[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 ("this documentation") for protocols, file formats, data portability, computer languages, and standards support. Additionally, overview documents cover inter-protocol relationships and interactions.

- **Copyrights.** This documentation is covered by Microsoft copyrights. Regardless of any other terms that are contained in the terms of use for the Microsoft website that hosts this documentation, you can make copies of it in order to develop implementations of the technologies that are described in this documentation and can distribute portions of it in your implementations that use these technologies or in your documentation as necessary to properly document the implementation. You can also distribute in your implementation, with or without modification, any schemas, IDLs, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications documentation.

- **No Trade Secrets.** Microsoft does not claim any trade secret rights in this documentation.

- **Patents.** Microsoft has patents that might cover your implementations of the technologies described in the Open Specifications documentation. Neither this notice nor Microsoft’s delivery of this documentation grants any licenses under those patents or any other Microsoft patents. However, a given Open Specifications document might be covered by the Microsoft Open Specifications Promise or the Microsoft Community Promise. If you would prefer a written license, or if the technologies described in this documentation are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting iplg@microsoft.com.

- **License Programs.** To see all of the protocols in scope under a specific license program and the associated patents, visit the Patent Map.

- **Trademarks.** The names of companies and products contained in this documentation might be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights. For a list of Microsoft trademarks, visit www.microsoft.com/trademarks.

- **Fictitious Names.** The example companies, organizations, products, domain names, email addresses, logos, people, places, and events that are depicted in this documentation are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

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

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

**Support.** For questions and support, please contact dochelp@microsoft.com.
## Revision Summary

<table>
<thead>
<tr>
<th>Date</th>
<th>Revision History</th>
<th>Revision Class</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>5/26/2010</td>
<td>1.2</td>
<td>None</td>
<td>Introduced no new technical or language changes.</td>
</tr>
<tr>
<td>9/8/2010</td>
<td>1.3</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
<tr>
<td>10/13/2010</td>
<td>1.4</td>
<td>Minor</td>
<td>Clarified the meaning of the technical content.</td>
</tr>
<tr>
<td>2/10/2011</td>
<td>2.0</td>
<td>None</td>
<td>Introduced no new technical or language changes.</td>
</tr>
<tr>
<td>2/22/2012</td>
<td>3.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
<tr>
<td>7/25/2012</td>
<td>3.1</td>
<td>Minor</td>
<td>Clarified the meaning of the technical content.</td>
</tr>
<tr>
<td>6/26/2013</td>
<td>4.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
<tr>
<td>3/31/2014</td>
<td>4.0</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>1/22/2015</td>
<td>5.0</td>
<td>Major</td>
<td>Updated for new product version.</td>
</tr>
<tr>
<td>7/7/2015</td>
<td>5.1</td>
<td>Minor</td>
<td>Clarified the meaning of the technical content.</td>
</tr>
<tr>
<td>11/2/2015</td>
<td>5.2</td>
<td>Minor</td>
<td>Clarified the meaning of the technical content.</td>
</tr>
<tr>
<td>1/20/2016</td>
<td>5.3</td>
<td>Minor</td>
<td>Clarified the meaning of the technical content.</td>
</tr>
<tr>
<td>3/22/2016</td>
<td>5.4</td>
<td>Minor</td>
<td>Clarified the meaning of the technical content.</td>
</tr>
<tr>
<td>11/2/2016</td>
<td>5.4</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>3/14/2017</td>
<td>5.4</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>10/3/2017</td>
<td>5.4</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>2/22/2018</td>
<td>5.4</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>3/23/2018</td>
<td>5.4</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>8/28/2018</td>
<td>5.4</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
</tbody>
</table>
# 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 ....................................................................................................................... 16
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], published 13 November 2000.

Section 2 of this specification is normative. 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 defined in [RFC2119]. All statements of optional behavior use either MAY, SHOULD, or SHOULD NOT.

1.2 References

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

1.2.1 Normative References

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


1.2.2 Informative References


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.

#### 1.5 Applicability Statement

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

MSXML3 is the only version of MSXML that is implemented in Windows Internet Explorer 7 and Windows Internet Explorer 8. Both MSXML3 and MSXML6 are implemented in Windows Internet Explorer 9, Windows Internet Explorer 10, Internet Explorer 11, and Internet Explorer 11 for Windows 10: MSXML3 is used in IE7 Mode and IE8 Mode, and MSXML6 is used in all other modes. MSXML6 is the only version of MSXML implemented in Microsoft Edge, which uses it only to implement XSLT [W3C-XSLT]. Microsoft Edge provides [XPATH] functionality natively; see [MS-XPATH] for more information.
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

```idl
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
IDL Definition

```idl
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](https://www.w3.org/TR/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](https://www.w3.org/TR/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**

```idl
interface Node {
    // Introduced in DOM Level 2:
    get Node nodeTypeString;
    get 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 updates to those products.

- Windows Internet Explorer 7
- Windows Internet Explorer 8
- Windows Internet Explorer 9
- Windows Internet Explorer 10
- Internet Explorer 11
- Microsoft Edge

Exceptions, if any, are noted in this section. If an update version, service pack or Knowledge Base (KB) number appears with a product name, the behavior changed in that update. The new behavior also applies to subsequent updates 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

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