**[MS-DOM3C]:**

**Internet Explorer Document Object Model (DOM) Level 3**

**Core Standards Support Document**

**Intellectual Property Rights Notice for Open Specifications Documentation**

* **Technical Documentation.** Microsoft publishes Open Specifications documentation forprotocols, 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 otherterms 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 technologiesdescribed 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](http://go.microsoft.com/fwlink/?LinkId=214445) or the  [Communit](http://go.microsoft.com/fwlink/?LinkId=214448)y  [Promis](http://go.microsoft.com/fwlink/?LinkId=214448)e. 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](mailto:iplg@microsoft.com).
* **Trademarks.** The names of companies and products contained in this documentation may becovered 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-mailaddresses, 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 rightsother than specifically described above, whether by implication, estoppel, or otherwise.

**Tools.** The Open Specifications do not require the use of Microsoft programming tools orprogramming 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.

*1 / 35*

*[MS-DOM3C] — v20120725*

*Internet Explorer Document Object Model (DOM) Level 3 Core Standards Support Document*

*Copyright © 2012 Microsoft Corporation.*

*Release: July 25, 2012*

**Revision Summary**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Revision History** |  |  |  |
| **Date** | **Revision Class** | **Comments** |  |
|  |  |  |  |  |
| 03/17/2010 | 0.1 | New | Released new document. |  |
|  |  |  |  |  |
| 03/26/2010 | 1.0 | None | Introduced no new technical or language changes. |  |
|  |  |  |  |  |
| 05/26/2010 | 1.2 | None | Introduced no new technical or language changes. |  |
|  |  |  |  |  |
| 09/08/2010 | 1.3 | Major | Significantly changed the technical content. |  |
|  |  |  |  |  |
| 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. |  |
|  |  |  |  |  |

*2 / 35*

*[MS-DOM3C] — v20120725*

*Internet Explorer Document Object Model (DOM) Level 3 Core Standards Support Document*

*Copyright © 2012 Microsoft Corporation.*

*Release: July 25, 2012*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table of Contents** | | | |  |
| **1** | [**Introduction ............................................................................................................**](#page4)**.** | | | **4** |
|  | 1.1 | [Glossary ..............................................................................................................](#page4). | | 4 |
|  | 1.2 | [References...........................................................................................................](#page4). | | 4 |
|  | 1.2.1 | | NormativeReferences....................................................................................... | 4 |
|  | 1.2.2 | | InformativeReferences..................................................................................... | 4 |
|  | 1.3 | [Microsoft Implementations .....................................................................................](#page4). | | 4 |
|  | 1.4 | [Standards Support Requirements ...........................................................................](#page5). | | 5 |
|  | 1.5 | [Notation ..............................................................................................................](#page6). | | 6 |
| **2** | [**Standards Support Statements................................................................................**](#page7)**.** | | | **7** |
|  | 2.1 | [Normative Variations.............................................................................................](#page7). | | 7 |
|  | [2.1.1 [DOM Level 3 - Core] Section 1.3.1, String Comparisons in the DOM .....................](#page7). | | | 7 |
|  | [2.1.2 [DOM Level 3 - Core] Section 1.3.3, XML Namespaces .........................................](#page7). | | | 7 |
|  | [2.1.3 [DOM Level 3 - Core] Section 1.4, Fundamental Interfaces: Core Module ...............](#page8). | | | 8 |
|  | [2.1.4 [DOM Level 3 - Core] Section 1.5, Extended Interfaces: XML Module ...................](#page29). | | | 29 |
|  | 2.2 | [Clarifications ......................................................................................................](#page29). | | 29 |
|  | [2.2.1 [DOM Level 3 - Core] Section 1.4, Fundamental Interfaces: Core Module .............](#page29). | | | 29 |
|  | 2.3 | [Error Handling ....................................................................................................](#page31). | | 31 |
|  | 2.4 | [Security.............................................................................................................](#page31). | | 31 |
| **3** | [**Change Tracking....................................................................................................**](#page32)**.** | | | **32** |
| **4** | **Index** | | **.....................................................................................................................** | **34** |

*3 / 35*

*[MS-DOM3C] — v20120725*

*Internet Explorer Document Object Model (DOM) Level 3 Core Standards Support Document*

*Copyright © 2012 Microsoft Corporation.*

*Release: July 25, 2012*

**1 Introduction**

This document describes the level of support provided by Windows® Internet Explorer® for the

*Document Object Model (DOM) Level 3 Core Specification Version 1.0* [[DOM Level 3 - Core](http://go.microsoft.com/fwlink/?LinkId=182717)],W3CRecommendation 07 April 2004.

The  [[DOM Level 3 - Core]](http://go.microsoft.com/fwlink/?LinkId=182717) 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 asdescribed in  [[RFC2119](http://go.microsoft.com/fwlink/?LinkId=90317)]. 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.co](mailto:dochelp@microsoft.com)m. We will assist you in finding the relevant information. Please check the archive site,  [http://msdn2.microsoft.com/en-us/library/E4BD6494-06AD-4aed-9823-445E921C962](http://msdn2.microsoft.com/en-us/library/E4BD6494-06AD-4aed-9823-445E921C9624)4, as an additional source.

[DOM Level 3 - Core] W3C, "Document Object Model (DOM) Level 3 Core Specification Version 1.0", W3C Recommendation 07 April 2004,  [http://www.w3.org/TR/DOM-Level-3-Core](http://go.microsoft.com/fwlink/?LinkId=182717)/

[DOM Level 3 - LS] W3C, "Document Object Model (DOM) Level 3 Load and Save Specification Version 1.0", W3C Recommendation 07 April 2004,  [http://www.w3.org/TR/DOM-Level-3-LS](http://go.microsoft.com/fwlink/?LinkId=182721)/

[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.tx](http://go.microsoft.com/fwlink/?LinkId=90317)t

[XPointer] Grosso, P., Maler, E., Marsh, J., and Walsh, N., "XPointer Framework", W3C Recommendation 25 March 2003,  [http://www.w3.org/TR/2003/REC-xptr-framework-20030325](http://go.microsoft.com/fwlink/?LinkID=183836&clcid=0x409)/

**1.2.2 Informative References**

None.

**1.3 Microsoft Implementations**

The following Microsoft products implement some portion of  [[DOM Level 3 - Core](http://go.microsoft.com/fwlink/?LinkId=182717)]:

Windows® Internet Explorer® 7

Windows® Internet Explorer® 8

*4 / 35*

*[MS-DOM3C] — v20120725*

*Internet Explorer Document Object Model (DOM) Level 3 Core Standards Support Document*

*Copyright © 2012 Microsoft Corporation.*

*Release: July 25, 2012*

Windows® Internet Explorer® 9

Windows® Internet Explorer® 10

In addition, each version of Windows® Internet Explorer® 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 Internet Explorer.

|  |  |  |
| --- | --- | --- |
| **Browser version** | **Document modes supported** |  |
|  |
|  |  |  |
| Internet Explorer 7 | Quirks mode |  |
|  | Standards mode |  |
|  |  |  |
| Internet Explorer 8 | Quirks mode |  |
|  | IE7 mode |  |
|  | IE8 mode |  |
|  |  |  |
| Internet Explorer 9 | Quirks mode |  |
|  | IE7 mode |  |
|  | IE8 mode |  |
|  | IE9 mode |  |
|  |  |  |
| Internet Explorer 10 | Quirks mode |  |
|  | IE7 mode |  |
|  | IE8 mode |  |
|  | IE9 mode |  |
|  | IE10 Mode |  |
|  |  |  |

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

**Note** "Standards mode" in Internet Explorer 7 and "IE7 mode" in Internet Explorer 8 refer to thesame document mode. "IE7 mode" is the preferred way of referring to this document mode across all versions of the browser.

**1.4 Standards Support Requirements**

To conform to  [[DOM Level 3 - Core]](http://go.microsoft.com/fwlink/?LinkId=182717) 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](http://go.microsoft.com/fwlink/?LinkId=90317)].)

The following table lists the sections of  [[DOM Level 3 - Core]](http://go.microsoft.com/fwlink/?LinkId=182717) and whether they are considered normative or informative.

**Sections** **Normative/Informative**

|  |  |
| --- | --- |
| 1 | Normative |
|  |  |
| Appendix A-F | Informative |
|  |  |

*5 / 35*

*[MS-DOM3C] — v20120725*

*Internet Explorer Document Object Model (DOM) Level 3 Core Standards Support Document*

*Copyright © 2012 Microsoft Corporation.*

*Release: July 25, 2012*

**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](http://go.microsoft.com/fwlink/?LinkId=90317)].) 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](#page4).

*6 / 35*

*[MS-DOM3C] — v20120725*

*Internet Explorer Document Object Model (DOM) Level 3 Core Standards Support Document*

*Copyright © 2012 Microsoft Corporation.*

*Release: July 25, 2012*

**2 Standards Support Statements**

This section contains a full list of variations and clarifications points in the Microsoft implementation of  [[DOM Level 3 - Core]](http://go.microsoft.com/fwlink/?LinkId=182717).

Section[2.1](#page7)includesonly those variations that violate a MUST requirement in the targetspecification.

Section[2.2](#page29)describesfurther variations from MAY and SHOULD requirements.Section[2.3](#page31)identifiesvariations in error handling.

Section[2.4](#page31)identifiesvariations that impact security.

**2.1 Normative Variations**

The following subsections detail the normative variations from MUST requirements in  [[DOM Level 3](http://go.microsoft.com/fwlink/?LinkId=182717) -  [Core]](http://go.microsoft.com/fwlink/?LinkId=182717).

**2.1.1 [DOM Level 3 - Core] Section 1.3.1, String Comparisons in the DOM**

V0001:

The specification states:

The character normalization, i.e. transforming into their fully normalized form as as defined in [XML 1.1], is assumed to happen at serialization time. The DOM Level 3 Load and Save module [DOM Level 3 - LS] provides a serialization mechanism (see the DOMSerializer interface, section 2.3.1) and uses the DOMConfiguration parameters "normalize-characters" and "check-character-normalization" to assure that text is fully normalized [XML 1.1]. Other serialization mechanisms built on top of the DOM Level 3 Core also have to assure that text is fully normalized.

*Quirks Mode, IE7 Mode, and IE8 Mode (All Versions)*

The DOM Level 3 Load and Save module  [[DOM Level 3 - LS]](http://go.microsoft.com/fwlink/?LinkId=182721) is not supported; therefore, this type of character normalization is not supported.

**2.1.2 [DOM Level 3 - Core] Section 1.3.3, XML Namespaces**

V0002:

The specification states:

As far as the DOM is concerned, special attributes used for declaring XML namespaces are still exposed and can be manipulated just like any other attribute. However, nodes are permanently bound to namespace URIs as they get created. Consequently, moving a node within a document, using the DOM, in no case results in a change of its namespace prefix or namespace URI. Similarly, creating a node with a namespace prefix and namespace URI, or changing the namespace prefix of a node, does not result in any addition, removal, or modification of any special attributes for declaring the appropriate XML namespaces. Namespace validation is not enforced; the DOM application is responsible. In particular, since the mapping between prefixes and namespace URIs is not enforced, in general, the resulting document cannot be serialized naively. For example, applications may have to declare every

*7 / 35*

*[MS-DOM3C] — v20120725*

*Internet Explorer Document Object Model (DOM) Level 3 Core Standards Support Document*

*Copyright © 2012 Microsoft Corporation.*

*Release: July 25, 2012*

namespace in use when serializing a document.

*Quirks Mode, IE7 Mode, IE8 Mode, and IE9 Mode (All Versions)*

Special attributes for declaring namespaces that begin with "xmlns:" on the root <html> element are not supported. These attributes do not affect the namespace URI of created elements.

**2.1.3 [DOM Level 3 - Core] Section 1.4, Fundamental Interfaces: Core Module**

V0003:

The specification defines the **DOMException**.

*Quirks Mode, IE7 Mode, and IE8 Mode (All Versions)*

The **DOMException** exception is not supported. JavaScript error objects are created when exceptions occur.

V0004:

The specification defines the **ExceptionCode** exception of the **DOMException.**

*Quirks Mode, IE7 Mode, and IE8 Mode (All Versions)*

The **ExceptionCode** exception is not supported. V0005:

The specification defines the **DOMStringList** interface.

*Quirks Mode, IE7 Mode, IE8 Mode, and IE9 Mode (All Versions)*

The **DOMStringList** interface is not supported. V0006:

The specification defines the **NameList** interface.

*All Document Modes (All Versions)*

The **NameList** interface is not supported. V0007:

The specification defines the **DOMImplementationList** interface.

*All Document Modes (All Versions)*

The **DOMImplementationList** interface is not supported. V0008:

The specification defines the **DOMImplementationSource** interface.

*All Document Modes (All Versions)*

The **DOMImplementationSource** interface is not supported.

*8 / 35*

*[MS-DOM3C] — v20120725*

*Internet Explorer Document Object Model (DOM) Level 3 Core Standards Support Document*

*Copyright © 2012 Microsoft Corporation.*

*Release: July 25, 2012*

V0009:

The specification states:

Interface DOMImplementation

The DOMImplementation interface provides a number of methods for performing operations that are independent of any particular instance of the document object model.

*Quirks Mode, IE7 Mode, and IE8 Mode (All Versions)*

The following methods of the **DOMImplementation** interface are not supported and return a DOMException of **NOT\_SUPPORTED\_ERR**:

**createDocumentType**

**createDocument**

*All Document Modes (All Versions)*

The **getFeature** method of the **DOMImplementation** interface is not supported and returns a DOMException of **NOT\_SUPPORTED\_ERR**.

V0057:

The specification defines the **getFeature** method of the **Document** object.

*All Document Modes (All Versions)*

The **getFeature** method of the **DOMImplementation** interface is not supported and returns a DOMException of **NOT\_SUPPORTED\_ERR**.

V0010:

The specification states:

Interface DocumentFragment

DocumentFragment is a "lightweight" or "minimal" Document object. It is very common to want to be able to extract a portion of a document's tree or to create a new fragment of a document.

Furthermore, various operations -- such as inserting nodes as children of another Node -- may take DocumentFragment objects as arguments; this results in all the child nodes of the DocumentFragment being moved to the child list of this node. The children of a DocumentFragment node are zero or more nodes representing the tops of any sub-trees defining the structure of the document. DocumentFragment nodes do not need to be well-formed XML documents (although they do need to follow

the rules imposed upon well-formed XML parsed entities, which can have multiple top nodes).

When a DocumentFragment is inserted into a Document (or indeed any other Node that may take children) the children of the DocumentFragment and not the DocumentFragment itself are inserted into the Node. This makes the DocumentFragment very useful when the user wishes to create nodes that are siblings; the DocumentFragment acts as the parent of these nodes so that the user can use the standard methods from the Node interface, such as Node.insertBefore and

*9 / 35*

*[MS-DOM3C] — v20120725*

*Internet Explorer Document Object Model (DOM) Level 3 Core Standards Support Document*

*Copyright © 2012 Microsoft Corporation.*

*Release: July 25, 2012*

Node.appendChild.

*Quirks Mode, IE7 Mode, and IE8 Mode (All Versions)*

The **DocumentFragment** interface inherits from the **Document** interface and has all of the methods and properties it defines.

V0011:

The specification defines the **Document** interface.

*Quirks Mode, IE7 Mode, and IE8 Mode (All Versions)*

The following attributes of the **Document** interface are not supported:

**inputEncoding**

**xmlEncoding**

**xmlStandalone**

**xmlVersion**

The following methods of the **Document** interface are not supported:

**adoptNode**

**createAttributeNS**

**createCDATASection**

**createElementNS**

**createProcessingInstruction**

**getElementsByTagNameNS**

**importNode**

*All Document Modes (All Versions)*

The following attributes of the **Document** interface are not supported:

**documentURI**

**domConfig**

**strictErrorChecking**

The following methods of the **Document** interface are not supported:

**createEntityReference**

**normalizeDocument**

**renameNode**

V0012:

*10 / 35*

*[MS-DOM3C] — v20120725*

*Internet Explorer Document Object Model (DOM) Level 3 Core Standards Support Document*

*Copyright © 2012 Microsoft Corporation.*

*Release: July 25, 2012*

The specification states:

Interface DOMImplementation

The DOMImplementation interface provides a number of methods for performing operations that are independent of any particular instance of the document object model.

*Quirks Mode, IE7 Mode, and IE8 Mode (All Versions)*

The **doctype** attribute of the **Document** interface is always NULL because the **DocumentType** interface is not supported.

V0013:

The specification defines the **DocumentURI** interface.

*All Document Modes (All Versions)*

The **DocumentURI** interface is not supported. V0014:

The specification states:

getElementById introduced in DOM Level 2

Returns the Element that has an ID attribute with the given value. If no such element exists, this returns null. If more than one element has an ID attribute with that value, what is returned is undefined. The DOM implementation is expected to use the attribute Attr.isId to determine if an attribute is of type ID.

Note: Attributes with the name "ID" or "id" are not of type ID unless so defined.

Parameters

elementId of type DOMString

The unique id value for an element.

Return Value

Element The matching element or null if there is none.

No Exceptions

*Quirks Mode and IE7 Mode (All Versions)*

The **getElementById** method of the **Document** interface performs a case-insensitive compare against the IDs of elements and searches "name" attributes in addition to "id" attributes.

V0072:

The specification states:

createAttribute

Creates an Attr of the given name. Note that the Attr instance can then be set on an Element using the setAttributeNode method.

*11 / 35*

*[MS-DOM3C] — v20120725*

*Internet Explorer Document Object Model (DOM) Level 3 Core Standards Support Document*

*Copyright © 2012 Microsoft Corporation.*

*Release: July 25, 2012*

To create an attribute with a qualified name and namespace URI, use the createAttributeNS method.

Parameters

name of type DOMString The name of the attribute.

Return Value

Attr A new Attr object with the nodeName attribute set to name, and localName, prefix, and namespaceURI set to null. The value of the attribute is the empty string.

Exceptions

DOMException INVALID\_CHARACTER\_ERR: Raised if the specified name is not an XML name according to the XML version in use specified in the Document.xmlVersion attribute.

*Quirks Mode, IE7 Mode, and IE8 Mode (Internet Explorer 7, Internet Explorer 8)*

The created **Attr** instance has a **nodeValue** value of "undefined" instead of an empty string.

V0015:

The specification states:

createDocumentFragment

Creates an empty DocumentFragment object.

Return Value

DocumentFragment A new DocumentFragment

No Parameters

No Exceptions

*Quirks Mode, IE7 Mode, and IE8 Mode (All Versions)*

A full document-derived object is returned when the **createDocumentFragment** method is called. The **createDocumentFragment** method returns a node-derived interface and inherits all the methods of the Document interface in addition to those on the **Node** interface. The **nodeType** and **nodeValue** attributes are correct on the returned object.

V0016:

The specification states:

createElement

Creates an element of the type specified. Note that the instance returned implements the Element interface, so attributes can be specified directly on the returned object.In addition, if there are known attributes with default values, Attr nodes representing them are automatically created and attached to the element.To create an element with a qualified name and namespace URI, use the createElementNS method.

Parameters

*12 / 35*

*[MS-DOM3C] — v20120725*

*Internet Explorer Document Object Model (DOM) Level 3 Core Standards Support Document*

*Copyright © 2012 Microsoft Corporation.*

*Release: July 25, 2012*

tagName of type DOMString

The name of the element type to instantiate. For XML, this is case-sensitive, otherwise it depends on the case-sensitivity of the markup language in use. In that case, the name is mapped to the canonical form of that markup by the DOM implementation.

Return Value Element

A new Element object with the nodeName attribute set to tagName, and localName, prefix, and namespaceURI set to null.

Exceptions

DOMException

INVALID\_CHARACTER\_ERR: Raised if the specified name is not an XML name according to the XML version in use specified in the Document.xmlVersion attribute.

*Quirks Mode, IE7 Mode, and IE8 Mode (All Versions)*

The **createElement** method is overloaded with one that takes no parameters. When no parameters are given, this method returns an element with a **tagName** of null.

The **createElement** method accepts full element declaration strings that contain otherwise invalid characters for the **tagName** parameter. A parameter string such as "<div id='div1'>" would return a **div** element with an **id** of div1. An INVALID\_CHARACTER\_ERR exception is not raised in this case.

*Quirks Mode, IE7 Mode, IE8 Mode, and IE9 Mode (All Versions)*

When an element that contains an XMLNS declaration, such as <html XMLNS:mns='http://www.contoso.com'>, is specified for the **tagName** parameter, the value of the **tagUrn** property for the new element is set to the specified URI.

V0017:

The specification defines the **Node** interface.

*Quirks Mode, IE7 Mode, and IE8 Mode (All Versions)*

The following definition groups of the **Node** interface are not supported:

DocumentPosition

NodeType

With **NodeType** definition group values, the actual document types in markup are created as instances of the **Comment** interface, causing the values of the instance to match those of the

**Comment** entry.

The following attributes of the **Node** interface are not supported:

**baseURI**

**localName**

**namespaceURI**

**prefix**

*13 / 35*

*[MS-DOM3C] — v20120725*

*Internet Explorer Document Object Model (DOM) Level 3 Core Standards Support Document*

*Copyright © 2012 Microsoft Corporation.*

*Release: July 25, 2012*

**textContent**

The following methods of the **Node** interface are not supported:

**compareDocumentPosition**

**getFeature**

**getUserData**

**isDefaultNamespace**

**isEqualNode**

**isSameNode**

**isSupported**

**lookupNamespaceURI**

**lookupPrefix**

**setUserData**

*All Document Modes (All Versions)*

The **baseURI** attribute of the **Node** interface is not supported.

The following methods of the **Node** interface are not supported:

**getFeature**

**getUserData**

**setUserData**

V0018:

The specification states:

Attributes

childNodes of type NodeList, readonly

A NodeList that contains all children of this node. If there are no children, this is a NodeList containing no nodes.

*Internet Explorer 8 (All Modes)*

Splitting multiple text nodes under an element using the **splitText** method can cause the **childNodes** collection to not be immediately updated. Other tree modifications cause the **childNodes** collection to resynchronize.

V0019:

The specification states:

Attributes

*14 / 35*

*[MS-DOM3C] — v20120725*

*Internet Explorer Document Object Model (DOM) Level 3 Core Standards Support Document*

*Copyright © 2012 Microsoft Corporation.*

*Release: July 25, 2012*

nodeName of type DOMString, readonlyThe name of this node, depending on its type; see the table above.

*Quirks Mode, IE7 Mode, and IE8 Mode (All Versions)*

The **nodeName** attribute of the **Node** interface returns uppercase values except for elements with names that resemble namespaces (such as <test:elementName>) when a proprietary namespace has been declared. In this case, the **nodeName** attribute drops the element prefixes and does not return uppercase values.

V0020:

The specification states:

Attributes

parentNode of type Node, readonly The parent of this node. All nodes, except Attr, Document, DocumentFragment, Entity, and Notation may have a parent. However, if a node has just been created and not yet added to the tree, or if it has been removed from the tree, this is null.

*Quirks Mode, IE7 Mode, and IE8 Mode (All Versions)*

With the **parentNode** attribute of the **Node** interface, when a parentless element has child nodes, a document fragment is created and set as the parent of that element.

V0021:

The specification states:

Method

appendChild modified in DOM Level 3 Adds the node newChild to the end of the list of children of this node. If the newChild is already in the tree, it is first removed.

Parameters

newChild of type Node The node to add. If it is a DocumentFragment object, the entire contents of the document fragment are moved into the child list of this node

Return Value

Node The node added.

Exceptions

DOMException

HIERARCHY\_REQUEST\_ERR: Raised if this node is of a type that does not allow children of the type of the newChild node, or if the node to append is one of this node's ancestors or this node itself, or if this node is of type Document and the DOM application attempts to append a second DocumentType or Element node.

*15 / 35*

*[MS-DOM3C] — v20120725*

*Internet Explorer Document Object Model (DOM) Level 3 Core Standards Support Document*

*Copyright © 2012 Microsoft Corporation.*

*Release: July 25, 2012*

WRONG\_DOCUMENT\_ERR: Raised if newChild was created from a different document than the one that created this node.

NO\_MODIFICATION\_ALLOWED\_ERR: Raised if this node is readonly or if the previous parent of the node being inserted is readonly.

NOT\_SUPPORTED\_ERR: if the newChild node is a child of the Document node, this exception might be raised if the DOM implementation doesn't support the removal of the DocumentType child or Element child.

*Quirks Mode, IE7 Mode, and IE8 Mode (All Versions)*

With the **appendChild** method of the **Node** interface, the conditions that trigger the HIERARCHY\_REQUEST\_ERR and WRONG\_DOCUMENT\_ERR errors both result in a JScript "Invalid argument" error (HRESULT 0x80070057).

The following elements trigger an exception when trying to dynamically insert or append new nodes:

**APPLET****AREA****BASE****BGSOUND**

**BR**

**COL****COMMENT****EMBED****FRAME**

**HR**

**IFRAME****IMG****INPUT****ISINDEX****LINK****META****NEXTID****NOEMBED****NOFRAMES****NOSCRIPT****OBJECT**

*16 / 35*

*[MS-DOM3C] — v20120725*

*Internet Explorer Document Object Model (DOM) Level 3 Core Standards Support Document*

*Copyright © 2012 Microsoft Corporation.*

*Release: July 25, 2012*

**PARAM**

**SCRIPT**

**STYLE**

**WBR**

*IE9 Mode and IE10 Mode (All Versions)*

With the **appendChild** method of the **Node** interface, the conditions that trigger the WRONG\_DOCUMENT\_ERR error cause the node to automatically be adopted and inserted. No exception is thrown.

V0022:

The specification states:

Method

cloneNode Returns a duplicate of this node, i.e., serves as a generic copy constructor for nodes.

Cloning an Attr directly, as opposed to be cloned as part of an Element cloning operation, returns a specified attribute (specified is true). Cloning an Attr always clones its children, since they represent its value, no matter whether this is a deep clone or not. In addition, clones of unspecified Attr nodes are specified. And, cloning Document, DocumentType, Entity, and Notation nodes is implementation dependent.

*Quirks Mode, IE7 Mode, and IE8 Mode (All Versions)*

With the **cloneNode** method of the **Node** interface, cloned **Attr** objects do not have the specified attribute set to true. Cloned **Attr** objects have their **nodeValue** attributes set to undefined.

V0023:

The specification states:

Methods

insertBefore modified in DOM Level 3

Inserts the node newChild before the existing child node refChild. If refChild is null, insert newChild at the end of the list of children.

If newChild is a DocumentFragment object, all of its children are inserted, in the same order, before refChild. If the newChild is already in the tree, it is first removed.

Note: Inserting a node before itself is implementation dependent.

Parameters

newChild of type Node The node to insert.

refChild of type Node

*17 / 35*

*[MS-DOM3C] — v20120725*

*Internet Explorer Document Object Model (DOM) Level 3 Core Standards Support Document*

*Copyright © 2012 Microsoft Corporation.*

*Release: July 25, 2012*

The reference node, i.e., the node before which the new node must be inserted.

Return Value

Node

The node being inserted.

Exceptions

DOMException

HIERARCHY\_REQUEST\_ERR: Raised if this node is of a type that does not allow children of the type of the newChild node, or if the node to insert is one of this node's ancestors or this node itself, or if this node is of type Document and the DOM application attempts to insert a second DocumentType or Element node.

WRONG\_DOCUMENT\_ERR: Raised if newChild was created from a different document than the one that created this node.

NO\_MODIFICATION\_ALLOWED\_ERR: Raised if this node is readonly or if the parent of the node being inserted is readonly.

NOT\_FOUND\_ERR: Raised if refChild is not a child of this node.

NOT\_SUPPORTED\_ERR: if this node is of type Document, this exception might be raised if the DOM implementation doesn't support the insertion of a DocumentType or Element node.

*Quirks Mode, IE7 Mode, and IE8 Mode (All Versions)*

With the **insertBefore** method of the **Node** interface, the conditions to trigger the HIERARCHY\_REQUEST\_ERR and WRONG\_DOCUMENT\_ERR errors both result in an "Invalid argument" error (HRESULT 0x80070057).

The following elements cause an exception when trying to dynamically insert or append new nodes:

**APPLET**

**AREA**

**BASE**

**BGSOUND**

**BR**

**COL**

**COMMENT**

**EMBED**

**FRAME**

**HR**

**IFRAME**

*18 / 35*

*[MS-DOM3C] — v20120725*

*Internet Explorer Document Object Model (DOM) Level 3 Core Standards Support Document*

*Copyright © 2012 Microsoft Corporation.*

*Release: July 25, 2012*

**IMG****INPUT****ISINDEX****LINK****META****NEXTID****NOEMBED****NOFRAMES****NOSCRIPT****OBJECT****PARAM****SCRIPT****STYLE****WBR**

*IE9 Mode and IE10 Mode (All Versions)*

With the **insertBefore** method of the **Node** interface, the conditions that trigger the WRONG\_DOCUMENT\_ERR error cause the node to automatically be adopted and inserted. No exception is thrown.

V0024:

The specification states:

normalize modified in DOM Level 3

Puts all Text nodes in the full depth of the sub-tree underneath this Node, including attribute nodes, into a "normal" form where only structure (e.g., elements, comments, processing instructions, CDATA sections, and entity references) separates Text nodes, i.e., there are neither adjacent Text nodes nor empty Text nodes. This can be used to ensure that the DOM view of a document is the same as if it were saved and re-loaded, and is useful when operations (such as XPointer [XPointer] lookups) that depend on a particular document tree structure are to be used. If the parameter "normalize-characters" of the DOMConfiguration object attached to the Node.ownerDocument is true, this method will also fully normalize

the characters of the Text nodes. Note: In cases where the document contains CDATASections, the normalize operation alone may not be sufficient, since XPointers

do not differentiate between Text nodes and CDATASection nodes. No Parameters

No Return Value

No Exceptions

*Quirks Mode, IE7 Mode, and IE8 Mode (All Versions)*

*19 / 35*

*[MS-DOM3C] — v20120725*

*Internet Explorer Document Object Model (DOM) Level 3 Core Standards Support Document*

*Copyright © 2012 Microsoft Corporation.*

*Release: July 25, 2012*

The **normalize** method of the **Node** interface does not collapse an empty text node into an adjacent text node. An empty text node is not removed if it is the only child of its parent.

V0025:

The specification states:

Method

removeChild modified in DOM Level 3 Removes the child node indicated by oldChild from the list of children, and returns it.

|  |  |  |
| --- | --- | --- |
| Parameters |  |  |
| oldChild of type Node | | The node being removed. |
| Return Value Node | The | node removed. |

Exceptions

DOMException NO\_MODIFICATION\_ALLOWED\_ERR: Raised if this node is readonly. NOT\_FOUND\_ERR: Raised if oldChild is not a child of this node.

NOT\_SUPPORTED\_ERR: if this node is of type Document, this exception might be raised if the DOM implementation doesn't support the removal of the DocumentType child or the Element child.

*Quirks Mode, IE7 Mode, and IE8 Mode (All Versions)*

With the **removeChild** method of the **Node** interface, the conditions that trigger the NOT\_FOUND\_ERR error result in a JavaScript "Invalid argument" error (HRESULT 0x80070057).

V0026:

The specification states:

Method

replaceChild modified in DOM Level 3

Replaces the child node oldChild with newChild in the list of children, and returns the oldChild node.If newChild is a DocumentFragment object, oldChild is replaced by all of the DocumentFragment children, which are inserted in the same order. If the newChild is already in the tree, it is first removed.Note: Replacing a node with itself is implementation dependent.

|  |  |  |
| --- | --- | --- |
| Parameters |  |  |
| newChild of type Node | | The new node to put in the child list. |
| oldChild of type Node | | The node being replaced in the list. |
| Return Value Node | The | node replaced. |

Exceptions

DOMException

HIERARCHY\_REQUEST\_ERR: Raised if this node is of a type that does not allow children of the type of the newChild node, or if the node to put in is one of this node's ancestors or this node itself, or if this node is of type Document and the result of the replacement operation would add a second DocumentType or Element on the Document node.

WRONG\_DOCUMENT\_ERR: Raised if newChild was created from a different document than the one that created this node.

NO\_MODIFICATION\_ALLOWED\_ERR: Raised if this node or the parent of the new node is readonly.

*20 / 35*

*[MS-DOM3C] — v20120725*

*Internet Explorer Document Object Model (DOM) Level 3 Core Standards Support Document*

*Copyright © 2012 Microsoft Corporation.*

*Release: July 25, 2012*

NOT\_FOUND\_ERR: Raised if oldChild is not a child of this node.

NOT\_SUPPORTED\_ERR: if this node is of type Document, this exception might be raised if the DOM implementation doesn't support the replacement of the DocumentType child or Element child.

*Quirks Mode, IE7 Mode, and IE8 Mode (All Versions)*

With the **replaceChild** method of the **Node** interface, the conditions that trigger the HIERARCHY\_REQUEST\_ERR, WRONG\_DOCUMENT\_ERR, and NOT\_FOUND\_ERR errors result in JavaScript "Invalid argument" errors (HRESULT 0x80070057).

V0027:

The specification defines the **NamedNodeMap** interface.

*Quirks Mode, IE7 Mode, and IE8 Mode (All Versions)*

The following methods are not supported by the **NamedNodeMap** interface:

**getNamedItemNS**

**removeNamedItemNS**

**setNamedItemNS**

V0028:

The specification states:

Method

item Returns the indexth item in the map. If index is greater than or equal to the number of nodes in this map, this returns null.

Parameters

index of type unsigned longIndex into this map.

Return Value

Node The node at the indexth position in the map, or null if that is not a valid index.

No Exceptions

*Quirks Mode and IE7 Mode (All Versions)*

With the **item** method of the **NamedNodeMap** interface, NULL is not returned when the index is greater than the number of nodes in the map, but instead results in an HRESULT 0x80070057, which creates a JavaScript Error message "Invalid argument."

V0029:

The specification states:

Method

removeNamedItem Removes a node specified by name. When this map contains the attributes attached to an element, if the removed attribute is known to have a default value, an attribute immediately appears containing the default value as

*21 / 35*

*[MS-DOM3C] — v20120725*

*Internet Explorer Document Object Model (DOM) Level 3 Core Standards Support Document*

*Copyright © 2012 Microsoft Corporation.*

*Release: July 25, 2012*

well as the corresponding namespace URI, local name, and prefix when applicable.

Parameters

name of type DOMStringThe nodeName of the node to remove.

Return Value

Node The node removed from this map if a node with such a name exists.

Exceptions

DOMException NOT\_FOUND\_ERR: Raised if there is no node named name in this map. NO\_MODIFICATION\_ALLOWED\_ERR: Raised if this map is readonly.

*Quirks Mode and IE7 Mode (All Versions)*

With the **removeNamedItem** method of the **NamedNodeMap** interface, NULL is returned instead of the NOT\_FOUND\_ERR exception.

V0030:

The specification states:

Method

setNamedItem Adds a node using its nodeName attribute. If a node with that name is already present in this map, it is replaced by the new one. Replacing a node by itself has no effect.As the nodeName attribute is used to derive the name which the node must be stored under, multiple nodes of certain types (those that have a "special" string value) cannot be stored as the names would clash. This is seen as preferable to allowing nodes to be aliased.

Parameters

arg of type Node A node to store in this map. The node will later be accessible using the value of its nodeName attribute.

Return Value

Node If the new Node replaces an existing node the replaced Node is returned, otherwise null is returned.

Exceptions

DOMException

WRONG\_DOCUMENT\_ERR: Raised if arg was created from a different document than the one that created this map.

NO\_MODIFICATION\_ALLOWED\_ERR: Raised if this map is readonly. INUSE\_ATTRIBUTE\_ERR: Raised if arg is an Attr that is already an attribute of another Element object. The DOM user must explicitly clone Attr nodes to re-use them in other elements.

HIERARCHY\_REQUEST\_ERR: Raised if an attempt is made to add a node doesn't belong in this NamedNodeMap. Examples would include trying to insert something other than an Attr node into an Element's map of attributes, or a non-Entity node into the DocumentType's map of Entities.

*Quirks Mode, IE7 Mode, and IE8 Mode (All Versions)*

With the **setNamedItem** method of the **NamedNodeMap** interface, an "Invalid Argument" exception is raised instead of INUSE\_ATTRIBUTE\_ERR. No exception is raised when the argument was created from a different document or when it is not an attribute.

V0031:

*22 / 35*

*[MS-DOM3C] — v20120725*

*Internet Explorer Document Object Model (DOM) Level 3 Core Standards Support Document*

*Copyright © 2012 Microsoft Corporation.*

*Release: July 25, 2012*

The specification states:

Method

substringData Extracts a range of data from the node.

Parameters

offset of type unsigned longStart offset of substring to extract. count of type unsigned longThe number of 16-bit units to extract.

Return Value DOMString The specified substring. If the sum of offset and count exceeds the

length, then all 16-bit units to the end of the data are returned.

Exceptions

DOMException

INDEX\_SIZE\_ERR: Raised if the specified offset is negative or greater than the number of 16-bit units in data, or if the specified count is negative. DOMSTRING\_SIZE\_ERR: Raised if the specified range of text does not fit into a DOMString.

*Quirks Mode, IE7 Mode, and IE8 Mode (All Versions)*

With the **substringData** method of the **CharacterData** interface, an exception is not thrown if the offset is greater than the number of 16-bit units in the data. An "Invalid argument" error is returned instead of a DOMException.

V0032:

The specification states:

IDL Definition

interface Attr : Node {

readonly attribute DOMString name;

readonly attribute boolean specified;

attribute DOMString value;

// raises(DOMException) on setting

// Introduced in DOM Level 2:

readonly attribute Element ownerElement;

// Introduced in DOM Level 3:

readonly attribute TypeInfo schemaTypeInfo;

// Introduced in DOM Level 3:

readonly attribute boolean isId;

};

*All Document Modes (All Versions)*

The following attributes of the **Attr** interface are not supported:

**schemaTypeInfo**

**isId**

V0033:

The specification defines the **Element** interface.

*23 / 35*

*[MS-DOM3C] — v20120725*

*Internet Explorer Document Object Model (DOM) Level 3 Core Standards Support Document*

*Copyright © 2012 Microsoft Corporation.*

*Release: July 25, 2012*

*Quirks Mode, IE7 Mode, and IE8 Mode (All Versions)*

The following methods of the **Element** interface are not supported:

**getAttributeNodeNS****getAttributeNS****hasAttributeNS****removeAttributeNS****schemaTypeInfo****setAttributeNodesNS****setAttributeNS****setIdAttribute****setIdAttributeNode****setIdAttributeNS**

Attribute subtrees of the **Attr** interface are not supported. Only strings are supported for the **Attr** object.

*IE9 Mode and IE10 Mode (All Versions)*

For any given attribute, only a single, immutable text node is supported as the subtree. V0034:

The specification states:

tagName of type DOMString, readonly

The name of the element. If Node.localName is different from null, this attribute is a qualified name. For example, in:

<elementExample id="demo">

...

</elementExample> ,

tagName has the value "elementExample". Note that this is case-preserving in XML, as are all of the operations of the DOM. The HTML DOM returns the tagName of an HTML element in the canonical uppercase form, regardless of the case in the source HTML document.

*Quirks Mode, IE7 Mode, and IE8 Mode (All Versions)*

The **tagName** attribute of the **Element** interface returns uppercase values except for elements with names that resemble namespaces (such as <test:elementName>) when a proprietary namespace has been declared because XML support is provided through MSXML. In this case, the **tagName** attribute drops the element prefixes and does not return uppercase values.

V0035:

The specification states:

*24 / 35*

*[MS-DOM3C] — v20120725*

*Internet Explorer Document Object Model (DOM) Level 3 Core Standards Support Document*

*Copyright © 2012 Microsoft Corporation.*

*Release: July 25, 2012*

Method

getElementsByTagName Returns a NodeList of all descendant Elements with a given tag name, in document order.

Parameters

name of type DOMString The name of the tag to match on. The special value "\*" matches all tags.

Return Value

NodeList A list of matching Element nodes.

No Exceptions

*Quirks Mode, IE7 Mode, and IE8 Mode (All Versions)*

With the **getElementsByTagName** method of the **Document** interface, if **object1.getElementsByTagName**("\*") is called, an empty collection is returned. If **object1.getElementsByTagName**("param") is called, a collection containing all parameters in thedocument is returned, as if the call were **document.getElementsByTagName**("param").

V0036:

The specification states:

Method

removeAttribute Removes an attribute by name. If a default value for the removed attribute is defined in the DTD, a new attribute immediately appears with the default value as well as the corresponding namespace URI, local name, and prefix when applicable. The implementation may handle default values from other schemas similarly but applications should use Document.normalizeDocument() to guarantee this information is up-to-date.If no attribute with this name is found, this method has no effect.To remove an attribute by local name and namespace URI, use the removeAttributeNS method.

Parameters

name of type DOMString The name of the attribute to remove.

Exceptions

DOMException NO\_MODIFICATION\_ALLOWED\_ERR: Raised if this node is readonly. No Return Value

*Quirks Mode and IE7 Mode (All Versions)*

The **removeAttribute** method supports an additional parameter called **iCaseSensitive**. The **iCaseSensitive** parameter specifies whether to use a case-sensitive search to find the attribute.Removal of event handler attributes (such as **onClick**) or the style attribute does not cause the actual event handler to be removed, or the inline style to be removed.

Default attributes are not re-created after the attribute is removed.

*Quirks Mode, IE7 Mode, and IE8 Mode (All Versions)*

The **removeAttributeNS** method is not supported.

*All Document Modes (All Versions)*

*25 / 35*

*[MS-DOM3C] — v20120725*

*Internet Explorer Document Object Model (DOM) Level 3 Core Standards Support Document*

*Copyright © 2012 Microsoft Corporation.*

*Release: July 25, 2012*

The **removeAttribute** method of the **Element** interface exists on **CSSStyleDeclaration** objects that are used for inline styles, runtime styles, and style rules.

The **removeAttribute** method returns a Boolean value that indicates whether the operation has succeeded or failed. The **removeAttribute** method is also available on the following objects: **element.currentStyle**, **element.runtimeStyle**, **element.style**, and **stylesheet.style**.

V0037:

The specification states:

Method

removeAttributeNode Removes the specified attribute node. If a default value for the removed Attr node is defined in the DTD, a new node immediately appears with the default value as well as the corresponding namespace URI, local name, and prefix when applicable. The implementation may handle default values from other schemas similarly but applications should use Document.normalizeDocument() to guarantee this information is up-to-date.

Parameters

oldAttr of type Attr The Attr node to remove from the attribute list.

Return Value

Attr The Attr node that was removed.

Exceptions

DOMExceptionNO\_MODIFICATION\_ALLOWED\_ERR: Raised if this node is readonly.NOT\_FOUND\_ERR: Raised if oldAttr is not an attribute of the element.

*Quirks Mode and IE7 Mode (All Versions)*

With the **removeAttributeNode** method of the **Element** interface, default attributes are not re-created after the attribute is removed. Removal of event handler attributes (such as **onClick**) or the style attribute does not cause the actual event handler to be removed, or the inline style to be removed.

V0038:

The specification states:

Method

setAttribute Adds a new attribute. If an attribute with that name is already present in the element, its value is changed to be that of the value parameter. This value is a simple string; it is not parsed as it is being set. So any markup (such as syntax to be recognized as an entity reference) is treated as literal text, and needs to be appropriately escaped by the implementation when it is written out. In order to assign an attribute value that contains entity references, the user must create an Attr node plus any Text and EntityReference nodes, build the appropriate subtree, and use setAttributeNode to assign it as the value of an attribute.To set an attribute with a qualified name and namespace URI, use the setAttributeNS method.

Parameters

name of type DOMString The name of the attribute to create or alter.value of type DOMStringValue to set in string form.

Exceptions

*26 / 35*

*[MS-DOM3C] — v20120725*

*Internet Explorer Document Object Model (DOM) Level 3 Core Standards Support Document*

*Copyright © 2012 Microsoft Corporation.*

*Release: July 25, 2012*

DOMException INVALID\_CHARACTER\_ERR: Raised if the specified name is not an XML name according to the XML version in use specified in the Document.xmlVersion attribute.

NO\_MODIFICATION\_ALLOWED\_ERR: Raised if this node is readonly.

No Return Value

*Quirks Mode and IE7 Mode (All Versions)*

The **setAttribute** method assigns attributes in a case-sensitive manner (expected case-insensitive for HTML).

The second parameter of the **setAttribute** method accepts only strings, not objects. An optional third parameter controls case sensitivity.

Attributes that apply a boolean initial state to the associated DOM properties (for example, **value** and **checked**) are incorrectly associated with their live property rather than their default property. For example, the **setAttribute** method ('checked', 'checked') toggles the DOM **checked** property (the live view of a check box) rather than the **defaultChecked** property (initial value).

The HTML **style** attribute and attributes that are event handlers do not apply their conditions when used with **setAttribute**.

The **setAttribute** method requires DOM property names to apply effects for certain attribute names, such as **className** (instead of 'class'), **htmlFor** (instead of 'for'), and **httpEquiv** (instead of 'http-equiv').

V0039:

The specification defines the **Text** interface.

*Quirks Mode, IE7 Mode, and IE8 Mode (All Versions)*

The following attribute and method of the **Text** interface are not supported:

**wholeText**attribute

**replaceWholeText**method

*All Document Modes (All Versions)*

The **isElementContentWhitespace** attribute of the **Text** interface is not supported.

V0040:

The specification states:

splitText

Breaks this node into two nodes at the specified offset, keeping both in the tree as siblings. After being split, this node will contain all the content up to the offset point. A new node of the same type, which contains all the content at and after the offset point, is returned. If the original node had a parent node, the new node is inserted as the next sibling of the original node. When the offset is equal to the length of this node, the new node has no data.

Parameters

offset of type unsigned long

*27 / 35*

*[MS-DOM3C] — v20120725*

*Internet Explorer Document Object Model (DOM) Level 3 Core Standards Support Document*

*Copyright © 2012 Microsoft Corporation.*

*Release: July 25, 2012*

The 16-bit unit offset at which to split, starting from 0.

Return Value

Text

The new node, of the same type as this node.

Exceptions

DOMException

INDEX\_SIZE\_ERR: Raised if the specified offset is negative or greater than the number of 16-bit units in data.

NO\_MODIFICATION\_ALLOWED\_ERR: Raised if this node is readonly.

*Quirks Mode, IE7 Mode, and IE8 Mode (Internet Explorer 8)*

With the **splitText** method of the **Text** interface, any time there is a modification to the markup, a cache of invalidated **childNodes** objects is maintained. Because calling the **splitText** method does not trigger a markup modification, the **childNodes** collection does not show changes made by **splitText** until the markup is modified (for example, by changing the text of a **DIV** elementanywhere on the page).

*Quirks Mode, IE7 Mode, and IE8 Mode (All Versions)*

The **splitText** method treats the argument as optional; if there is no unsigned long offset provided, then and offset of zero (0) will be used.

V0041:

The specification states:

Interface Comment

This interface inherits from CharacterData and represents the content of a comment, i.e., all the characters between the starting '<!--' and ending '-->'. Note that this is the definition of a comment in XML, and, in practice, HTML, although some HTML tools may implement the full SGML comment structure

*Quirks Mode, IE7 Mode, and IE8 Mode (All Versions)*

With the **Comment** interface, **comments** are derived from **ELEMENT** rather than from **Node**. V0042:

The specification defines the **TypeInfo** interface.

*All Document Modes (All Versions)*

The **TypeInfo** interface is not supported. V0043:

The specification defines the **UserDataHandler** interface

*All Document Modes (All Versions)*

The **UserDataHandler** interface is not supported. V0044:

*28 / 35*

*[MS-DOM3C] — v20120725*

*Internet Explorer Document Object Model (DOM) Level 3 Core Standards Support Document*

*Copyright © 2012 Microsoft Corporation.*

*Release: July 25, 2012*

The specification defines the **DOMError** interface.

*All Document Modes (All Versions)*

The **DOMError** interface is not supported.

V0045:

The specification defines the **DOMErrorHandler** interface.

*All Document Modes (All Versions)*

The **DOMErrorHandler** interface is not supported.

V0046:

The specification defines the **DOMLocator** interface.

*All Document Modes (All Versions)*

The **DOMLocator** interface is not supported.

V0047:

The specification defines the **DOMConfiguration** interface.

*All Document Modes (All Versions)*

The **DOMConfiguration** interface is not supported.

**2.1.4 [DOM Level 3 - Core] Section 1.5, Extended Interfaces: XML Module**

V0048:

The specification states:

In order to fully support this module, an implementation must also support the "Core" feature defined in Fundamental Interfaces: Core Module and the feature "XMLVersion" with version "1.0" defined inDocument.xmlVersion.

*All Document Modes (All Versions)*

The **XML Module** extended interface is not supported because Core Module methods such as **getUserData** are not implemented. Additionally, if the **hasFeature** method is run with **XMLVersion** equal to "1.0", a value of **false** is returned.

**2.2 Clarifications**

The following subsections identify clarifications to recommendations made by  [[DOM Level 3 - Core]](http://go.microsoft.com/fwlink/?LinkId=182717).

**2.2.1 [DOM Level 3 - Core] Section 1.4, Fundamental Interfaces: Core Module**

C0001:

The specification states:

Method

*29 / 35*

*[MS-DOM3C] — v20120725*

*Internet Explorer Document Object Model (DOM) Level 3 Core Standards Support Document*

*Copyright © 2012 Microsoft Corporation.*

*Release: July 25, 2012*

hasFeature Test if the DOM implementation implements a specific feature and version, as specified in DOM

*Quirks Mode, IE7 Mode, and IE8 Mode (All Versions)*

The **hasFeature** method of the **DOMImplementation** interface returns false for the Core module and the version strings "1.0", "2.0", and "3.0".

*IE9 Mode and IE10 Mode (All Versions)*

The **hasFeature** method of the **DOMImplementation** interface returns false for the Core module and the version string "3.0" because not all DOM L3 Core methods are supported.

C0002:

The specification states:

createTextNode

Creates a Text node given the specified string. Parameters

data of type DOMString The data for the node.

Return Value

Text

The new Text object.

*Quirks Mode, IE7 Mode, and IE8 Mode (All Versions)*

The **data** parameter of the **createTextNode** method of the **Document** interface is optional. A text node is created even when no parameter is provided.

C0003:

The specification states:

createComment

Creates a Comment node given the specified string.

*Quirks Mode, IE7 Mode, and IE8 Mode (All Versions)*

With the **createComment** method of the **Document** interface, the **data** parameter is treated as optional and creates a Comment node even when no parameter is provided.

C0004:

The specification states:

getAttribute Retrieves an attribute value by name.

Parameters name of type DOMStringThe name of the attribute to retrieve. Return Value DOMStringThe Attr value as a string, or the empty string if that attribute does not have a specified or default value.

No Exceptions

*30 / 35*

*[MS-DOM3C] — v20120725*

*Internet Explorer Document Object Model (DOM) Level 3 Core Standards Support Document*

*Copyright © 2012 Microsoft Corporation.*

*Release: July 25, 2012*

*Quirks Mode and IE7 Mode (All Versions)*

The **getAttribute** method of the **Element** interface supports an additional parameter that controls case-sensitivity and object extrapolation (such as for the **style** object).

C0005:

The specification states:

Method

getAttributeNode Retrieves an attribute node by name.To retrieve an attribute node by qualified name and namespace URI, use the getAttributeNodeNS method.

Parameters

name of type DOMString The name (nodeName) of the attribute to retrieve.

Return Value

Attr The Attr node with the specified name (nodeName) or null if there is no such attribute.

No Exceptions

*Quirks Mode and IE7 Mode (All Versions)*

The **getAttributeNode** method of the **Element** interface returns attribute nodes that have not been specified by the author or have default values other than **NULL**

**2.3 Error Handling**

There are no additional considerations for error handling.

**2.4 Security**

There are no additional security considerations.

*31 / 35*

*[MS-DOM3C] — v20120725*

*Internet Explorer Document Object Model (DOM) Level 3 Core Standards Support Document*

*Copyright © 2012 Microsoft Corporation.*

*Release: July 25, 2012*

**3 Change Tracking**

This section identifies changes that were made to the [MS-DOM3C] 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.

*32 / 35*

*[MS-DOM3C] — v20120725*

*Internet Explorer Document Object Model (DOM) Level 3 Core Standards Support Document*

*Copyright © 2012 Microsoft Corporation.*

*Release: July 25, 2012*

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](mailto:protocol@microsoft.com).

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

*33 / 35*

*[MS-DOM3C] — v20120725*

*Internet Explorer Document Object Model (DOM) Level 3 Core Standards Support Document*

*Copyright © 2012 Microsoft Corporation.*

*Release: July 25, 2012*

**4 Index**

**A**

Attributes  [baseURI](#page8) 8  [childNodes](#page8) 8  [doctype](#page8) 8  [domConfig](#page8) 8  [inputEncoding](#page8) 8  [isId](#page8) 8  [localName](#page8) 8

[namespaceURI](#page8) 8  [nodeName](#page8) 8  [parentNode](#page8) 8  [prefix](#page8) 8  [schemaTypeInfo](#page8) 8

[strictErrorChecking](#page8) 8  [tagName](#page8) 8  [wholeText](#page8) 8  [xmlEncoding](#page8) 8  [xmlStandalone](#page8) 8

xmlVersion ( [section 2.1.3](#page8) 8,  [section 2.1.4](#page29) 29)

**C**

[Change tracking](#page32) 32

**E**

[Extended Interfaces: XML Module](#page29) 29

**F**

Fundamental Interfaces: Core Module ( [section 2.1.](#page8)3

8,  [section 2.2.1](#page29) 29)

**G**

[Glossary](#page4) 4

**I**

[Informative references](#page4) 4 Interfaces

[Attr](#page8) 8  [CharacterData](#page8) 8  [Comment](#page8) 8

Document ( [section 2.1.3](#page8) 8,  [section 2.1.3](#page8) 8,  [section 2.1.3](#page8) 8,  [section 2.1.3](#page8) 8)

[DocumentFragment](#page8) 8  [DocumentType](#page8) 8  [DOMConfiguration](#page8) 8  [DOMError](#page8) 8  [DOMErrorHandler](#page8) 8

DOMImplementation ( [section 2.1.3](#page8) 8,  [sectio](#page8)n  [2.1.3](#page8) 8,  [section 2.1.3](#page8) 8,  [section 2.2.1](#page29) 29,  [section 2.2.1](#page29) 29)

[DOMImplementationList](#page8) 8  [DOMImplementationSource](#page8) 8

[DOMLocator](#page8) 8

[DOMStringList](#page8) 8

[Element](#page8) 8

[NamedNodeMap](#page8) 8

[NameList](#page8) 8

[Node](#page8) 8

[Text](#page8) 8

[TypeInfo](#page8) 8

[UserDataHandler](#page8) 8

[XML Module](#page29) 29

[Introduction](#page4) 4

**M**

Methods  [adoptNode](#page8) 8

appendChild ( [section 2.1.3](#page8) 8,  [section 2.1.3](#page8) 8)  [cloneNode](#page8) 8

[compareDocumentPosition](#page8) 8  [createAttributeNS](#page8) 8  [createCDATASection](#page8) 8  [createComment](#page29) 29  [createDocument](#page8) 8  [createDocumentFragment](#page8) 8  [createDocumentType](#page8) 8

createElementNS ( [section 2.1.3](#page8) 8,  [section 2.1.](#page8)3 8)

[createEntityReference](#page8) 8  [createProcessingInstruction](#page8) 8  [createTextNode](#page29) 29  [getAttribute](#page29) 29  [getAttributeNode](#page29) 29  [getAttributeNodeNS](#page8) 8  [getAttributeNS](#page8) 8  [getElementById](#page8) 8  [getElementsByTagName](#page8) 8  [getElementsByTagNameNS](#page8) 8

getFeature ( [section 2.1.3](#page8) 8,  [section 2.1.3](#page8) 8,  [section 2.1.3](#page8) 8)

[getNamedItemNS](#page8) 8

getUserData ( [section 2.1.3](#page8) 8,  [section 2.1.3](#page8) 8)  [hasAttributeNS](#page8) 8

hasFeature ( [section 2.2.1](#page29) 29,  [section 2.2.1](#page29) 29)  [importNode](#page8) 8

insertBefore ( [section 2.1.3](#page8) 8,  [section 2.1.3](#page8) 8)  [isDefaultNamespace](#page8) 8

[isEqualNode](#page8) 8  [isSameNode](#page8) 8  [isSupported](#page8) 8  [item](#page8) 8

[lookupNamespaceURI](#page8) 8

lookupPrefix ( [section 2.1.3](#page8) 8,  [section 2.1.3](#page8) 8)  [normalize](#page8) 8

[normalizeDocument](#page8) 8  [removeAttribute](#page8) 8  [removeAttributeNode](#page8) 8  [removeAttributeNS](#page8) 8  [removeChild](#page8) 8

*34 / 35*

*[MS-DOM3C] — v20120725*

*Internet Explorer Document Object Model (DOM) Level 3 Core Standards Support Document*

*Copyright © 2012 Microsoft Corporation.*

*Release: July 25, 2012*

[removeNamedItem](#page8) 8  [removeNamedItemNS](#page8) 8  [renameNode](#page8) 8  [replaceChild](#page8) 8  [replaceWholeText](#page8) 8  [schemaTypeInfo](#page8) 8

setAttribute ( [section 2.1.3](#page8) 8,  [section 2.1.3](#page8) 8)  [setAttributeNodesNS](#page8) 8

[setAttributeNS](#page8) 8  [setIdAttribute](#page8) 8  [setIdAttributeNode](#page8) 8  [setIdAttributeNS](#page8) 8  [setNamedItem](#page8) 8  [setNamedItemNS](#page8) 8

setUserData ( [section 2.1.3](#page8) 8,  [section 2.1.3](#page8) 8)  [splitText](#page8) 8

[substringData](#page8) 8

**N**

[Normative references](#page4) 4

**R**

References

[informative](#page4) 4

[normative](#page4) 4

**S**

[String Comparisons in the DOM](#page7) 7

**T**

[textContent](#page8) 8  [Tracking changes](#page32) 32

**X**

[XML Namespaces](#page7) 7

*35 / 35*

*[MS-DOM3C] — v20120725*

*Internet Explorer Document Object Model (DOM) Level 3 Core Standards Support Document*

*Copyright © 2012 Microsoft Corporation.*

*Release: July 25, 2012*

[PDF to Word](http://pdfonline.blogspot.com)