

[MS-DOM2CEX]: Microsoft XML Extensions to the Document Object Model (DOM) Level 2 Core Specification

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
03/26/2010	1.0	New	Released new document.
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	Extension Overview (Synopsis).....	5
1.3.1	Organization of This Documentation	6
1.4	Relationship to Standards and Other Extensions	6
1.5	Applicability Statement.....	6
2	Extensions.....	7
2.1	Extensions to the Document Interface	7
2.1.1	Attributes.....	7
2.1.1.1	documentElement.....	7
2.1.2	Methods.....	8
2.1.2.1	createNode	8
2.1.2.2	nodeFromID.....	8
2.2	Extensions to the NamedNodeMap Interface.....	8
2.2.1	Methods.....	9
2.2.1.1	getQualifiedItem.....	9
2.2.1.2	nextNode.....	9
2.2.1.3	removeQualifiedItem.....	10
2.2.1.4	reset.....	10
2.3	Extensions to the Node Interface	10
2.3.1	Attributes.....	11
2.3.1.1	nodeTypeString	11
2.3.1.2	text	11
2.4	Extensions to the NodeList Interface	11
2.4.1	Methods.....	12
2.4.1.1	nextNode.....	12
2.4.1.2	reset.....	12
3	Security Considerations.....	13
4	Appendix A: Product Behavior.....	14
5	Change Tracking.....	15
6	Index	17

1 Introduction

This document describes extensions provided by the Microsoft XML Core Services (MSXML) 3.0 for the *Document Object Model (DOM) Level 2 Core Specification Version 1.0* [\[DOM Level 2 - Core\]](#), W3C Recommendation 13 November, 2000.

Sections 1.7 and 2 of this specification are normative and can contain the terms MAY, SHOULD, MUST, MUST NOT, and SHOULD NOT as defined in RFC 2119. All other sections and examples in this specification are informative.

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.

[DOM Level 2 - Core] W3C, "Document Object Model (DOM) Level 2 Core Specification Version 1.0", W3C Recommendation 13 November, 2000, <http://www.w3.org/TR/DOM-Level-2-Core/>

[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>

1.2.2 Informative References

[CSS-Level2-2009] Bos, B., Celik, T., Hickson, I., and Wium Lie, H., Eds., "Cascading Style Sheets Level 2 Revision 1 (CSS 2.1) Specification", W3C Candidate Recommendation 08 September 2009, <http://www.w3.org/TR/2009/CR-CSS2-20090908/>

[DOM Level 2 - HTML] W3C, "Document Object Model (DOM) Level 2 HTML Specification Version 1.0", W3C Recommendation, January 2003, <http://www.w3.org/TR/2003/REC-DOM-Level-2-HTML-20030109/>

[DOM Level 2 - Style] W3C, "Document Object Model (DOM) Level 2 Style Specification Version 1.0", W3C Recommendation, November 2000, <http://www.w3.org/TR/2000/REC-DOM-Level-2-Style-20001113/>

[ECMA-262] ECMA International, "ECMAScript Language Specification" ECMA-262, December 1999, <http://www.ecma-international.org/publications/standards/Ecma-262.htm>

[HTML] World Wide Web Consortium, "HTML 4.01 Specification", December 1999, <http://www.w3.org/TR/html4/>

[MS-CSS21E] Microsoft Corporation, "[Internet Explorer Extensions to the Cascading Style Sheets \(CSS\) 2.1 Specification](#)".

[MS-DOM2CE] Microsoft Corporation, "[Internet Explorer Extensions to the Document Object Model \(DOM\) Level 2 Core Specification](#)".

[MS-ES3EX] Microsoft Corporation, "[Microsoft JScript Extensions to the ECMAScript Language Specification 3rd Edition](#)".

[MS-HTML401E] Microsoft Corporation, "[Internet Explorer Extensions to the HTML 4.01 Specification](#)".

1.3 Extension Overview (Synopsis)

The extensions described in this document were selected for their applicability to [\[DOM Level 2 - Core\]](#).

The additional Document Object Model interfaces, attributes, methods and collections are organized based on section 1.2, Fundamental Interfaces, of [\[DOM Level 2 - Core\]](#) as follows:

Document

- **Attributes**

- [documentElement](#)

- **Methods**

- [createNode](#)

- [nodeFromID](#)

NamedNodeMap

- **Methods**

- [getQualifiedItem](#)

- [nextNode](#)

- [removeQualifiedItem](#)

- [reset](#)

Node

- **Attributes**

- [nodeTypeString](#)

- [text](#)

NodeList

- **Methods**

- [nextNode](#)

- [reset](#)

1.3.1 Organization of This Documentation

This document is organized as follows:

- **Interfaces:** The extensions are listed according to interface at the highest level.
- **Attributes, Methods:** The interface members are described at the next levels.

1.4 Relationship to Standards and Other Extensions

The following documents provide information on additional extensions provided by Microsoft XML.

- [\[MS-CSS21E\]](#): Extensions to the [\[CSS-Level2-2009\]](#) and [\[DOM Level 2 - Style\]](#) specifications.
- [\[MS-HTML401E\]](#): Extensions to the [\[HTML\]](#) and the [\[DOM Level 2 - HTML\]](#) specifications.
- [\[MS-DOM2CE\]](#): Extensions to the [\[HTML\]](#) and the [\[DOM Level 2 - HTML\]](#) specifications for Internet Explorer.
- [\[MS-ES3EX\]](#): Extensions to the ECMAScript [\[ECMA-262\]](#) specification.

1.5 Applicability Statement

This document specifies a set of extensions to the [\[DOM Level 2 - Core\]](#) specification. The extensions in this document provide access to some features that are unique to Microsoft XML Core Services (MSXML) 3.0, Windows® Internet Explorer® 7 and Windows® Internet Explorer® 8.

MSXML3 is the only version of MSXML that is implemented in Internet Explorer 7 and Internet Explorer 8. Both MSXML3 and MSXML6 are implemented in Windows® Internet Explorer® 9 and Windows® Internet Explorer® 10.

2 Extensions

This section specifies additional attributes and methods to elements from [\[DOM Level 2 - Core\]](#) that are available in Windows® Internet Explorer®.

The extensions to [\[DOM Level 2 - Core\]](#) are as follows:

- Extensions to the [Document](#) Interface
- Extensions to the [NamedNodeMap](#) Interface
- Extensions to the [Node](#) Interface
- Extensions to the [NodeList](#) Interface

2.1 Extensions to the Document Interface

The **Document** (IXMLDOMDocument) interface implements the **Document** interface in [\[DOM Level 2 - Core\]](#). The interface represents the top level of the XML source and includes members for retrieving and creating all other XML objects.

The **Document** interface is extended by the [documentElement](#) attribute. For details, see [Attributes](#).

The **Document** interface is extended by the following methods. For details, see [Methods](#).

- [createNode](#)
- [nodeFromID](#)

IDL Definition

```
interface Document : Node {  
    // Introduced in DOM Level 2:  
    Element          documentElement  
    // Introduced in DOM Level 2:  
    Node             createNode  
    Node             nodeFromId  
};
```

2.1.1 Attributes

The **Document** interface as specified in the [\[DOM Level 2 - Core\]](#) is extended by the [documentElement](#) attribute.

2.1.1.1 documentElement

documentElement of type DOMString

Contains the root element of the document.

The **documentElement** attribute is read/write. It returns an `IXMLDOMElement` that represents the single element that represents the root of the XML document tree. It returns Null if no root exists.

When setting the **documentElement** attribute, the specified element node is inserted into the child list of the document after any document type node. To precisely place the node within the children of the document, call the **insertBefore** method of the `IXMLDOMNode`.

The **parentNode** attribute is reset to the document node as a result of this operation.

2.1.2 Methods

The **Document** interface as specified in the [\[DOM Level 2 - Core\]](#) is extended by the addition of the following methods:

- [createNode](#)
- [nodeFromID](#)

2.1.2.1 createNode

The **createNode** method creates a node using the supplied type, name, and namespace.

Parameters

Type

A variant. A value that uniquely identifies the node type. This can be specified using either the integer value or the string value. For the complete list of values, see the [XML DOM Enumerated Constants](#) or the description for the `name` parameter.

name

A string containing the value for the new node's **nodeName** property. The relationship between the `name` and `Type` parameters is summarized in the Remarks section of this topic.

namespaceURI

A string defining the namespace URI. If specified, the node is created in the context of the `namespaceURI` parameter with the prefix specified on the node name. If the `name` parameter does not have a prefix, this is treated as the default namespace.

Return Values

Returns the newly created node.

2.1.2.2 nodeFromID

The **nodeFromID** method returns the node that matches the ID attribute.

Parameters

sID

The string containing the value of the ID to match.

Return Values

Returns the node that matches the supplied ID. If no nodes match, returns null.

2.2 Extensions to the NamedNodeMap Interface

The **NamedNodeMap** (IXMLDOMNamedNodeMap) interface implements the **NamedNodeMap** interface in [\[DOM Level 2 - Core\]](#). The interface adds support for namespaces and iteration through the collection of attribute nodes.

The **NamedNodeMap** interface is extended by the following methods. For details, see [Methods](#).

- [getQualifiedItem](#)
- [nextNode](#)
- [removeQualifiedItem](#)
- [reset](#)

IDL Definition

```
interface NamedNodeMap {  
    // Introduced in DOM Level 2:  
    Node          getQualifiedItem (in DOMString namespaceURI);  
    Node          nextNode;  
    Node          removeQualifiedItem (in DOMString namespaceURI);  
    Node          reset;  
};
```

2.2.1 Methods

The **NamedNodeMap** interface as specified in the [\[DOM Level 2 - Core\]](#) is extended by the addition of the following methods:

- [getQualifiedItem](#)
- [nextNode](#)
- [removeQualifiedItem](#)
- [reset](#)

2.2.1.1 getQualifiedItem

The **getQualifiedItem** method returns the attribute with the specified namespace and attribute name.

Parameters

`baseName`

The string specifying the base name of the attribute, without namespace qualification.

`namespaceURI`

The string specifying the namespace prefix that qualifies the attribute name.

Return Value

The **getQualifiedItem** method returns the attribute node specified by the `baseName` and `namespaceURI` parameters. Returns Null if the attribute is not in the collection or if the item is not an attribute.

2.2.1.2 nextNode

The **nextNode** method returns the next node in the collection.

Parameters

None.

Return Value

The **nextNode** method returns an `IXMLDOMNode`, which refers to the next node in the collection. Returns Null if there is no next node.

2.2.1.3 removeQualifiedItem

The **removeQualifiedItem** method removes the attribute with the specified namespace and attribute name.

Parameters

`baseName`

The string specifying the base name of the attribute, without namespace qualification.

`namespaceURI`

The string specifying the namespace prefix that qualifies the attribute name.

Return Value

The **removeQualifiedItem** method returns the attribute node removed, or Null if no node was removed.

2.2.1.4 reset

The **reset** method resets the iterator.

Parameters

None.

Return Value

None.

No JScript Error

2.3 Extensions to the Node Interface

The **Node** (`IXMLDOMNode`) interface implements the **Node** interface in [\[DOM Level 2 - Core\]](#). The interface represents the primary datatype for the entire Document Object Model.

The **Node** interface is extended by the following attributes. For details, see [Attributes](#).

- [nodeTypeString](#)
- [text](#)

IDL Definition

```
interface Node {  
    // Introduced in DOM Level 2;
```

```

        readonly attribute Node           nodeTypeString;
        read-write attribute Node         text;
    };

```

2.3.1 Attributes

The **Node** interface as specified in the [\[DOM Level 2 - Core\]](#) is extended by the addition of the following attributes:

- [nodeTypeString](#)
- [text](#)

2.3.1.1 nodeTypeString

nodeTypeString of type `DOMString`, **read-only**

Returns the node type in string form.

The **nodeTypeString** attribute is read-only. It contains the string version of the node type. To return the enumeration value, use the **nodeType** attribute.

2.3.1.2 text

text of type `DOMString`

Represents the text content of the node or the concatenated text representing the node and its descendants.

The `text` attribute is read/write. When concatenated, the text represents the contents of text or CDATA nodes. All concatenated text nodes are normalized according to **xml:space** attributes and the value of the `preserveWhiteSpace` switch. Concatenated CDATA text is not normalized. (Child nodes that contain `NODE_COMMENT` and `NODE_PROCESSING_INSTRUCTION` nodes are not concatenated.) The **text** attribute trims the whitespace on the edges of the result, and "normalizes" `\r\n => \n`, but otherwise just concatenates text.

Retrieves and sets the string representing the text contents of this node or the concatenated text representing this node and its descendants.

For more precise control over text manipulation in an XML document, use the lower-level **nodeValue** property, which returns the raw text associated with a `NODE_TEXT` node.

2.4 Extensions to the NodeList Interface

The **NodeList** (`IXMLDOMNodeList`) interface implements the **NodeList** interface in [\[DOM Level 2 - Core\]](#). This supports iteration through the live collection, in addition to indexed access.

The **NodeList** interface is extended by the following attributes. For details see [Methods](#).

- [nextNode](#)
- [reset](#)

IDL Definition

```
interface NodeList {  
    Node      nextNode;  
    Node      reset;  
};
```

2.4.1 Methods

The **NodeList** interface as specified in the [\[DOM Level 2 - Core\]](#) is extended by the addition of the following methods:

- [nextNode](#)
- [reset](#)

2.4.1.1 nextNode

The **nextNode** method returns the next node in the collection.

Parameters

None

Return Value

An `IXMLDOMNode` refers to the next node in the collection. Returns Null if there is no next node.

2.4.1.2 reset

Resets the iterator.

The **reset** method reinitializes the iterator to point before the first node in the `IXMLDOMNodeList` so that the next call to `nextNode` returns the first item in the list.

Parameters

None.

Return Value

None.

No JScript Error

3 Security Considerations

There are no additional security considerations.

4 Appendix A: Product Behavior

The information in this specification is applicable to the following Microsoft products or supplemental software. References to product versions include released service packs:

- Windows® Internet Explorer® 7
- Windows® Internet Explorer® 8
- Windows® Internet Explorer® 9
- Windows® Internet Explorer® 10

Exceptions, if any, are noted below. If a service pack or Quick Fix Engineering (QFE) number appears with the product version, behavior changed in that service pack or QFE. The new behavior also applies to subsequent service packs of the product unless otherwise specified. If a product edition appears with the product version, behavior is different in that product edition.

Unless otherwise specified, any statement of optional behavior in this specification that is prescribed using the terms SHOULD or SHOULD NOT implies product behavior in accordance with the SHOULD or SHOULD NOT prescription. Unless otherwise specified, the term MAY implies that the product does not follow the prescription.

5 Change Tracking

This section identifies changes that were made to the [MS-DOM2CEX] 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.

6 Index

A

[Applicability](#) 6
Attributes
 [documentElement](#) 7
 [nodeTypeString](#) 11
 [text](#) 11

C

[Change tracking](#) 15

G

[Glossary](#) 4

I

[Implementer - security considerations](#) 13
[Informative references](#) 4
Interfaces
 [Document](#) 7
 [NamedNodeMap](#) 8
 [Node](#) 10
 [NodeList](#) 11
[Introduction](#) 4

M

Methods
 [createNode](#) 8
 [getQualifiedItem](#) 9
 nextNode ([section 2.2.1.2](#) 9, [section 2.4.1.1](#) 12)
 [nodeFromID](#) 8
 [removeQualifiedItem](#) 10
 reset ([section 2.2.1.4](#) 10, [section 2.4.1.2](#) 12)

N

[Normative references](#) 4

O

[Overview \(synopsis\)](#) 5

P

[Product behavior](#) 14

R

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

S

[Security - implementer considerations](#) 13

T

[Tracking changes](#) 15