[MS-XMLH]:
Internet Explorer XML 1.0 (Fourth Edition) 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.

Copyright © 2012 Microsoft Corporation.

Revision Summary

Date	Revision History	Revision Class	Comments
09/08/2010	0.1	New	Released new document.
10/13/2010	0.2	Minor	Clarified the meaning of the technical content.
02/10/2011	1.0	No change	Introduced no new technical or language changes.
02/22/2012	2.0	Major	Significantly changed the technical content.
07/25/2012	2.1	Minor	Clarified the meaning of the technical content.

Copyright © 2012 Microsoft Corporation.

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 [XML] Section 4.7, Notation Declarations 6
2.2 Clarifications 6
2.2.1 [XML] Section 2.3, Common Syntactic Constructs 6
2.2.2 [XML] Section 2.8, Prolog and Document Type Declaration 7
2.2.3 [XML] Section 2.9, Standalone Document Declaration 7
2.2.4 [XML] Section 2.1, White Space Handling 8
2.2.5 [XML] Section 3, Logical Structures 8
2.2.6 [XML] Section 3.1, Start-Tags, End-Tags, and Empty-Element Tags 8
2.2.7 [XML] Section 3.2, Element Type Declarations 9
2.2.8 [XML] Section 3.2.1, Element Content 10
2.2.9 [XML] Section 3.2.2, Mixed Content 10
2.2.10 [XML] Section 3.3, Attribute-List Declarations 11
2.2.11 [XML] Section 3.3.1, Attribute Types 11
2.2.12 [XML] Section 3.3.2, Attribute Defaults 14
2.2.13 [XML] Section 3.4, Conditional Sections 16
2.2.14 [XML] Section 4.1, Character and Entity References 16
2.2.15 [XML] Section 4.2, Entity Declarations 16
2.2.16 [XML] Section 4.2.2, External Entities 17
2.2.17 [XML] Section 4.7, Notation Declarations 18
2.2.18 [XML] Section 5.1, Validating and Non-Validating Processors 18
2.2.19 [XML] Section 5.2, Using XML Processors 18
2.3 Error Handling 19
2.4 Security 19

3 Change Tracking 20

4 Index 22

Copyright © 2012 Microsoft Corporation.

1 Introduction

This document describes the level of support provided by Windows® Internet Explorer® 9 and Windows® Internet Explorer® 10 for the *Extensible Markup Language (XML) 1.0 (Fourth Edition)* [XML], W3C Recommendation 16 August 2006, edited in place 29 September 2006.

The [XML] 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.

[NamespacesXML1.1] Bray, T., Hollander, D., Layman, A., and Tobin, R., Eds., "Namespaces in XML 1.1 (Second Edition)", W3C Recommendation 16 August 2006, http://www.w3.org/TR/xml-names11/

[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

[XML] World Wide Web Consortium, "Extensible Markup Language (XML) 1.0 (Fourth Edition)", W3C Recommendation, August 2006, http://www.w3.org/TR/2006/REC-xml-20060816/

1.2.2 Informative References

None.

1.3 Microsoft Implementations

Windows® Internet Explorer® 9 and Windows® Internet Explorer® 10 implement some portion of the [XML] specification.

In addition, each version of ie implements multiple document modes, which can vary individually in their support of the standard. The following table lists the document modes available in each version of Windows® Internet Explorer®:

Copyright © 2012 Microsoft Corporation.

Browser Version	Documents Modes Supported
Internet Explorer 9	Quirks mode IE7 mode IE8 mode IE9 mode
Internet Explorer 10	Quirks mode IE7 mode IE8 mode IE9 mode IE10 Mode

Internet Explorer 9 and Internet Explorer 10 implement the [XML] specification as described in this document only in IE9 mode and IE10 Mode.

Throughout this document, the document mode appears first followed by the browser version in parentheses. Only those document modes and browser versions for which there is a variation note will be listed. If the document mode is not listed, conformance to the specification can be assumed.

1.4 Standards Support Requirements

To conform to [XML], 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 [XML] and whether they are considered normative or informative.

Chapters	Normative/Informative
1-2	Normative
Appendices A-E	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.

Copyright © 2012 Microsoft Corporation.

2 Standards Support Statements

This section contains a full list of clarifications in the Microsoft implementation of [XML].

- 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 [XML].

2.1.1 [XML] Section 4.7, Notation Declarations

V0001:

The specification states:

```
XML processors MUST provide applications with the name and external identifier(s)
of any notation declared and referred to in an attribute value, attribute
definition, or entity declaration. They MAY additionally resolve the external
identifier into the system identifier, file name, or other information needed to
allow the application to call a processor for data in the notation described. (It
is not an error, however, for XML documents to declare and refer to notations for
which notation-specific applications are not available on the system where the
XML processor or application is running.)
```

IE9 Mode and IE10 Mode (All Versions)

Names and external identifiers of declared notations are not provided to applications.

2.2 Clarifications

The following subsections identify clarifications to recommendations made by [XML].

2.2.1 [XML] Section 2.3, Common Syntactic Constructs

C0001:

The specification states:

```
The Namespaces in XML Recommendation [XML Names] assigns a meaning to names
containing colon characters. Therefore, authors should not use the colon in
XML names except for namespace purposes, but XML processors must accept the
colon as a name character.
```

IE9 Mode and IE10 Mode (All Versions)

Colons are accepted in names only when they are used in accordance with [NamespacesXML1.1].

Copyright © 2012 Microsoft Corporation.

2.2.2 [XML] Section 2.8, Prolog and Document Type Declaration

C0002:

The specification states:

```
Validity constraint: Root Element Type
The Name in the document type declaration MUST match the element type of
the root element.
```

IE9 Mode and IE10 Mode (All Versions)

The **Name** attribute in the document type declaration can differ from the element type of the root element.

2.2.3 [XML] Section 2.9, Standalone Document Declaration

C0003:

The specification states:

```
Validity constraint: Standalone Document Declaration
The standalone document declaration MUST have the value "no" if any external
markup declarations contain declarations of:

       attributes with default values, if elements to which these attributes apply appear in
the document without specifications of values for these attributes, or

       entities (other than amp, lt, gt, apos, quot), if references to those entities appear
in the document, or

       attributes with tokenized types, where the attribute appears in the document with a
value such that normalization will produce a different value from that which would be
produced in the absence of the declaration, or

       element types with element content, if white space occurs directly within any instance
of those types.
```

IE9 Mode and IE10 Mode (All Versions)

A standalone document declaration is not required to have a value of `no` based on the following external markup declarations:

- Attributes with default values, if elements to which these attributes apply appear in the document without specifications of values for these attributes.
- Entities, other than **amp**, **lt**, **gt**, **apos**, and **quot**, if references to those entities appear in the document.
- Attributes with tokenized types, where the attribute appears in the document with a value such that normalization produces a different value from what is produced in the absence of the declaration.
- Element types with element content, if white space occurs directly within any instance of those types.

Copyright © 2012 Microsoft Corporation.

2.2.4 [XML] Section 2.1, White Space Handling

C0004:

The specification states:

```
An XML processor MUST always pass all characters in a document that are not
markup through to the application. A validating XML processor MUST also inform
the application which of these characters constitute white space appearing
in element content.
```

IE9 Mode and IE10 Mode (All Versions)

Applications are not informed about which characters constitute white space in element content.

C0005:

The specification states:

```
The value "default" signals that applications' default white-space processing
modes are acceptable for this element; the value "preserve" indicates the
intent that applications preserve all the white space. This declared intent is
considered to apply to all elements within the content of the element where it
is specified, unless overridden with another instance of the xml:space attribute.
This specification does not give meaning to any value of xml:space other than
"default" and "preserve". It is an error for other values to be specified; the
XML processor MAY report the error or MAY recover by ignoring the attribute
specification or by reporting the (erroneous) value to the application.
Applications may ignore or reject erroneous values.
```

IE9 Mode and IE10 Mode (All Versions)

An error is reported if the **xml:space** attribute is used with a value other than `default` or `preserve`.

2.2.5 [XML] Section 3, Logical Structures

C0006:

The specification states:

```
Validity constraint: Element Valid
An element is valid if there is a declaration matching elementdecl where the Name matches the
element type, and one of the following holds: . . .
```

IE9 Mode and IE10 Mode (All Versions)

Elements are not checked for validity.

2.2.6 [XML] Section 3.1, Start-Tags, End-Tags, and Empty-Element Tags

C0007:

The specification states:

Copyright © 2012 Microsoft Corporation.

```
Validity constraint: Attribute Value Type
The attribute MUST have been declared; the value MUST be of the type declared
for it.
```

IE9 Mode and IE10 Mode (All Versions)

Attribute value types are not checked for validity.

C0008:

The specification states:

```
[Definition: An empty-element tag takes a special form:]
Tags for Empty Elements[44]
EmptyElemTag   ::=   '<' Name (S Attribute)* S? '/>'[WFC: Unique Att Spec]
Empty-element tags may be used for any element which has no content, whether
or not it is declared using the keyword EMPTY.
For interoperability, the empty-element tag SHOULD be used, and SHOULD only
be used, for elements which are declared EMPTY.
```

IE9 Mode and IE10 Mode (All Versions)

Empty element tags can be used for any element that has no content, including elements that are not declared by using the **EMPTY** keyword.

2.2.7 [XML] Section 3.2, Element Type Declarations

C0009:

The specification states:

```
The element structure of an XML document may, for validation purposes, be
constrained using element type and attribute-list declarations. An element
type declaration constrains the element's content.

Element type declarations often constrain which element types can appear as
children of the element. At user option, an XML processor MAY issue a warning
when a declaration mentions an element type for which no declaration is
provided, but this is not an error.
```

IE9 Mode and IE10 Mode (All Versions)

The element structure of an XML document is not constrained for validation purposes.

C0010:

The specification states:

```
Validity constraint: Unique Element Type Declaration
An element type MUST NOT be declared more than once.
```

IE9 Mode and IE10 Mode (All Versions)

Elements can be declared more than once.

Copyright © 2012 Microsoft Corporation.

2.2.8 [XML] Section 3.2.1, Element Content

C0011:

The specification states:

```
Definition: An element type has element content when elements of that type MUST
contain only child elements (no character data), optionally separated by white
space (characters matching the nonterminal S).] [Definition: In this case, the
constraint includes a content model, a simple grammar governing the allowed
types of the child elements and the order in which they are allowed to appear.]
The grammar is built on content particles (cps), which consist of names, choice
lists of content particles, or sequence lists of content particles:

Element-content Models . . .
```

IE9 Mode and IE10 Mode (All Versions)

Element content is not checked against the Element-content Models grammar.

C0012:

The specification states:

```
Validity constraint: Proper Group/PE Nesting

Parameter-entity replacement text MUST be properly nested with parenthesized
groups. That is to say, if either of the opening or closing parentheses in a
choice, seq, or Mixed construct is contained in the replacement text for a
parameter entity, both MUST be contained in the same replacement text.

For interoperability, if a parameter-entity reference appears in a choice, seq,
or Mixed construct, its replacement text SHOULD contain at least one non-blank
character, and neither the first nor last non-blank character of the replacement
text SHOULD be a connector (| or ,).
```

IE9 Mode and IE10 Mode (All Versions)

Parameter-entity replacement text does not have to be properly nested in parenthesized groups.

2.2.9 [XML] Section 3.2.2, Mixed Content

C0013:

The specification states:

```
Validity constraint: No Duplicate Types

The same name MUST NOT appear more than once in a single mixed-content declaration.
```

IE9 Mode and IE10 Mode (All Versions)

More than one instance of an element type may appear in a single mixed-content declaration.

Copyright © 2012 Microsoft Corporation.

2.2.10 [XML] Section 3.3, Attribute-List Declarations

C0014:

The specification states:

```
Definition: Attribute-list declarations specify the name, data type, and default value (if
any) of each attribute associated with a given element type:]

Attribute-list Declaration
[52]   AttlistDecl   ::=   '<!ATTLIST' S Name AttDef* S? '>'
[53]   AttDef   ::=   S Name S AttType S DefaultDecl

The Name in the AttlistDecl rule is the type of an element. At user option, an XML processor
MAY issue a warning if attributes are declared for an element type not itself declared, but
this is not an error. The Name in the AttDef rule is the name of the attribute.
```

IE9 Mode and IE10 Mode (All Versions)

No warning is issued when an attribute is declared for an element type that has not been declared.

C0015:

The specification states:

```
When more than one AttlistDecl is provided for a given element type, the contents
of all those provided are merged. When more than one definition is provided for
the same attribute of a given element type, the first declaration is binding and
later declarations are ignored. For interoperability, writers of DTDs may choose
to provide at most one attribute-list declaration for a given element type, at
most one attribute definition for a given attribute name in an attribute-list
declaration, and at least one attribute definition in each attribute-list
declaration. For interoperability, an XML processor MAY at user option issue a
warning when more than one attribute-list declaration is provided for a given
element type, or more than one attribute definition is provided for a given
attribute, but this is not an error.
```

IE9 Mode and IE10 Mode (All Versions)

No warning is issued when more than one attribute-list declaration is provided for an element type or more than one attribute definition is provided for a given attribute.

2.2.11 [XML] Section 3.3.1, Attribute Types

C0016:

The specification states:

```
Validity constraint: ID

Values of type ID MUST match the Name production. A name MUST NOT appear more
than once in an XML document as a value of this type; i.e., ID values MUST
uniquely identify the elements which bear them.
```

IE9 Mode and IE10 Mode (All Versions)

Copyright © 2012 Microsoft Corporation.

An **ID** value is not required to uniquely identify the element that is associated with it.

C0017:

The specification states:

```
Validity constraint: One ID per Element Type

An element type MUST NOT have more than one ID attribute specified.
```

IE9 Mode and IE10 Mode (All Versions)

An element type may have more than one **ID** attribute specified.

C0018:

The specification states:

```
Validity constraint: ID Attribute Default

An ID attribute MUST have a declared default of #IMPLIED or #REQUIRED.
```

IE9 Mode and IE10 Mode (All Versions)

An **ID** attribute is not required to have a declared default of `#IMPLIED` or `#REQUIRED`.

C0019:

The specification states:

```
Validity constraint: IDREF

Values of type IDREF MUST match the Name production, and values of type IDREFS
MUST match Names; each Name MUST match the value of an ID attribute on some
element in the XML document; i.e. IDREF values MUST match the value of some
ID attribute.
```

IE9 Mode and IE10 Mode (All Versions)

An **IDREF** entity attribute is not required to match the value of an **ID** attribute on an element in the XML document.

C0020:

The specification states:

```
Validity constraint: Entity Name

Values of type ENTITY MUST match the Name production, values of type ENTITIES
MUST match Names; each Name MUST match the name of an unparsed entity declared
in the DTD.
```

IE9 Mode and IE10 Mode (All Versions)

Copyright © 2012 Microsoft Corporation.

Entity attribute types and the **ENTITIES** set of entities are not checked for validity. Entity attribute types are not required to match the name of an unparsed entity that is declared in the DTD.

C0021:

The specification states:

```
Validity constraint: Name Token

Values of type NMTOKEN MUST match the Nmtoken production; values of type NMTOKENS
MUST match Nmtokens.
```

IE9 Mode and IE10 Mode (All Versions)

Name token attribute types and the **NMTOKENS** set of name tokens are not checked for validity.

C0022:

The specification states:

```
Validity constraint: Notation Attributes

Values of this type MUST match one of the notation names included in the
declaration; all notation names in the declaration MUST be declared.
```

IE9 Mode and IE10 Mode (All Versions)

A **notation** attribute is not required to match a notation that is specified in the notation declarations, and all of the notations that are declared in the notation declarations are not required to be declared.

C0023:

The specification states:

```
Validity constraint: One Notation Per Element Type

An element type MUST NOT have more than one NOTATION attribute specified.
```

IE9 Mode and IE10 Mode (All Versions)

An element type may have more than one **NOTATION** attribute.

C0024:

The specification states:

```
Validity constraint: No Notation on Empty Element

For compatibility, an attribute of type NOTATION MUST NOT be declared on an
element declared EMPTY.
```

IE9 Mode and IE10 Mode (All Versions)

Copyright © 2012 Microsoft Corporation.

A **NOTATION** attribute may be declared on an element that is declared by using the **EMPTY** keyword.

C0025:

The specification states:

```
Validity constraint: No Duplicate Tokens

The notation names in a single NotationType attribute declaration, as well as
the NmTokens in a single Enumeration attribute declaration, MUST all be distinct.
```

IE9 Mode and IE10 Mode (All Versions)

Both **NotationalType** attribute types and **Enumeration** attribute types may contain identical elements.

C0026:

The specification states:

```
Validity constraint: Enumeration

Values of this type MUST match one of the Nmtoken tokens in the declaration.
For interoperability, the same Nmtoken SHOULD NOT occur more than once in the
enumerated attribute types of a single element type.
```

IE9 Mode and IE10 Mode (All Versions)

Enumeration values are not required to be declared. The same token may occur more than once in an enumerated attribute type.

2.2.12 [XML] Section 3.3.2, Attribute Defaults

C0027:

The specification states:

```
In an attribute declaration, #REQUIRED means that the attribute MUST always be
provided, #IMPLIED that no default value is provided.
```

IE9 Mode and IE10 Mode (All Versions)

Attributes are not checked to determine if they are required or if they have default values.

C0028:

The specification states:

```
Definition: If the declaration is neither #REQUIRED nor #IMPLIED, then the
AttValue value contains the declared default value; the #FIXED keyword states
that the attribute MUST always have the default value. When an XML processor
encounters an element without a specification for an attribute for which it has
read a default value declaration, it MUST report the attribute with the declared
```


Copyright © 2012 Microsoft Corporation.

```
default value to the application
```

IE9 Mode and IE10 Mode (All Versions)

For an element that is not specified and that includes an attribute that has a declared default value, the attribute is not reported to the application.

C0029:

The specification states:

```
Validity constraint: Required Attribute

If the default declaration is the keyword #REQUIRED, then the attribute MUST
be specified for all elements of the type in the attribute-list declaration.
```

IE9 Mode and IE10 Mode (All Versions)

An attribute is not required to be specified for all element types in the attribute-list declaration.

C0030:

The specification states:

```
Validity constraint: Attribute Default Value Syntactically Correct

The declared default value MUST meet the syntactic constraints of the declared attribute
type. That is, the default value of an attribute: . . .

Note that only the syntactic constraints of the type are required here; other constraints
(e.g. that the value be the name of a declared unparsed entity, for an attribute of type
ENTITY) will be reported by a validating parser only if an element without a specification
for this attribute actually occurs.
```

IE9 Mode and IE10 Mode (All Versions)

The default value for an attribute is not checked against the syntactic constraints of the attribute type.

C0031:

The specification states:

```
Validity constraint: Fixed Attribute Default

If an attribute has a default value declared with the #FIXED keyword, instances
of that attribute MUST match the default value.
```

IE9 Mode and IE10 Mode (All Versions)

The value of an attribute is not checked against a fixed default value of the attribute type.

Copyright © 2012 Microsoft Corporation.

2.2.13 [XML] Section 3.4, Conditional Sections

C0032:

The specification states:

```
Validity constraint: Proper Conditional Section/PE Nesting

If any of the "<![", "[", or "]]>" of a conditional section is contained in the
replacement text for a parameter-entity reference, all of them MUST be contained
in the same replacement text.
```

IE9 Mode and IE10 Mode (All Versions)

All the syntactic elements that are referred to in a conditional section (**<![**, **[**, and **]]>**) are not checked to be present in the same replacement text.

2.2.14 [XML] Section 4.1, Character and Entity References

C0033:

The specification states:

```
Validity constraint: Entity Declared

In a document with an external subset or parameter entity references with
"standalone='no'", the Name given in the entity reference MUST match that in an
entity declaration. For interoperability, valid documents SHOULD declare the
entities amp, lt, gt, apos, quot, in the form specified in 4.6 Predefined
Entities. The declaration of a parameter entity MUST precede any reference to it.
Similarly, the declaration of a general entity MUST precede any attribute-list
declaration containing a default value with a direct or indirect reference to
that general entity.
```

IE9 Mode and IE10 Mode (All Versions)

An entity reference is not required to match an entity in the entity declaration.

2.2.15 [XML] Section 4.2, Entity Declarations

C0034:

The specification states:

```
The Name identifies the entity in an entity reference or, in the case of an
unparsed entity, in the value of an ENTITY or ENTITIES attribute. If the same
entity is declared more than once, the first declaration encountered is binding;
at user option, an XML processor MAY issue a warning if entities are declared
multiple times.
```

IE9 Mode and IE10 Mode (All Versions)

No warning is issued if entities are declared multiple times.

Copyright © 2012 Microsoft Corporation.

2.2.16 [XML] Section 4.2.2, External Entities

C0035:

The specification states:

```
External Entity Declaration

[75] ExternalID ::= 'SYSTEM' S SystemLiteral
                    | 'PUBLIC' S PubidLiteral S SystemLiteral
[76] NDataDecl  ::= S 'NDATA' S Name  [VC: Notation Declared]

If the NDataDecl is present, this is a general unparsed entity; otherwise it is a parsed
entity.

Validity constraint: Notation Declared

The Name MUST match the declared name of a notation.
```

IE9 Mode and IE10 Mode (All Versions)

An external entity notation is not required to match a declared notation.

C0036:

The specification states:

```
An XML processor attempting to retrieve the entity's content may use any
combination of the public and system identifiers as well as additional
information outside the scope of this specification to try to generate an
alternative URI reference. If the processor is unable to do so, it MUST use
the URI reference specified in the system literal. Before a match is attempted,
all strings of white space in the public identifier MUST be normalized to single
space characters (#x20), and leading and trailing white space MUST be removed.
```

IE9 Mode and IE10 Mode (All Versions)

External entities are not resolved.

If one of the following public identifiers is used, the entities that are defined by XHTML are resolved:

- `//W3C//DTD XHTML 1.0 Transitional//EN`
- `//W3C//DTD XHTML 1.1//EN`
- `//W3C//DTD XHTML 1.0 Strict//EN`
- `//W3C//DTD XHTML 1.0 Frameset//EN`
- `//W3C//DTD XHTML Basic 1.0//EN`
- `//W3C//DTD XHTML 1.1 plus MathML 2.0//EN`
- `//W3C//DTD XHTML 1.1 plus MathML 2.0 plus SVG 1.1//EN`
- `//W3C//DTD MathML 2.0//EN`
- `//WAPFORUM//DTD XHTML Mobile 1.0//EN`

Copyright © 2012 Microsoft Corporation.

2.2.17 [XML] Section 4.7, Notation Declarations

C0037:

The specification states:

```
Validity constraint: Unique Notation Name

A given Name MUST NOT be declared in more than one notation declaration.
```

IE9 Mode and IE10 Mode (All Versions)

A notation is not checked if it is unique.

2.2.18 [XML] Section 5.1, Validating and Non-Validating Processors

C0038:

The specification states:

```
Conforming XML processors fall into two classes: validating and non-validating.

Validating and non-validating processors alike MUST report violations of this
specification's well-formedness constraints in the content of the document entity
and any other parsed entities that they read.
```

IE9 Mode and IE10 Mode (All Versions)

A non-validating XML parser is used.

C0039:

The specification states:

```
[Definition: Validating processors MUST, at user option, report violations of
the constraints expressed by the declarations in the DTD, and failures to fulfill
the validity constraints given in this specification.] To accomplish this,
validating XML processors MUST read and process the entire DTD and all external
parsed entities referenced in the document.
```

IE9 Mode and IE10 Mode (All Versions)

A non-validating XML parser is used.

2.2.19 [XML] Section 5.2, Using XML Processors

C0040:

The specification states:

```
The behavior of a validating XML processor is highly predictable; it must read
every piece of a document and report all well-formedness and validity violations.
Less is required of a non-validating processor; it need not read any part of the
```


Copyright © 2012 Microsoft Corporation.

```
document other than the document entity. ...
```

IE9 Mode and IE10 Mode (All Versions)

A non-validating XML parser is used.

2.3 Error Handling

There are no additional considerations for error handling.

2.4 Security

There are no additional security considerations.

Copyright © 2012 Microsoft Corporation.

3 Change Tracking

This section identifies changes that were made to the [MS-XMLH] 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.

Copyright © 2012 Microsoft Corporation.

- 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 the document to remove beta tagging.	N	Content updated.

Copyright © 2012 Microsoft Corporation.

4 Index

A

Attribute Defaults 14
Attribute Types 11
Attribute-List Declarations 11

C

Change tracking 20
Character and Entity References 16
Common Syntactic Constructs 6
Conditional Sections 16

E

Element Content 10
Element Type Declarations 9
Entity Declarations 16
External Entities 17

G

Glossary 4

I

Informative references 4
Introduction 4

L

Logical Structures 8

M

Mixed Content 10

N

Normative references 4
Notation Declarations (section 2.1.1 6, section 2.2.17 18)

P

Prolog and Document Type Declaration 7

R

References
  informative 4
  normative 4

S

Standalone Document Declaration 7
Start-Tags - End-Tags - and Empty-Element Tags 8

T

Tracking changes 20

U

Using XML Processors 18

V

Validating and Non-Validating Processors 18

W

White Space Handling 8

Copyright © 2012 Microsoft Corporation.