this specification was admirably brief, but at the same time left many details unstated. This second edition aims to fill the gaps left in the first edition, while restructuring the presentation to provide for independent citation and re-use of the pseudo-attribute construct.

Neither the syntax nor the semantics of the `xml-stylesheet` processing instructions have been changed, beyond the restructuring just mentioned. Conformance constraints on processors have been eased slightly, in that `xml-stylesheet` processing instructions occurring within the internal or external subset may be ignored.

The following list identifies the major changes which have been made:

- Provided definitions for a number of terms used but not defined in the first edition;
- Added a conformance section, distinguishing between processor and document conformance, all of which was implicit in the first edition;
- Identified a number of error cases, which were implicit in the first edition's appeal to the parallel with element start tag processing, and specified expected processor behaviour;
- In recognition of deployed processor behaviour, allowed `xml-stylesheet` processing instructions to be ignored unless they are among the [children] of the document information item;
- Added a number of references, but removed the explicit dependence on the HTML 4.0 specification by adding descriptions of the meanings of each of the pseudo-attributes consistent with their HTML 4.0 use but brought up-to-date;
- Removed the (non-normative) Rationale section, as it contained a number of out-of-date assumptions;
- Made the `type` pseudo-attribute optional, as agreed by [existing erratum](#).