



Associating Style Sheets with XML documents

Version 1.0

W3C Recommendation 29 June 1999

This version:

<http://www.w3.org/1999/06/REC-xml-stylesheet-19990629>

Latest version:

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

Previous version:

<http://www.w3.org/TR/1999/xml-stylesheet-19990428>

Editor:

James Clark <jjc@jclark.com>

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

Abstract

This document allows a style sheet 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 document has been reviewed by W3C Members and other interested parties and has been endorsed by the Director as a W3C [Recommendation](#). It is a stable document and may be used as reference material or cited as a normative reference from other documents. 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 list of known errors in this specifications is available at <http://www.w3.org/TR/1999/xml-stylesheet-19990629/errata>.

Comments on this specification may be sent to <www-xml-stylesheet-comments@w3.org>. The archive of public comments is available at <http://w3.org/Archives/Public/www-xml-stylesheet-comments>.

A list of current W3C Recommendations and other technical documents can be found at <http://www.w3.org/TR>.

The Working Group expects additional mechanisms for linking style sheets to XML document to be defined in a future specification.

The use of XML processing instructions in this specification should not be taken as a precedent. The W3C does not anticipate recommending the use of processing instructions in any future specification. The [Rationale](#) explains why they were used in this specification.

This document was produced as part of the [W3C XML Activity](#).

Table of contents

1 [The xml-stylesheet processing instruction](#)

Appendices

A [References](#)

B [Rationale](#)

1 The `xml-stylesheet` processing instruction

Style Sheets can be associated with an XML [\[XML10\]](#) document by using a processing instruction whose target is `xml-stylesheet`. This processing instruction follows the behaviour of the HTML 4.0 `<LINK REL="stylesheet">` [\[HTML40\]](#).

The `xml-stylesheet` processing instruction is parsed in the same way as a start-tag, with the exception that entities other than predefined entities must not be referenced.

The following grammar is given using the same notation as the grammar in the XML Recommendation [\[XML10\]](#). Symbols in the grammar that are not defined here are defined in the XML Recommendation.

xml-stylesheet processing instruction

- [1] `StyleSheetPI` ::= `'<?xml-stylesheet' (S PseudoAtt)* S? '?>'`
- [2] `PseudoAtt` ::= `Name S? '=' S? PseudoAttValue`
- [3] `PseudoAttValue` ::= `('"' ([^"&] | CharRef | PredefEntityRef)* '"'`
`| "'" ([^'&] | CharRef | PredefEntityRef)* "'"`
`- (Char* '?>' Char*)`
- [4] `PredefEntityRef` ::= `'&' | '<' | '>' | '"' | '''`

In [PseudoAttValue](#), a [CharRef](#) or a [PredefEntityRef](#) is interpreted in the same manner as in a normal XML attribute value. The actual value of the pseudo-attribute is the value after each reference is replaced by the character it references. This replacement is not performed automatically by an XML processor.

The `xml-stylesheet` processing instruction is allowed only in the prolog of an XML document. The syntax of XML constrains where processing instructions are allowed in the prolog; the `xml-stylesheet` processing instruction is allowed anywhere in the prolog that meets these constraints.

NOTE: If the `xml-stylesheet` processing instruction occurs in the external

DTD subset or in a parameter entity, it is possible that it may not be processed by a non-validating XML processor (see [\[XML10\]](#)).

The following pseudo attributes are defined

```
href CDATA #REQUIRED
type CDATA #REQUIRED
title CDATA #IMPLIED
media CDATA #IMPLIED
charset CDATA #IMPLIED
alternate (yes|no) "no"
```

The semantics of the pseudo-attributes are exactly as with `<LINK REL="stylesheet">` in HTML 4.0, with the exception of the `alternate` pseudo-attribute. If `alternate="yes"` is specified, then the processing instruction has the semantics of `<LINK REL="alternate stylesheet">` instead of `<LINK REL="stylesheet">`.

NOTE: Since the value of the `href` attribute is a URI reference, it may be a relative URI and it may contain a fragment identifier. In particular the URI reference may contain only a fragment identifier. Such a URI reference is a reference to a part of the document containing the `xml-stylesheet` processing instruction (see [\[RFC2396\]](#)). The consequence is that the `xml-stylesheet` processing instruction allows style sheets to be embedded in the same document as the `xml-stylesheet` processing instruction.

In some cases, style sheets may be linked with an XML document by means external to the document. For example, earlier versions of HTTP [\[RFC2068\]](#) (section 19.6.2.4) allowed style sheets to be associated with XML documents by means of the `Link` header. Any links to style sheets that are specified externally to the document are considered to occur before the links specified by the `xml-stylesheet` processing instructions. This is the same as in HTML 4.0 (see [section 14.6](#)).

Here are some examples from HTML 4.0 with the corresponding processing instruction:

```
<LINK href="mystyle.css" rel="style sheet" type="text/css">
<?xml-stylesheet href="mystyle.css" type="text/css"?>

<LINK href="mystyle.css" title="Compact" rel="stylesheet"
type="text/css">
<?xml-stylesheet href="mystyle.css" title="Compact" type="text/css"?>

<LINK href="mystyle.css" title="Medium" rel="alternate stylesheet"
type="text/css">
<?xml-stylesheet alternate="yes" href="mystyle.css" title="Medium"
type="text/css"?>
```

Multiple `xml-stylesheet` processing instructions are also allowed with exactly the same semantics as with `LINK REL="stylesheet"`. For example,

```
<LINK rel="alternate stylesheet" title="compact" href="small-base.css"
type="text/css">
<LINK rel="alternate stylesheet" title="compact" href="small-extras.css"
type="text/css">
<LINK rel="alternate stylesheet" title="big print" href="bigprint.css"
type="text/css">
<LINK rel="stylesheet" href="common.css" type="text/css">
```

would be equivalent to:

```
<?xml-stylesheet alternate="yes" title="compact" href="small-base.css"
```

```
type="text/css"?>
<?xml-stylesheet alternate="yes" title="compact" href="small-extras.css"
type="text/css"?>
<?xml-stylesheet alternate="yes" title="big print" href="bigprint.css"
type="text/css"?>
<?xml-stylesheet href="common.css" type="text/css"?>
```

A References

HTML40

World Wide Web Consortium. *HTML 4.0 Specification*. W3C Recommendation.
See <http://www.w3.org/TR/REC-html40>

RFC2068

R. Fielding, J. Gettys, J. Mogul, H. Frystyk Nielsen, and T. Berners-Lee.
Hypertext Transfer Protocol -- HTTP/1.1. IETF RFC 2068. See
<http://www.ietf.org/rfc/rfc2068.txt>.

RFC2396

T. Berners-Lee, R. Fielding, and L. Masinter. *Uniform Resource Identifiers (URI):
Generic Syntax*. IETF RFC 2396. See <http://www.ietf.org/rfc/rfc2396.txt>.

XML10

World Wide Web Consortium. *Extensible Markup Language (XML) 1.0*. W3C
Recommendation. See <http://www.w3.org/TR/1998/REC-xml-19980210>

B Rationale

There was an urgent requirement for a specification for style sheet linking that could be completed in time for the next release from major browser vendors. Only by choosing a simple mechanism closely based on a proven existing mechanism could the specification be completed in time to meet this requirement.

Use of a processing instruction avoids polluting the main document structure with application specific processing information.

The mechanism chosen for this version of the specification is not a constraint on the additional mechanisms planned for future versions. There is no expectation that these will use processing instructions; indeed they may not include the linking information in the source document.