[MS-XSLT]:

Microsoft XSLTransformations (XSLT) Standards Support Document

Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation ("this documentation") for protocols, file formats, data portability, computer languages, and standards support. Additionally, overview documents cover inter-protocol relationships and interactions.
- **Copyrights**. This documentation is covered by Microsoft copyrights. Regardless of any other terms that are contained in the terms of use for the Microsoft website that hosts this documentation, you can make copies of it in order to develop implementations of the technologies that are described in this documentation and can distribute portions of it in your implementations that use these technologies or in your documentation as necessary to properly document the implementation. You can also distribute in your implementation, with or without modification, any schemas, IDLs, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications documentation.
- **No Trade Secrets**. Microsoft does not claim any trade secret rights in this documentation.
- **Patents**. Microsoft has patents that might cover your implementations of the technologies described in the Open Specifications documentation. Neither this notice nor Microsoft's delivery of this documentation grants any licenses under those patents or any other Microsoft patents. However, a given Open Specifications document might be covered by the Microsoft Open Specifications Promise or the Microsoft Community Promise. If you would prefer a written license, or if the technologies described in this documentation are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting iplg@microsoft.com.
- **License Programs**. To see all of the protocols in scope under a specific license program and the associated patents, visit the Patent Map.
- **Trademarks**. The names of companies and products contained in this documentation might be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights. For a list of Microsoft trademarks, visit www.microsoft.com/trademarks.
- **Fictitious Names**. The example companies, organizations, products, domain names, email addresses, logos, people, places, and events that are depicted in this documentation are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

Reservation of Rights. All other rights are reserved, and this notice does not grant any rights other than as specifically described above, whether by implication, estoppel, or otherwise.

Tools. The Open Specifications documentation does not require the use of Microsoft programming tools or programming environments in order for you to develop an implementation. If you have access to Microsoft programming tools and environments, you are free to take advantage of them. Certain Open Specifications documents are intended for use in conjunction with publicly available standards specifications and network programming art and, as such, assume that the reader either is familiar with the aforementioned material or has immediate access to it.

Support. For questions and support, please contact dochelp@microsoft.com.

Revision Summary

Date	Revision History	Revision Class	Comments
3/17/2010	0.1	New	Released new document.
3/26/2010	1.0	None	Introduced no new technical or language changes.
5/26/2010	1.2	None	Introduced no new technical or language changes.
9/8/2010	1.3	Major	Significantly changed the technical content.
10/13/2010	1.4	Minor	Clarified the meaning of the technical content.
2/10/2011	2.0	None	Introduced no new technical or language changes.
2/22/2012	3.0	Major	Significantly changed the technical content.
7/25/2012	3.1	Minor	Clarified the meaning of the technical content.
6/26/2013	4.0	Major	Significantly changed the technical content.
3/31/2014	4.0	None	No changes to the meaning, language, or formatting of the technical content.
1/22/2015	5.0	Major	Updated for new product version.
7/7/2015	5.1	Minor	Clarified the meaning of the technical content.
11/2/2015	5.2	Minor	Clarified the meaning of the technical content.
3/22/2016	5.3	Minor	Clarified the meaning of the technical content.
11/2/2016	5.3	None	No changes to the meaning, language, or formatting of the technical content.
3/14/2017	5.3	None	No changes to the meaning, language, or formatting of the technical content.
10/3/2017	5.3	None	No changes to the meaning, language, or formatting of the technical content.
2/22/2018	5.3	None	No changes to the meaning, language, or formatting of the technical content.
3/23/2018	5.3	None	No changes to the meaning, language, or formatting of the technical content.
8/28/2018	5.3	None	No changes to the meaning, language, or formatting of the technical content.

Table of Contents

1 Introduction

This document describes the level of support provided by the Microsoft XML Core Services (MSXML) 3.0 and 6.0 for *XSL Transformations (XSLT), Version 1.0* [W3C-XSLT], W3C Recommendation 16 November 1999.

The [W3C-XSLT] specification may contain guidance for authors of webpages and browser users, in addition to user agents (browser applications). Statements found in this document apply only to normative requirements in the specification targeted to user agents, not those targeted to authors.

1.1 Glossary

MAY, SHOULD, MUST, SHOULD NOT, MUST NOT: These terms (in all caps) are used as defined in [RFC2119]. All statements of optional behavior use either MAY, SHOULD, or SHOULD NOT.

1.2 References

Links to a document in the Microsoft Open Specifications library point to the correct section in the most recently published version of the referenced document. However, because individual documents in the library are not updated at the same time, the section numbers in the documents may not match. You can confirm the correct section numbering by checking the Errata.

1.2.1 Normative References

We conduct frequent surveys of the normative references to assure their continued availability. If you have any issue with finding a normative reference, please contact dochelp@microsoft.com. We will assist you in finding the relevant information.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997, http://www.rfc-editor.org/rfc/rfc2119.txt

[W3C-XSLT] World Wide Web Consortium, "XSL Transformations (XSLT) Version 1.0", W3C Recommendation 16 November 1999, http://www.w3.org/TR/1999/REC-xslt-19991116

1.2.2 Informative References

[MS-XPATH] Microsoft Corporation, "Microsoft XML XPath Standards Support Document".

[MSDN-NLS] Microsoft Corporation, "National Language Support Functions [Automation]", http://msdn.microsoft.com/en-us/library/ms221370.aspx

[UnicodeCollationAlgorithm] The Unicode Consortium, "Unicode Collation Algorithm", Unicode® Technical Standard #10 Version 9.0.0 2016-05-18, http://www.unicode.org/reports/tr10/index.html

[XPATH] Clark, J. and DeRose, S., "XML Path Language (XPath), Version 1.0", W3C Recommendation, November 1999, http://www.w3.org/TR/1999/REC-xpath-19991116/

1.3 Microsoft Implementations

Throughout this document, Microsoft XML Core Services (MSXML) 3.0 is referred to as *MSXML3* and Microsoft XML Core Services (MSXML) 6.0 is referred to as *MSXML6*.

MSXML3 is the only version of MSXML that is implemented in Windows Internet Explorer 7 and Windows Internet Explorer 8. Both MSXML3 and MSXML6 are implemented in Windows Internet Explorer 9, Windows Internet Explorer 10, Internet Explorer 11, and Internet Explorer 11 for Windows 10: MSXML3 is used in IE7 Mode and IE8 Mode, and MSXML6 is used in all other modes. MSXML6 is

the only version of MSXML implemented in Microsoft Edge, which uses it only to implement XSLT [W3C-XSLT]. Microsoft Edge provides [XPATH] functionality natively; see [MS-XPATH] for more information.

1.4 Standards Support Requirements

To conform to [W3C-XSLT], a user agent must implement all required portions of the specification. Any optional portions that have been implemented must also be implemented as described by the specification. Normative language is usually used to define both required and optional portions. (For more information, see [RFC2119].)

The following table lists the sections of [W3C-XSLT] and whether they are considered normative or informative.

Chapters	Normative/Informative
1-18	Normative
Appendices A-G	Informative

1.5 Notation

The following notations are used in this document to differentiate between notes of clarification, variation from the specification, and extension points.

Notation	Explanation
C####	Identifies a clarification of ambiguity in the target specification. This includes imprecise statements, omitted information, discrepancies, and errata. This does not include data formatting clarifications.
V####	Identifies an intended point of variability in the target specification such as the use of MAY, SHOULD, or RECOMMENDED. (See [RFC2119].) This does not include extensibility points.
E####	Identifies extensibility points (such as optional implementation-specific data) in the target specification, which can impair interoperability.

For document mode and browser version notation, see section 1.3.

2 Standards Support Statements

This section contains all variations and clarifications for the Microsoft implementation of [W3C-XSLT].

- Section 2.1 describes normative variations from the MUST requirements of the specification.

- Section 2.2 describes clarifications of the MAY and SHOULD requirements.

- Section 2.3 considers error handling aspects of the implementation.

- Section 2.4 considers security aspects of the implementation.

2.1   Normative Variations

The following subsections describe normative variations from the MUST requirements of [W3C-XSLT].

2.1.1   [W3C-XSLT] Section 1, Introduction

V0001:

The specification states:

```
The MIME media types text/xml and application/xml should be used for
XSLT stylesheets. It is possible that a media type will be registered specifically
for XSLT stylesheets; if and when it is, that media type may also be used.
```

MSXML3

An XML style sheet is transformed only when **type**="text/xsl". XML style sheets of **type**="text/xml" or of **type**="application/xml" are not transformed.

MSXML6

Processing instructions for XML style sheets are supported, but an XML style sheet is not automatically associated with XML content.

2.1.2   [W3C-XSLT] Section 2.7, Embedding Stylesheets

V0002:

The specification states:

```
NOTE: A stylesheet that is embedded in the document to which it is to be applied or
that may be included or imported into an stylesheet that is so embedded typically
needs to contain a template rule that specifies that xsl:stylesheet elements are to
be ignored.
```

MSXML3 and MSXML6

Embedded style sheets are not supported.

2.1.3   [W3C-XSLT] Section 12.1, Multiple Source Documents

V0003:

The specification states:

```
If the URI reference does not contain a fragment identifier, then a node-set
containing just the root node of the document is returned. If the URI reference
does contain a fragment identifier, the function returns a node-set containing the
nodes in the tree identified by the fragment identifier of the URI reference. The
semantics of the fragment identifier is dependent on the media type of the result
of retrieving the URI. If there is an error in processing the fragment identifier,
the XSLT processor may signal the error; if it does not signal the error, it must
recover by returning an empty node-set. Possible errors include:

    • The fragment identifier identifies something that cannot be represented by an
XSLT node-set (such as a range of characters within a text node).


    • The XSLT processor does not support fragment identifiers for the media-type
of the retrieval result. An XSLT processor is not required to support any
particular media types. The documentation for an XSLT processor should specify for
which media types the XSLT processor supports fragment identifiers.
```

MSXML3 and MSXML6

The fragment identifier in the **document()** function is not processed.

2.1.4 [W3C-XSLT] Section 12.3, Number Formatting

V0004:

The specification states:

```
The format pattern must not contain the currency sign (#x00A4); support for this
feature was added after the initial release of JDK 1.1. The decimal-format name
must be a QName, which is expanded as described in [2.4 Qualified Names]. It is an
error if the stylesheet does not contain a declaration of the decimal-format with the
specified expanded-name.
```

MSXML3 and MSXML6

The format pattern can contain the currency sign character reference (`#x00A4`).

2.1.5 [W3C-XSLT] Section 16.2, HTML Output Method

V0005:

The specification states:

```
The html output method should escape non-ASCII characters in URI attribute values
using the method recommended in Section B.2.1 of the HTML 4.0 Recommendation.
```

MSXML3 and MSXML6

The **href** attribute of the **a** element supports only ASCII characters.

2.2   Clarifications

The following subsections describe clarifications of the MAY and SHOULD requirements of [W3C-XSLT].

2.2.1 [W3C-XSLT] Section 1, Introduction

C0001:

The specification states:

```
This document does not specify how an XSLT stylesheet is associated with an XML
document. It is recommended that XSL processors support the mechanism described in
[XML Stylesheet]. When this or any other mechanism yields a sequence of more than
one XSLT stylesheet to be applied simultaneously to a XML document, then the effect
should be the same as applying a single stylesheet that imports each member of the
sequence in order (see [W3C-XSLT] [2.6.2 Stylesheet Import]).
```

MSXML3

An XML style sheet is transformed only where **type**="text/xsl". XML style sheets of
type="text/xml" or of **type**="application/xml" are not transformed.

MSXML6

Processing instructions for XML style sheets are supported, but an XML style sheet is not automatically
associated with XML content.

2.2.2 [W3C-XSLT] Section 2.1, XSLT Namespace

C0024:

The specification states:

```
XSLT processors must use the XML namespaces mechanism [XML Names] to recognize
elements and attributes from this namespace. Elements from the XSLT namespace
are recognized only in the stylesheet not in the source document. The complete
list of XSLT-defined elements is specified in [B Element Syntax Summary].
```

MSXML6

Only **Instruction** elements from the complete list of XSLT-defined elements specified in Appendix B of
[W3C-XSLT] are recognized.

2.2.3 [W3C-XSLT] Section 3.4, Whitespace Stripping

C0002:

The specification states:

```
Thus, the applicable match for a particular element name is determined as follows:

-First, any match with lower import precedence than another match is ignored.
-Next, any match with a NameTest that has a lower default priority than the default priority
of the NameTest of another match is ignored.

It is an error if this leaves more than one match. An XSLT processor may signal the
error; if it does not signal the error, it must recover by choosing, from amongst
the matches that are left, the one that occurs last in the stylesheet.
```

MSXML3 and MSXML6

No error is signaled for stripped white space. Instead, the last match in the style sheet is chosen.

2.2.4 [W3C-XSLT] Section 5.5, Conflict Resolution for Template Rules

C0003:

The specification states:

```
It is an error if [the algorithm in section 5.5] leaves more than one matching template rule.
An XSLT processor may signal the error; if it does not signal the error, it must recover by
choosing, from amongst the matching template rules that are left, the one that occurs last in
the stylesheet.
```

MSXML3 and MSXML6

No error is signaled. Instead, the last matching template rule in the style sheet is chosen.

2.2.5 [W3C-XSLT] Section 7.1.1, Literal Result Elements

C0004:

The specification states:

```
If a namespace URI is declared to be an alias for multiple different namespace URIs, then the
declaration with the highest import precedence is used.It is an error if there is more than
one such declaration. An XSLT processor may signal the error; if it does not signal the
error, it must recover by choosing,from amongst the declarations with the highest import
precedence, the one that occurs last in the stylesheet.
```

MSXML3 and MSXML6

No error is signaled. Instead, the last matching template rule in the style sheet is chosen.

2.2.6 [W3C-XSLT] Section 7.1.2, Creating Elements with xsl : element

C0005:

The specification states:

```
The name attribute is interpreted as an attribute value template. It is an error if
the string that results from instantiating the attribute value template is not a
QName. An XSLT processor may signal the error; if it does not signal the error,
then it must recover by making the result of instantiating the xsl:element
element be the sequence of nodes created by instantiating the content of the
xsl:element element, excluding any initial attribute nodes.
```

MSXML3 and MSXML6

An error is signaled when the string that results from instantiating the attribute value template is not a QName.

2.2.7 [W3C-XSLT] Section 7.1.3, Creating Attributes with xsl : attribute

C0006:

The specification states:

```
The name attribute is interpreted as an attribute value template. It is an error if
```

the string that results from instantiating the attribute value template is not a
QName or is the string xmlns. An XSLT processor may signal the error; if it does
not signal the error, it must recover by not adding the attribute to the result
tree.

MSXML3 and MSXML6

An error is signaled when the string that results from instantiating the attribute value template is not
a QName or is the string xmlns.

C0007:

The specification states:

XSLT processors may make use of the prefix of the **QName** specified in the **name**
attribute when selecting the prefix used for outputting the created attribute as
XML; however, they are not required to do so and, if the prefix is **xmlns,**
they must not do so. Thus, although it is not an error to do:
<xsl:attribute name="xmlns:xsl"
namespace="whatever">http://www.w3.org/1999/XSL/Transform</xsl:attribute>

it will not result in a namespace declaration being output.

MSXML3 and MSXML6

A new prefix for the namespace is automatically generated.

C0008:

The specification states:

The following are all errors:

   • Adding an attribute to an element after children have been added to it;
implementations may either signal the error or ignore the attribute.

   • Adding an attribute to a node that is not an element; implementations may
either signal the error or ignore the attribute.

   • Creating nodes other than text nodes during the instantiation of the content

of the **xsl:attribute** element; implementations may either signal the error or
ignore the offending nodes.

MSXML3 and MSXML6

No error is signaled. Instead, any attribute added to an element after children have been added to it is
ignored. Likewise, any node created other than text nodes during the instantiation of the content of
the **xsl:attribute** element is ignored.

2.2.8 [W3C-XSLT] Section 7.1.4, Named Attribute Sets

C0009:

The specification states:

It is an error if there are two attribute sets that have the same expanded-name and

equal import precedence and that both contain the same attribute, unless there is a
definition of the attribute set with higher import precedence that also contains
the attribute. An XSLT processor may signal the error; if it does not signal the
error, it must recover by choosing from amongst the definitions that specify the
attribute that have the highest import precedence the one that was specified last
in the stylesheet.

MSXML3 and MSXML6

No error is signaled. Instead, the last attribute that occurs in the style sheet is chosen.

2.2.9 [W3C-XSLT] Section 7.3, Creating Processing Instructions

C0010:

The specification states:

It is an error if the string that results from instantiating the name attribute is
not both an **NCName** and a **PITarget**. An XSLT processor may signal the error; if it
does not signal the error, it must recover by not adding the processing instruction
to the result tree.

MSXML3 and MSXML6

An error is signaled if the string that results from instantiating the **name** attribute is not both an
NCName and a PITarget.

C0011:

The specification states:

It is an error if instantiating the content of **xsl:processing-instruction** creates
nodes other than text nodes. An XSLT processor may signal the error; if it does not
signal the error, it must recover by ignoring the offending nodes together with
their content.

MSXML3 and MSXML6

The offending nodes and their content are ignored.

C0012:

The specification states:

It is an error if the result of instantiating the content of the
xsl:processing-instruction contains the string ?>. An XSLT processor may signal the
error; if it does not signal the error, it must recover by inserting a space after
any occurrence of ? that is followed by a >.

MSXML3 and MSXML6

A space is inserted after any occurrence of a "?" that occurs immediately before a ">". That is, a space
is inserted between a "?" and a ">". The result is "? >".

2.2.10 [W3C-XSLT] Section 7.4, Creating Comments

C0013:

The specification states:

> It is an error if instantiating the content of **xsl:comment** creates nodes other than
> text nodes. An XSLT processor may signal the error; if it does not signal the
> error, it must recover by ignoring the offending nodes together with their content.

MSXML3 and MSXML6

The offending nodes and their content are ignored.

C0014:

The specification states:

> It is an error if the result of instantiating the content of the **xsl:comment**
> contains the string -- or ends with -. An XSLT processor may signal the error; if
> it does not signal the error, it must recover by inserting a space after any
> occurrence of - that is followed by another - or that ends the comment.

MSXML3 and MSXML6

A space is inserted after any occurrence of a "–" that is immediately followed by another "–". That is, a space is inserted between a "–" and a "–". The result is "– –".

2.2.11 [W3C-XSLT] Section 10, Sorting

C0015:

The specification states:

> NOTE:It is possible for two conforming XSLT processors not to sort exactly the
> same. Some XSLT processors may not support some languages. Furthermore, there may
> be variations possible in the sorting of any particular language that are not
> specified by the attributes on **xsl:sort**, for example, whether Hiragana or Katakana
> is sorted first in Japanese. Future versions of XSLT may provide additional
> attributes to provide control over these variations. Implementations may also use
> implementation-specific namespaced attributes on **xsl:sort** for this.

MSXML3 and MSXML6

The processor does not use implementation-specific namespaced attributes on **xsl:sort**.

C0016:

The specification states:

> NOTE:It is recommended that implementers consult [UnicodeCollationAlgorithm]
> for information on internationalized sorting.

MSXML3 and MSXML6

The processor uses the functions in [MSDN-NLS] in Windows to sort Unicode strings.

2.2.12 [W3C-XSLT] Section 11.2, Values of Variables and Parameters

C0017:

The specification states:

```
It is an error if a member of the sequence of nodes created by instantiating the
template is an attribute node or a namespace node, since a root node cannot have an
attribute node or a namespace node as a child. An XSLT processor may signal the
error; if it does not signal the error, it must recover by not adding the attribute
node or namespace node.
```

MSXML3 and MSXML6

No attribute node or namespace node is added.

2.2.13 [W3C-XSLT] Section 12.1, Multiple Source Documents

C0018:

The specification states:

```
If there is an error retrieving the resource [an XML document specified by URI
reference], then the XSLT processor may signal an error; if it does not signal an
error, it must recover by returning an empty node-set. One possible kind of
retrieval error is that the XSLT processor does not support the URI scheme used by
the URI. An XSLT processor is not required to support any particular URI schemes.
The documentation for an XSLT processor should specify which URI schemes the XSLT
processor supports.
```

MSXML3 and MSXML6

If there is an error retrieving the resource, the XSLT processor signals an error and does not return an empty node set.

The XSLT processor supports the following URI schemes:

file

Local file

ftp

File Transfer Protocol

gopher

Gopher protocol

http

Hypertext Transfer Protocol

mailto

Electronic mail address

Note   In response to a mailto request, the XSLT processor first requests privilege to display a New Mail window using the default mail sender. The window is displayed only if privilege is granted.

2.2.14 [W3C-XSLT] Section 16, Output

C0019:

The specification states:

```
An XSLT processor may output the result tree as a sequence of bytes, although it is
not required to be able to do so (see [W3C-XSLT] [17 Conformance]). The xsl:output
element allows stylesheet authors to specify how they wish the result tree to be
output. If an XSLT processor outputs the result tree, it should do so as specified
by the xsl:output element; however, it is not required to do so.
```

MSXML3 and MSXML6

The processor does not produce the result tree as a sequence of bytes.

2.2.15 [W3C-XSLT] Section 16.1, XML Output Method

C0020:

The specification states:

```
NOTE: An XSLT processor may need to add namespace declarations in the course of
outputting the result tree as XML.
```

MSXML3 and MSXML6

The XSLT processor adds namespace declarations if needed.

2.2.16 [W3C-XSLT] Section 16.2, HTML Output Method

C0021:

The specification states:

```
The html output method may output a character using a character entity reference, if
one is defined for it in the version of HTML that the output method is using.
```

MSXML3 and MSXML6

The **html** output method writes a character using a character entity reference, if one is defined for it in the version of HTML that the output method is using.

2.2.17 [W3C-XSLT] Section 16.4, Disabling Output Escaping

C0022:

The specification states:

```
It is an error for output escaping to be disabled for a text node that is used for
something other than a text node in the result tree. Thus, it is an error to
disable output escaping for an xsl:value-of or xsl:text element that is used to
generate the string-value of a comment, processing instruction or attribute node;
it is also an error to convert a result tree fragment to a number or a string if
the result tree fragment contains a text node for which escaping was disabled. In
both cases, an XSLT processor may signal the error; if it does not signal the
```

error, it must recover by ignoring the **disable-output-escaping** attribute.

MSXML3 and MSXML6

No error is signaled. The XSLT processor recovers by ignoring the **disable-output-escaping** attribute.

C0023:

The specification states:

```
An XSLT processor is not required to support disabling output escaping. If an
xsl:value-of or xsl:text specifies that output escaping should be disabled and the
XSLT processor does not support this, the XSLT processor may signal an error; if it
does not signal an error, it must recover by not disabling output escaping.
```

MSXML3 and MSXML6

The XSLT processor supports disabling output escaping.

2.3   Error Handling

There are no additional error handling considerations.

2.4   Security

There are no additional security considerations.

3   Change Tracking

No table of changes is available. The document is either new or has had no changes since its last release.

4 Index