

[MS-XSLT]:

Microsoft XSLTransformations (XSLT) Standards Support Document

Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation for protocols, file formats, languages, standards as well as overviews of the interaction among each of these technologies.
- **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 may make copies of it in order to develop implementations of the technologies described in the Open Specifications and may distribute portions of it in your implementations using these technologies or your documentation as necessary to properly document the implementation. You may also distribute in your implementation, with or without modification, any schema, IDL's, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications.
- **No Trade Secrets.** Microsoft does not claim any trade secret rights in this documentation.
- **Patents.** Microsoft has patents that may cover your implementations of the technologies described in the Open Specifications. Neither this notice nor Microsoft's delivery of the documentation grants any licenses under those or any other Microsoft patents. However, a given Open Specification may be covered by Microsoft [Open Specification Promise](#) or the [Community Promise](#). If you would prefer a written license, or if the technologies described in the Open Specifications are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting iplg@microsoft.com.
- **Trademarks.** The names of companies and products contained in this documentation may be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights.
- **Fictitious Names.** The example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events 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 specifically described above, whether by implication, estoppel, or otherwise.

Tools. The Open Specifications do 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 are intended for use in conjunction with publicly available standard specifications and network programming art, and assumes that the reader either is familiar with the aforementioned material or has immediate access to it.

Revision Summary

Date	Revision History	Revision Class	Comments
03/17/2010	0.1	New	Released new document.
03/26/2010	1.0	None	Introduced no new technical or language changes.
05/26/2010	1.2	None	Introduced no new technical or language changes.
09/08/2010	1.3	Major	Significantly changed the technical content.
10/13/2010	1.4	Minor	Clarified the meaning of the technical content.
02/10/2011	2.0	No change	Introduced no new technical or language changes.
02/22/2012	3.0	Major	Significantly changed the technical content.
07/25/2012	3.1	Minor	Clarified the meaning of the technical content.

Table of Contents

1	Introduction	4
1.1	Glossary	4
1.2	References.....	4
1.2.1	Normative References	4
1.2.2	Informative References	4
1.3	Microsoft Implementations.....	4
1.4	Standards Support Requirements	5
1.5	Notation	5
2	Standards Support Statements.....	6
2.1	Normative Variations.....	6
2.1.1	[W3C-XSLT] Section 1, Introduction	6
2.1.2	[W3C-XSLT] Section 2.7, Embedding Stylesheets.....	6
2.1.3	[W3C-XSLT] Section 12.1, Multiple Source Documents.....	7
2.1.4	[W3C-XSLT] Section 12.3, Number Formatting	7
2.1.5	[W3C-XSLT] Section 16.2, HTML Output Method	7
2.2	Clarifications	8
2.2.1	[W3C-XSLT] Section 1, Introduction	8
2.2.2	[W3C-XSLT] Section 2.1, XSLT Namespace	8
2.2.3	[W3C-XSLT] Section 3.4, Whitespace Stripping	8
2.2.4	[W3C-XSLT] Section 5.5, Conflict Resolution for Template Rules	9
2.2.5	[W3C-XSLT] Section 7.1.1, Literal Result Elements.....	9
2.2.6	[W3C-XSLT] Section 7.1.2, Creating Elements with <code>xsl : element</code>	9
2.2.7	[W3C-XSLT] Section 7.1.3, Creating Attributes with <code>xsl : attribute</code>	10
2.2.8	[W3C-XSLT] Section 7.1.4, Named Attribute Sets.....	11
2.2.9	[W3C-XSLT] Section 7.3, Creating Processing Instructions	11
2.2.10	[W3C-XSLT] Section 7.4, Creating Comments	12
2.2.11	[W3C-XSLT] Section 10, Sorting.....	12
2.2.12	[W3C-XSLT] Section 11.2, Values of Variables and Parameters	13
2.2.13	[W3C-XSLT] Section 12.1, Multiple Source Documents	13
2.2.14	[W3C-XSLT] Section 16, Output	14
2.2.15	[W3C-XSLT] Section 16.1, XML Output Method	14
2.2.16	[W3C-XSLT] Section 16.2, HTML Output Method	14
2.2.17	[W3C-XSLT] Section 16.4, Disabling Output Escaping	15
2.3	Error Handling	15
2.4	Security.....	15
3	Change Tracking.....	16
4	Index	18

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 described in [RFC2119]. All statements of optional behavior use either MAY, SHOULD, or SHOULD NOT.

1.2 References

References to Microsoft Open Specifications documentation do not include a publishing year because links are to the latest version of the technical documents, which are updated frequently. References to other documents include a publishing year when one is available.

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. Please check the archive site, <http://msdn2.microsoft.com/en-us/library/E4BD6494-06AD-4aed-9823-445E921C9624>, as an additional source.

[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] Clark, J., Ed., "XSL Transformations (XSLT) Version 1.0", W3C Recommendation, November 1999, <http://www.w3.org/TR/1999/REC-xslt-19991116>

1.2.2 Informative References

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

[UnicodeCollationAlgorithm] Davis, M. and Whistler, K., "Unicode Collation Algorithm", 2009-10-08, <http://www.unicode.org/reports/tr10/index.html>

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 and Windows® Internet Explorer® 10.

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 a full list of variations, clarifications, and extension points in the Microsoft implementation of [\[W3C-XSLT\]](#).

- Section [2.1](#) includes only those variations that violate a MUST requirement in the target specification.
- Section [2.2](#) describes further variations from MAY and SHOULD requirements.
- Section [2.3](#) identifies variations in error handling.
- Section [2.4](#) identifies variations that impact security.

2.1 Normative Variations

The following subsections detail the normative variations from MUST requirements in [\[W3C-XSLT\]](#).

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 identify clarifications to recommendations made by [\[W3C-XSLT\]](#).

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 considerations for error handling.

2.4 Security

There are no additional security considerations.

3 Change Tracking

This section identifies changes that were made to the [MS-XSLT] protocol document between the February 2012 and July 2012 releases. Changes are classified as New, Major, Minor, Editorial, or No change.

The revision class **New** means that a new document is being released.

The revision class **Major** means that the technical content in the document was significantly revised. Major changes affect protocol interoperability or implementation. Examples of major changes are:

- A document revision that incorporates changes to interoperability requirements or functionality.
- An extensive rewrite, addition, or deletion of major portions of content.
- The removal of a document from the documentation set.
- Changes made for template compliance.

The revision class **Minor** means that the meaning of the technical content was clarified. Minor changes do not affect protocol interoperability or implementation. Examples of minor changes are updates to clarify ambiguity at the sentence, paragraph, or table level.

The revision class **Editorial** means that the language and formatting in the technical content was changed. Editorial changes apply to grammatical, formatting, and style issues.

The revision class **No change** means that no new technical or language changes were introduced. The technical content of the document is identical to the last released version, but minor editorial and formatting changes, as well as updates to the header and footer information, and to the revision summary, may have been made.

Major and minor changes can be described further using the following change types:

- New content added.
- Content updated.
- Content removed.
- New product behavior note added.
- Product behavior note updated.
- Product behavior note removed.
- New protocol syntax added.
- Protocol syntax updated.
- Protocol syntax removed.
- New content added due to protocol revision.
- Content updated due to protocol revision.
- Content removed due to protocol revision.
- New protocol syntax added due to protocol revision.

- Protocol syntax updated due to protocol revision.
- Protocol syntax removed due to protocol revision.
- New content added for template compliance.
- Content updated for template compliance.
- Content removed for template compliance.
- Obsolete document removed.

Editorial changes are always classified with the change type **Editorially updated**.

Some important terms used in the change type descriptions are defined as follows:

- **Protocol syntax** refers to data elements (such as packets, structures, enumerations, and methods) as well as interfaces.
- **Protocol revision** refers to changes made to a protocol that affect the bits that are sent over the wire.

The changes made to this document are listed in the following table. For more information, please contact protocol@microsoft.com.

Section	Tracking number (if applicable) and description	Major change (Y or N)	Change type
1 Introduction	Updated document to remove beta tagging.	N	Content updated.

4 Index

C

[Change tracking](#) 16
[Conflict Resolution for Template Rules](#) 9
[Creating Attributes with xsl : attribute](#) 10
[Creating Comments](#) 12
[Creating Elements with xsl : element](#) 9
[Creating Processing Instructions](#) 11

D

[Disabling Output Escaping](#) 15

E

[Embedding Stylesheets](#) 6

G

[Glossary](#) 4

H

HTML Output Method ([section 2.1.5](#) 7, [section 2.2.16](#) 14)

I

[Informative references](#) 4
Introduction ([section 1](#) 4, [section 2.1.1](#) 6, [section 2.2.1](#) 8)

L

[Literal Result Elements](#) 9

M

Multiple Source Documents ([section 2.1.3](#) 7, [section 2.2.13](#) 13)

N

[Named Attribute Sets](#) 11
[Normative references](#) 4
[Number Formatting](#) 7

O

[Output](#) 14

R

References
[informative](#) 4
[normative](#) 4

S

[Sorting](#) 12

T

[Tracking changes](#) 16

V

[Values of Variables and Parameters](#) 13

W

[Whitespace Stripping](#) 8

X

[XML Output Method](#) 14
[XSLT Namespace](#) 8