[MS-DOM2CEX]:

Microsoft XML Extensions to the Document Object Model (DOM) Level 2 Core Specification

Intellectual Property Rights Notice for Open Specifications Documentation

Revision Summary

Date	Revision History	Revision Class	Comments
3/26/2010	1.0	New	Released new document.
5/26/2010	1.2	None	Introduced no new technical or language changes.
9/8/2010	1.3	Major	Significantly changed the technical content.
10/13/2010	1.4	Minor	Clarified the meaning of the technical content.
2/10/2011	2.0	None	Introduced no new technical or language changes.
2/22/2012	3.0	Major	Significantly changed the technical content.
7/25/2012	3.1	Minor	Clarified the meaning of the technical content.
6/26/2013	4.0	Major	Significantly changed the technical content.
3/31/2014	4.0	None	No changes to the meaning, language, or formatting of the technical content.
1/22/2015	5.0	Major	Updated for new product version.
7/7/2015	5.1	Minor	Clarified the meaning of the technical content.
11/2/2015	5.2	Minor	Clarified the meaning of the technical content.
1/20/2016	5.3	Minor	Clarified the meaning of the technical content.
3/22/2016	5.4	Minor	Clarified the meaning of the technical content.
11/2/2016	5.4	None	No changes to the meaning, language, or formatting of the technical content.
3/14/2017	5.4	None	No changes to the meaning, language, or formatting of the technical content.
10/3/2017	5.4	None	No changes to the meaning, language, or formatting of the technical content.
2/22/2018	5.4	None	No changes to the meaning, language, or formatting of the technical content.
3/23/2018	5.4	None	No changes to the meaning, language, or formatting of the technical content.
8/28/2018	5.4	None	No changes to the meaning, language, or formatting of the technical content.

Table of Contents

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.

[DOM Level 2 - Core] World Wide Web Consortium, "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] World Wide Web Consortium, "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] World Wide Web Consortium, "Document Object Model (DOM) Level 2 HTML Specification Version 1.0", W3C Recommendation 09 January 2003, http://www.w3.org/TR/2003/REC-DOM-Level-2-HTML-20030109/

[DOM Level 2 - Style] World Wide Web Consortium, "Document Object Model (DOM) Level 2 Style Specification Version 1.0", W3C Recommendation 13 November 2000, http://www.w3.org/TR/2000/REC-DOM-Level-2-Style-20001113/

[ECMA-262-1999] Ecma International, "ECMAScript Language Specification", Standard ECMA-262 3rd Edition - December 1999, http://www.ecma-international.org/publications/files/ECMA-ST-ARCH/ECMA-262,%203rd%20edition,%20December%201999.pdf

[HTML] World Wide Web Consortium, "HTML 4.01 Specification", W3C Recommendation, December 1999, http://www.w3.org/TR/html4/

[MS-CSS21E] Microsoft Corporation, "Internet Explorer Extensions to Cascading Style Sheets (CSS) 2.1 and DOM Level 2 Style Specifications".

[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 Third Edition".

[MS-HTML401E] Microsoft Corporation, "Internet Explorer Extensions to HTML 4.01 and DOM Level 2 HTML Specifications".

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

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

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

1.3   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-262-1999] specification.

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

```
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 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