**[MS-DOM2CE]:**

**Internet Explorer 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](http://go.microsoft.com/fwlink/?LinkId=214445) or the [Microsoft Community Promise](http://go.microsoft.com/fwlink/?LinkId=214448). 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.
* **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](http://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.

**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 | 5.1 | None | No changes to the meaning, language, or formatting of the technical content. |
| 3/22/2016 | 5.2 | Minor | Clarified the meaning of the technical content. |
| 11/2/2016 | 5.2 | None | No changes to the meaning, language, or formatting of the technical content. |
| 3/14/2017 | 5.2 | None | No changes to the meaning, language, or formatting of the technical content. |

Table of Contents

[1 Introduction 4](#_Toc477342805)

[1.1 Glossary 4](#_Toc477342806)

[1.2 References 4](#_Toc477342807)

[1.2.1 Normative References 4](#_Toc477342808)

[1.2.2 Informative References 4](#_Toc477342809)

[1.3 Extension Overview (Synopsis) 5](#_Toc477342810)

[1.3.1 Organization of This Documentation 6](#_Toc477342811)

[1.4 Relationship to Standards and Other Extensions 6](#_Toc477342812)

[1.5 Applicability Statement 6](#_Toc477342813)

[2 Extensions 7](#_Toc477342814)

[2.1 Extensions to the Element Interface 7](#_Toc477342815)

[2.1.1 Attributes 8](#_Toc477342816)

[2.1.1.1 canHaveChildren 8](#_Toc477342817)

[2.1.1.2 canHaveHTML 8](#_Toc477342818)

[2.1.1.3 parentElement 8](#_Toc477342819)

[2.1.1.4 scopeName 9](#_Toc477342820)

[2.1.1.5 sourceIndex 9](#_Toc477342821)

[2.1.1.6 tagUrn 9](#_Toc477342822)

[2.1.2 Methods 9](#_Toc477342823)

[2.1.2.1 applyElement 9](#_Toc477342824)

[2.1.2.2 clearAttributes 10](#_Toc477342825)

[2.1.2.3 contains 10](#_Toc477342826)

[2.1.2.4 insertAdjacentElement 10](#_Toc477342827)

[2.1.2.5 mergeAttributes 11](#_Toc477342828)

[2.1.2.6 removeNode 12](#_Toc477342829)

[2.1.2.7 replaceNode 12](#_Toc477342830)

[2.1.2.8 swapNode 12](#_Toc477342831)

[2.1.3 Collections 13](#_Toc477342832)

[2.1.3.1 all 13](#_Toc477342833)

[2.1.3.2 children 14](#_Toc477342834)

[2.2 Extensions to the Comment Interface 15](#_Toc477342835)

[2.2.1 Attributes 15](#_Toc477342836)

[2.2.1.1 text 15](#_Toc477342837)

[2.2.1.2 atomic 15](#_Toc477342838)

[2.3 Extensions to the Document Interface 16](#_Toc477342839)

[2.3.1 Methods 16](#_Toc477342840)

[2.3.1.1 clear 16](#_Toc477342841)

[2.3.1.2 removeNode 17](#_Toc477342842)

[2.3.1.3 replaceNode 17](#_Toc477342843)

[2.3.1.4 swapNode 17](#_Toc477342844)

[2.4 Extensions to the Text Interface 18](#_Toc477342845)

[2.4.1 Methods 18](#_Toc477342846)

[2.4.1.1 removeNode 18](#_Toc477342847)

[2.4.1.2 replaceNode 18](#_Toc477342848)

[2.4.1.3 swapNode 19](#_Toc477342849)

[3 Security Considerations 20](#_Toc477342850)

[4 Appendix A: Product Behavior 21](#_Toc477342851)

[5 Change Tracking 22](#_Toc477342852)

[6 Index 23](#_Toc477342853)

# Introduction

This document describes extensions provided by Microsoft web browsers for the *Document Object Model (DOM) Level 2 Core Specification Version 1.0*[[DOM Level 2 - Core]](http://go.microsoft.com/fwlink/?LinkId=182703), W3C Recommendation 13 November, 2000.

Section 2 of this specification is normative. All other sections and examples in this specification are informative.

## Glossary

**MAY, SHOULD, MUST, SHOULD NOT, MUST NOT:** These terms (in all caps) are used as defined in [[RFC2119]](http://go.microsoft.com/fwlink/?LinkId=90317). All statements of optional behavior use either MAY, SHOULD, or SHOULD NOT.

## 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](http://msdn.microsoft.com/en-us/library/dn781092.aspx).

### 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] W3C, "Document Object Model (DOM) Level 2 Core Specification Version 1.0", W3C Recommendation 13 November, 2000, [http://www.w3.org/TR/DOM-Level-2-Core/](http://go.microsoft.com/fwlink/?LinkId=182703)

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

### Informative References

[CSS-Level2-2009] Bos, B., Celik, T., Hickson, I., and Wium Lie, H., Eds., "Cascading Style Sheets Level 2 Revision 1 (CSS 2.1) Specification", W3C Candidate Recommendation 08 September 2009, [http://www.w3.org/TR/2009/CR-CSS2-20090908/](http://go.microsoft.com/fwlink/?LinkId=182880)

[DOM Level 2 - HTML] W3C, "Document Object Model (DOM) Level 2 HTML Specification Version 1.0", W3C Recommendation, January 2003, [http://www.w3.org/TR/2003/REC-DOM-Level-2-HTML-20030109/](http://go.microsoft.com/fwlink/?LinkId=182708)

[DOM Level 2 - Style] W3C, "Document Object Model (DOM) Level 2 Style Specification Version 1.0", W3C Recommendation, November 2000, [http://www.w3.org/TR/2000/REC-DOM-Level-2-Style-20001113/](http://go.microsoft.com/fwlink/?LinkId=182710)

[HTML] World Wide Web Consortium, "HTML 4.01 Specification", December 1999, [http://www.w3.org/TR/html4/](http://go.microsoft.com/fwlink/?LinkId=89880)

[MS-CSS21E] Microsoft Corporation, "[Internet Explorer Extensions to Cascading Style Sheets (CSS) 2.1 and DOM Level 2 Style Specifications](file:///E%3A%5CTarget%5CIE%5CPublished%5CBooks%5CMS-DOM2CE%5C%5BMS-CSS21E%5D.pdf#Section_410044a186ab4fadad89f8a72e646ad9)".

[MS-DOM2CEX] Microsoft Corporation, "[Microsoft XML Extensions to the Document Object Model (DOM) Level 2 Core Specification](file:///E%3A%5CTarget%5CIE%5CPublished%5CBooks%5CMS-DOM2CE%5C%5BMS-DOM2CEX%5D.pdf#Section_a6643286fa9d4ad1913cdc602fe9a9d7)".

[MS-DOM2EE] Microsoft Corporation, "[Internet Explorer Extensions to the Document Object Model (DOM) Level 2 Events Specification](file:///E%3A%5CTarget%5CIE%5CPublished%5CBooks%5CMS-DOM2CE%5C%5BMS-DOM2EE%5D.pdf#Section_fc95d4415a0247c09588e4ef5a9b6749)".

[MS-DOM2E] Microsoft Corporation, "[Internet Explorer Document Object Model (DOM) Level 2 Events Standards Support Document](file:///E%3A%5CTarget%5CIE%5CPublished%5CBooks%5CMS-DOM2CE%5C%5BMS-DOM2E%5D.pdf#Section_d5a3f4ce8ba14755b31a45b52595305d)".

[MS-HTML401E] Microsoft Corporation, "[Internet Explorer Extensions to HTML 4.01 and DOM Level 2 HTML Specifications](file:///E%3A%5CTarget%5CIE%5CPublished%5CBooks%5CMS-DOM2CE%5C%5BMS-HTML401E%5D.pdf#Section_6e91051f13504dac8429fd5ef3e9193c)".

## Extension Overview (Synopsis)

The extensions described in this document were selected for their applicability to [[DOM Level 2 - Core]](http://go.microsoft.com/fwlink/?LinkId=182703).

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:

**Element**

* **Attributes**
	+ [canHaveChildren](#Section_9f6debce76cf4a4eadb926263aedcfcf)
	+ [canHaveHTML](#Section_7e60bc8094734db9a86cb322aa4128e6)
	+ [parentElement](#Section_595be4fc11f84040ba16d9aa3787852e)
	+ [scopeName](#Section_9206bf8a5eae4c93a9044645d92a2674)
	+ [sourceIndex](#Section_9f0fc588250e45e7bc44af26891b9e6e)
	+ [tagUrn](#Section_57a0354710374fcd8b005e8b41cb2034)
* **Methods**
	+ [applyElement](#Section_6bd86cfb54464329bf651ba741807c5c)
	+ [clearAttributes](#Section_136e0aa48ec34584b73275e9937ca18e)
	+ [contains](#Section_f7e56a72fdeb4434abf18e5f3741b29d)
	+ [insertAdjacentElement](#Section_62398cd7dc7143aeb7f5e549fe7f3856)
	+ [mergeAttributes](#Section_5bd53cf76e204e92a6779d8349b80594)
	+ [removeNode](#Section_d979f092cd6f4666b9ecb9e2ebc11db1)
	+ [replaceNode](#Section_fbb0e373853544bba242dd2f6e8593f9)
	+ [swapNode](#Section_f6a1cc593cad459397f7e39edba7c187)
* **Collections**
	+ [all](#Section_0c0ef006184f4a74882ae36c3b0f5e5c)
	+ [children](#Section_905dcce064ce4a2caf74680ea02e6481)

**Comment**

* **Attributes**
	+ [text](#Section_d92f7cfeaee64e949b16dbc9c12f2a23)
	+ [atomic](#Section_ddd659a2260640cd9553c0b7bfa7a463)

**Document**

* **Methods**
	+ [clear](#Section_75eec0c939374541bb661d4c2beb3e73)
	+ [removeNode](#Section_5dc3ebcfc4224e7680ba5d62c715ea28)
	+ [replaceNode](#Section_7dd45a0247fa46df873bea45776993eb)
	+ [swapNode](#Section_d923939b30c64d9a8735668d889b250c)

**Text**

* **Methods**
	+ [removeNode](#Section_40237f46e1e64db2af413ab214f24cdb)
	+ [replaceNode](#Section_416c0dacabf94cbfa4aa0a62e249e891)
	+ [swapNode](#Section_446899e80c0d4326950eaf96dbbb5f53)

### Organization of This Documentation

This document is organized as follows:

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

## Relationship to Standards and Other Extensions

The following documents provide information on additional extensions.

* [[MS-CSS21E]](file:///E%3A%5CTarget%5CIE%5CPublished%5CBooks%5CMS-DOM2CE%5C%5BMS-CSS21E%5D.pdf#Section_410044a186ab4fadad89f8a72e646ad9): Extensions to the [[CSS-Level2-2009]](http://go.microsoft.com/fwlink/?LinkId=182880) and [[DOM Level 2 - Style]](http://go.microsoft.com/fwlink/?LinkId=182710) specifications.
* [[MS-HTML401E]](file:///E%3A%5CTarget%5CIE%5CPublished%5CBooks%5CMS-DOM2CE%5C%5BMS-HTML401E%5D.pdf#Section_6e91051f13504dac8429fd5ef3e9193c): Extensions to the [[HTML]](http://go.microsoft.com/fwlink/?LinkId=89880) and the [[DOM Level 2 - HTML]](http://go.microsoft.com/fwlink/?LinkId=182708) specifications.
* [[MS-DOM2CEX]](file:///E%3A%5CTarget%5CIE%5CPublished%5CBooks%5CMS-DOM2CE%5C%5BMS-DOM2CEX%5D.pdf#Section_a6643286fa9d4ad1913cdc602fe9a9d7): Extensions to the [HTML] and the [DOM Level 2 - HTML] specifications for Microsoft XML Core Services.
* [[MS-DOM2EE]](file:///E%3A%5CTarget%5CIE%5CPublished%5CBooks%5CMS-DOM2CE%5C%5BMS-DOM2EE%5D.pdf#Section_fc95d4415a0247c09588e4ef5a9b6749): Extensions to the [[MS-DOM2E]](file:///E%3A%5CTarget%5CIE%5CPublished%5CBooks%5CMS-DOM2CE%5C%5BMS-DOM2E%5D.pdf#Section_d5a3f4ce8ba14755b31a45b52595305d) specifications.

## Applicability Statement

This document specifies a set of extensions to the [[DOM Level 2 - Core]](http://go.microsoft.com/fwlink/?LinkId=182703) specification. The extensions in this document provide access to some features that are unique to Windows Internet Explorer 7, Windows Internet Explorer 8, Windows Internet Explorer 9, Windows Internet Explorer 10, Internet Explorer 11, Internet Explorer 11 for Windows 10, and Microsoft Edge.

# Extensions

This section specifies additional attributes and methods to elements from [[DOM Level 2 - Core]](http://go.microsoft.com/fwlink/?LinkId=182703) that are available in Microsoft web browsers.

The extensions to [DOM Level 2 - Core] are as follows:

* Extensions to the [Element](#Section_8d60fee625a641c0ba6f862645a6bbd2) Interface
* Extensions to the [Comment](#Section_3aa97825d5434822b70806d696853479) Interface
* Extensions to the [Document](#Section_2fdf10781c5b424e94dc032d2c89f73e) Interface
* Extensions to the [Text](#Section_69a330c7f49f4c4f9f1388f745d15708) Interface

## Extensions to the Element Interface

Extensions have been added to the **Element** interface of the Document Object Model (DOM) Level 2 Core Specification [[DOM Level 2 - Core]](http://go.microsoft.com/fwlink/?LinkId=182703). The extensions of the **Element** interface are:

**Attributes**

The **Elements** interface is extended by the following attributes. For details, see [Attributes](#Section_8c4d2a1fb5f845158e1e01ceaa9435b4).

* [canHaveChildren](#Section_9f6debce76cf4a4eadb926263aedcfcf)
* [canHaveHTML](#Section_7e60bc8094734db9a86cb322aa4128e6)
* [parentElement](#Section_595be4fc11f84040ba16d9aa3787852e)
* [scopeName](#Section_9206bf8a5eae4c93a9044645d92a2674)
* [sourceIndex](#Section_9f0fc588250e45e7bc44af26891b9e6e)
* [tagUrn](#Section_57a0354710374fcd8b005e8b41cb2034)

**Methods**

The **Elements** interface is extended by the following methods. For details, see [Methods](#Section_6d2ad3f1da0a49de8c387d91c56819b5).

* [applyElement](#Section_6bd86cfb54464329bf651ba741807c5c)
* [clearAttributes](#Section_136e0aa48ec34584b73275e9937ca18e)
* [contains](#Section_f7e56a72fdeb4434abf18e5f3741b29d)
* [insertAdjacentElement](#Section_62398cd7dc7143aeb7f5e549fe7f3856)
* [mergeAttributes](#Section_5bd53cf76e204e92a6779d8349b80594)
* [removeNode](#Section_d979f092cd6f4666b9ecb9e2ebc11db1)
* [replaceNode](#Section_fbb0e373853544bba242dd2f6e8593f9)
* [swapNode](#Section_f6a1cc593cad459397f7e39edba7c187)

**Collections**

The **Elements** interface is extended by the following collections. For details, see [Collections](#Section_a0df5f555427409a83512cd5f7f95356).

* [all](#Section_0c0ef006184f4a74882ae36c3b0f5e5c)
* [children](#Section_905dcce064ce4a2caf74680ea02e6481)

The following IDL definition documents the **Element** interface:

interface Element : Node {

// Extension of DOM Level 2:

 readonly attribute boolean canHaveChildren;

 readonly attribute boolean canHaveHTML;

 readonly attribute boolean parentElement;

 readonly attribute long sourceIndex;

// Extension of DOM Level 2:

Element applyElement

void clearAttributes;

boolean contains;

Node insertAdjacentElement

Node mergeAttributes

boolean removeNode;

Node replaceNode (in Node newNode)

 raises(DOMException);

Node swapNode;

};

### Attributes

The **Elements** interface as specified in the [[DOM Level 2 - Core]](http://go.microsoft.com/fwlink/?LinkId=182703) is extended by the addition of the following attributes:

* [canHaveChildren](#Section_9f6debce76cf4a4eadb926263aedcfcf)
* [canHaveHTML](#Section_7e60bc8094734db9a86cb322aa4128e6)
* [parentElement](#Section_595be4fc11f84040ba16d9aa3787852e)
* [sourceIndex](#Section_9f0fc588250e45e7bc44af26891b9e6e)

#### canHaveChildren

**canHaveChildren** of type boolean, **read-only**

Gets a Boolean value indicating whether the object can contain child objects. Objects do not have to contain children for the **canHaveChildren** attribute to return true. The **canHaveChildren** attribute is useful in determining whether objects can be appended as children. The property has no default value.

#### canHaveHTML

**canHaveHTML** of type boolean

Retrieves the Boolean value indicating whether the object can contain rich HTML markup. The property is read-only for all objects except the **defaults** object, which is read-write. The property has no default value.

#### parentElement

**parentElement** of type Element, **read-only**

Retrieves the parent object in the object hierarchy. The highest object returns null as its parent. The property has no default value.

#### scopeName

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

**scopeName** of type String, **read-only**

Gets the namespace defined for the element. The property has a default value of HTML.

#### sourceIndex

**sourceIndex** of type long, **read-only**

Retrieves the ordinal position of the object, in source order, as the object appears in the document's **all** collection. The property has no default value.

#### tagUrn

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

**tagUrn** of type String

Sets or gets the Uniform Resource Name (URN) specified in the namespace declaration. The property has a default value of null.

### Methods

The **Element** interface is extended by the addition of the following methods:

* [applyElement](#Section_6bd86cfb54464329bf651ba741807c5c)
* [clearAttributes](#Section_136e0aa48ec34584b73275e9937ca18e)
* [contains](#Section_f7e56a72fdeb4434abf18e5f3741b29d)
* [insertAdjacentElement](#Section_62398cd7dc7143aeb7f5e549fe7f3856)
* [mergeAttributes](#Section_5bd53cf76e204e92a6779d8349b80594)
* [removeNode](#Section_d979f092cd6f4666b9ecb9e2ebc11db1)
* [replaceNode](#Section_fbb0e373853544bba242dd2f6e8593f9)
* [swapNode](#Section_f6a1cc593cad459397f7e39edba7c187)

#### applyElement

applyElement

The **applyElement** method makes the element either a child or parent of another element. The **applyElement** method is accessible at run time. If elements are removed at run time before the closing tag is parsed, areas of the document might not render.

**Parameters**

**oNewElement** of type Element

An object that becomes the child or parent of the current element.

**sWhere** of type String

Optional. If **outside**, the specified element becomes parent of the current element. The default parameter is **outside**. If **inside**, the specified element becomes a child of the current element, but contains all the child elements of the current element.

**Return Value**

**Element** The applied element, **oNewElement**.

**JScript Error**

**E\_INVALIDARG**: Raised if **oNewElement** is null, **oNewElement** is outside the root node, or if **oNewElement** is not owned by the current document.

#### clearAttributes

clearAttributes

The **clearAttributes** method removes attributes and values from the object. It clears persistent HTML attributes only; the **ID** attribute, styles, and script-only properties are not affected.

**Parameters**

None.

**Return Value**

None.

**No Jscript Error**

#### contains

contains

The **contains** method checks whether the specified element is contained within the object.

**Parameters**

oElement

Element object that specifies the element to check.

**Return Value**

Boolean. Returns one of the following possible values:

     True - The element is contained within the object.

     False - The element is not contained within the object.

**JScript Error**

The error **'null' is null or not an object** is raised if **elements** defined in the script are not present.

#### insertAdjacentElement

insertAdjacentElement

The **insertAdjacentElement** method inserts an element at the specified location. Text cannot be inserted while a document is loading; the onload event must be completed before attempting to call the **insertAdjacentElement** method.

**Parameters**

sWhere

A String that specifies where to insert the HTML element, using one of the following values:

     beforeBegin - Inserts **oElement** immediately before the object.

     afterBegin - Inserts **oElement** after the start of the object, but before all other content in the object.

     beforeEnd - Inserts **oElement** immediately before the end of the object, but after all other content in the object.

     afterEnd - Inserts **oElement** immediately after the end of the object.

oElement

Object that specifies the element to be inserted adjacent to the object that invoked the **insertAdjacentElement** method.

**Return Value**

Returns an element object.

**No JScript Error**

#### mergeAttributes

mergeAttributes

The **mergeAttributes** method copies all read/write HTML attributes, events, and styles from one element to another specified element.

**Parameters**

oSource

Pointer to an **Object** that specifies the attributes copied to the object that invokes the attributes of the **mergeAttributes** method.

bPreserve

Optional. Pointer to a Boolean value that specifies one of the following values:

     True - Default. The **id** and **name** attributes of the element to which attributes are being merged is preserved.

     False - The **id** and **name** attributes of the element to which attributes are being merged are not preserved.

**Return Value**

No return value.

#### removeNode

removeNode

The **removeNode** method removes the specified object from the document hierarchy. The **removeNode** method is accessible at run time. If elements are removed at run time, before the closing tag is parsed, areas of the document might not render.

**Parameters**

bRemoveChildren

Optional. A Boolean parameter that specifies one of the following values:

     False - Default. The **childNodes** collection of the object is not removed.

     True - The **childNodes** collection of the object is removed.

**Return Value**

Returns a reference to the removed object.

#### replaceNode

replaceNode

The **replaceNode** method replaces the object with another element. When a node is replaced, all values that are associated with the replaced object are removed. The **replaceNode** method is accessible at run time. If elements are removed at run time before the closing tag is parsed, areas of the document might not render.

**Parameters**

oNewNode

An object that specifies the new element to replace the object.

**Return Value**

Returns a reference to the removed object.

#### swapNode

swapNode

The **swapNode** method exchanges the location of two objects in the document hierarchy. This method is accessible at run time. If elements are removed at run time, before the closing tag is parsed, areas of the document might not render.

**Parameters**

oNode

Element that specifies the existing element.

**Return Values**

Returns a reference to the object that invoked the method.

### Collections

The following collections are extensions to the **Elements** interface as specified in the [[DOM Level 2 - Core]](http://go.microsoft.com/fwlink/?LinkId=182703):

* [all](#Section_0c0ef006184f4a74882ae36c3b0f5e5c)
* [children](#Section_905dcce064ce4a2caf74680ea02e6481)

#### all

all

Returns a reference to the collection of elements contained by the object.

**Note.**  The **all** extension is not supported (does not exist) in IE11 Mode.

The **all** collection includes one element object for each valid HTML tag. If a valid tag has a matching end tag, both tags are represented by the same element object.

The collection returned by the document's **all** collection always includes a reference to the **html**, **head**, and **title** objects regardless of whether the tags are present in the document. If the BODY tag is not present, but other HTML tags are, a **body** object is added to the **all** collection.

If the document contains invalid or unknown tags, the collection includes one element object for each. Unlike valid end tags, unknown end tags are represented by their own element objects. The order of the element objects is the HTML source order. Although the collection indicates the order of tags, it does not indicate hierarchy.

The **name** property only applies to some elements such as **form** elements. If the vIndex is set to a string matching the value of a **name** property in an element that the **name** property does not apply, then that element is not added to the collection.

**Syntax**

[ collAll = ] object.**all**

[ oObject = ] object.**all**(vIndex [, iSubIndex])

**Possible Values**

**collAll**   Array of elements contained by the object.

**oObject**   Reference to an individual item in the array of elements contained by the object.

**vIndex**   Required. Integer or string that specifies the element or collection to retrieve. If this parameter is an integer, the method returns the element in the collection at the specified position, where the first element has value 0, the second has 1, and so on. If this parameter is a string and there is more than one element with the **name** or **id** property equal to the string, the method returns a collection of matching elements.

**iSubIndex**   Optional. Position of an element to retrieve. This parameter is used when vIndex is a string. The method uses the string to construct a collection of all elements that have a **name** or **id** property equal to the string, and then retrieves from this collection the element at the position specified by iSubIndex.

**Members Table**

The following table lists the members exposed by the **all** object.

Attributes

| Attribute | Description |
| --- | --- |
| length | Sets or retrieves the number of objects in a collection. |

Methods

| Method | Description |
| --- | --- |
| item | Retrieves an object from various collections, including the **all** collection. |
| namedItem | Retrieves an object or a collection from a specified collection. |
| tags | Retrieves a collection of objects that have the specified HTML tag name. |
| urns | Retrieves a collection of all objects to which a specified behavior is attached. |

#### children

children

Retrieves a collection of DHTML Objects that are direct descendants of the object.

Similar to the objects contained in the **all** collection, the objects contained in the **children** collection are undefined if the child elements are overlapping tags.

The **children** collection can contain HTML elements.

**Syntax**

[ collAll = ] object.**children**

[ oObject = ] object.**children**(vIndex [, iSubIndex])

**Possible Values**

**collAll**   Array of elements contained by the object.

**oObject**   Reference to an individual item in the array of elements contained by the object.

**vIndex**   Required. Integer or string that specifies the element or collection to retrieve. If this parameter is an integer, the method returns the element in the collection at the specified position, where the first element has value 0, the second has 1, and so on. If this parameter is a string and there is more than one element with the **name** or **id** property equal to the string, the method returns a collection of matching elements.

**iSubIndex**   Optional. Position of an element to retrieve. This parameter is used when **vIndex** is a string. The method uses the string to construct a collection of all elements that have a **name** or **id** property equal to the string, and then retrieves from this collection the element at the position specified by **iSubIndex**.

**Members Table**

The following table lists the members exposed by the **children** object.

Attributes

| Attribute | Description |
| --- | --- |
| constructor | Returns a reference to the constructor of an object. |
| length | Sets or retrieves the number of objects in a collection. |

Methods

| Method | Description |
| --- | --- |
| item | Retrieves an object from various collections, including the **all** collection. |
| tags | Retrieves a collection of objects that have the specified HTML tag name. |
| urns | Retrieves a collection of all objects to which a specified behavior is attached. |

## Extensions to the Comment Interface

The following attributes of the **Comment** Interface are prototype extensions to the Comment interface of the Document Object Model (DOM) Level 2 Core Specification [[DOM Level 2 - Core]](http://go.microsoft.com/fwlink/?LinkId=182703).

**Attributes**

The **Comment** interface is extended by the [text](#Section_d92f7cfeaee64e949b16dbc9c12f2a23) and [atomic](#Section_ddd659a2260640cd9553c0b7bfa7a463) attributes. For details, see [Attributes](#Section_0531ff35167d4913803f0ccabc7255f4).

The following IDL Definition documents the **Comment** interface:

1. *interface Comment : CharacterData {*
2. *// Extension of DOM Level 2:*
3. *attribute text;*
4. *readonly attribute atomic;*
5. *};*

### Attributes

The **Comment** interface as specified in the [[DOM Level 2 - Core]](http://go.microsoft.com/fwlink/?LinkId=182703) is extended by the addition of the [text](#Section_d92f7cfeaee64e949b16dbc9c12f2a23) attribute.

#### text

**text** of typeDOMString

The **text** attribute retrieves or sets the text of the object as a string.

#### atomic

atomic

The **atomic** attribute is functional but has been deprecated. The use of **atomic** indicates whether **comment** is a standalone tag or is used in a tag pair.

**Parameters**

None

**Return Values**

Returns one of the following possible values:

     1 - if the comment is self-closing <!-- --> style.

     0 - if the comment uses the deprecated <comment> tag.

## Extensions to the Document Interface

The following methods are extensions to the **Document** interface of the Document Object Model (DOM) Level 2 Core Specification [[DOM Level 2 - Core]](http://go.microsoft.com/fwlink/?LinkId=182703):

* [clear](#Section_75eec0c939374541bb661d4c2beb3e73)
* [removeNode](#Section_5dc3ebcfc4224e7680ba5d62c715ea28)
* [replaceNode](#Section_7dd45a0247fa46df873bea45776993eb)
* [swapNode](#Section_d923939b30c64d9a8735668d889b250c)

The following IDL definition documents the **Document** interface:

1. interface Element : Node {
2. // Extension of DOM Level 2:
3. boolean clear;
4. boolean removeNode;
5. Node replaceNode (in Node newNode)
6. raises(DOMException);
7. Node swapNode;
8. };

### Methods

The following properties are extensions to the **Document** interface as specified in the [[DOM Level 2 - Core]](http://go.microsoft.com/fwlink/?LinkId=182703):

* [clear](#Section_75eec0c939374541bb661d4c2beb3e73)
* [removeNode](#Section_5dc3ebcfc4224e7680ba5d62c715ea28)
* [replaceNode](#Section_7dd45a0247fa46df873bea45776993eb)
* [swapNode](#Section_d923939b30c64d9a8735668d889b250c)

#### clear

clear

The **clear** method removes all key/value pairs from the DOM Storage area. Session storage is cleared immediately. Local storage key/value pairs are removed from memory, and disk storage quota is updated.

**Return Value**

None

**JScript Error**

None

#### removeNode

removeNode

The **removeNode** method removes the object from the document hierarchy.

**Parameters**

bRemoveChildren

Optional. A Boolean parameter that specifies one of the following values:

     False - Default. **childNodes** collection of the object is not removed.

     True - **childNodes** collection of the object is removed.

**Return Value**

Returns a reference to the removed object.

#### replaceNode

replaceNode

The **replaceNode** method replaces the object with another element. When a node is replaced, all values that are associated with the replaced object are removed. The **replaceNode** method is accessible at run time. If elements are removed at run time before the closing tag is parsed, areas of the document might not render.

**Parameters**

oNewNode

An element that specifies the new element to replace the object.

**Return Value**

Returns a reference to the removed object.

#### swapNode

swapNode

The **swapNode** method exchanges the location of two objects in the document hierarchy. This method is accessible at run time. If elements are removed at run time, before the closing tag is parsed, areas of the document might not render.

**Parameters**

oNode

Element that specifies the existing element.

**Return Values**

Returns a reference to the object that invoked the method.

## Extensions to the Text Interface

The following methods are extensions to the **Text** interface of the Document Object Model (DOM) Level 2 Core Specification [[DOM Level 2 - Core]](http://go.microsoft.com/fwlink/?LinkId=182703):

* [removeNode](#Section_40237f46e1e64db2af413ab214f24cdb)
* [replaceNode](#Section_416c0dacabf94cbfa4aa0a62e249e891)
* [swapNode](#Section_446899e80c0d4326950eaf96dbbb5f53)

The following IDL definition documents the **Document** interface:

1. interface Element : Node {
2. // Extension of DOM Level 2:
3. boolean removeNode;
4. Node replaceNode (in Node newNode)
5. raises(DOMException);
6. Node swapNode;
7. };

### Methods

The following properties are extensions to the **Text** interface as specified in the [[DOM Level 2 - Core]](http://go.microsoft.com/fwlink/?LinkId=182703):

* [removeNode](#Section_40237f46e1e64db2af413ab214f24cdb)
* [replaceNode](#Section_416c0dacabf94cbfa4aa0a62e249e891)
* [swapNode](#Section_446899e80c0d4326950eaf96dbbb5f53)

#### removeNode

removeNode

The **removeNode** removes the object from the document hierarchy.

**Parameters**

bRemoveChildren

Optional. A Boolean parameter that specifies one of the following values:

     False - Default. **childNodes** collection of the object is not removed.

     True - **childNodes** collection of the object is removed.

**Return Value**

Returns a reference to the removed object.

#### replaceNode

replaceNode

The **replaceNode** method replaces the object with another element.

**Parameters**

oNewNode

An element that specifies the new element to replace the object.

**Return Value**

Returns a reference to the removed object.

#### swapNode

swapNode

The **swapNode** method exchanges the location of two objects in the document hierarchy. This method is accessible at run time. If elements are removed at run time, before the closing tag is parsed, areas of the document might not render.

**Parameters**

oNode

Element that specifies the existing element.

**Return Values**

Returns a reference to the object that invoked the method.

# Security Considerations

No security considerations.

# Appendix A: Product Behavior

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

* Windows Internet Explorer 7
* Windows Internet Explorer 8
* Windows Internet Explorer 9
* Windows Internet Explorer 10
* Internet Explorer 11
* Internet Explorer 11 for Windows 10
* Microsoft Edge

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

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

# Change Tracking

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

# Index

A

[Applicability](#section_aab7fe074fb448ec85536139f7aae931) 6

Attributes

 [canHaveChildren](#section_9f6debce76cf4a4eadb926263aedcfcf) 8

 [canHaveHTML](#section_7e60bc8094734db9a86cb322aa4128e6) 8

 [parentElement](#section_595be4fc11f84040ba16d9aa3787852e) 8

 [sourceIndex](#section_9f0fc588250e45e7bc44af26891b9e6e) 9

 [text](#section_d92f7cfeaee64e949b16dbc9c12f2a23) 15

C

[Change tracking](#section_bd90937744854faab5cd8018f6eaefb1) 22

Collections

 [all](#section_0c0ef006184f4a74882ae36c3b0f5e5c) 13

 [children](#section_905dcce064ce4a2caf74680ea02e6481) 14

G

[Glossary](#section_d1d07ae5acd64cc5a6f318f16765b2ac) 4

I

[Implementer - security considerations](#section_ff70f943336348d79c4db900bfc9a400) 20

[Informative references](#section_47695a9b07e4475e805382917e38ab19) 4

Interfaces

 [Comment](#section_3aa97825d5434822b70806d696853479) 15

 [Document](#section_2fdf10781c5b424e94dc032d2c89f73e) 16

 [Element](#section_8d60fee625a641c0ba6f862645a6bbd2) 7

 [Text](#section_69a330c7f49f4c4f9f1388f745d15708) 18

[Introduction](#section_bb7bac5990064c978a2ff5c4df90fd1c) 4

M

Methods

 [applyElement](#section_6bd86cfb54464329bf651ba741807c5c) 9

 [atomic](#section_ddd659a2260640cd9553c0b7bfa7a463) 15

 [clear](#section_75eec0c939374541bb661d4c2beb3e73) 16

 [clearAttributes](#section_136e0aa48ec34584b73275e9937ca18e) 10

 [contains](#section_f7e56a72fdeb4434abf18e5f3741b29d) 10

 [insertAdjacentElement](#section_62398cd7dc7143aeb7f5e549fe7f3856) 10

 [mergeAttributes](#section_5bd53cf76e204e92a6779d8349b80594) 11

 removeNode ([section 2.1.2.6](#section_d979f092cd6f4666b9ecb9e2ebc11db1) 12, [section 2.3.1.2](#section_5dc3ebcfc4224e7680ba5d62c715ea28) 17, [section 2.4.1.1](#section_40237f46e1e64db2af413ab214f24cdb) 18)

 replaceNode ([section 2.1.2.7](#section_fbb0e373853544bba242dd2f6e8593f9) 12, [section 2.3.1.3](#section_7dd45a0247fa46df873bea45776993eb) 17, [section 2.4.1.2](#section_416c0dacabf94cbfa4aa0a62e249e891) 18)

 swapNode ([section 2.1.2.8](#section_f6a1cc593cad459397f7e39edba7c187) 12, [section 2.3.1.4](#section_d923939b30c64d9a8735668d889b250c) 17, [section 2.4.1.3](#section_446899e80c0d4326950eaf96dbbb5f53) 19)

N

[Normative references](#section_550a9ef82285441891ae0274b807dd04) 4

O

[Overview (synopsis)](#section_cbc3c457348b4965a153a4447e0e563b) 5

P

[Product behavior](#section_3585acc23b314c8a91a696048009d946) 21

R

[References](#section_642759ba25fd4e8ebb8e6a41a0fb5d34) 4

 [informative](#section_47695a9b07e4475e805382917e38ab19) 4

 [normative](#section_550a9ef82285441891ae0274b807dd04) 4

S

[Security - implementer considerations](#section_ff70f943336348d79c4db900bfc9a400) 20

T

[Tracking changes](#section_bd90937744854faab5cd8018f6eaefb1) 22