[MS-DOM2H]:

Internet Explorer Document Object Model (DOM) Level 2 HTML Standards Support Document

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

Support. For questions and support, please contact dochelp@microsoft.com.

Revision Summary

Date	Revision History	Revision Class	Comments
3/17/2010	0.1	New	Released new document.
3/26/2010	1.0	None	Introduced no new technical or language changes.
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	Minor	Clarified the meaning of the technical content.
2/22/2012	3.0	Major	Significantly changed the technical content.
7/25/2012	3.1	Minor	Clarified the meaning of the technical content.
6/26/2013	4.0	Major	Significantly changed the technical content.
3/31/2014	4.0	None	No changes to the meaning, language, or formatting of the technical content.
1/22/2015	5.0	Major	Updated for new product version.
7/7/2015	5.1	Minor	Clarified the meaning of the technical content.
11/2/2015	5.2	Minor	Clarified the meaning of the technical content.
3/22/2016	5.2	None	No changes to the meaning, language, or formatting of the technical content.
7/19/2016	5.3	Minor	Clarified the meaning of the technical content.
11/2/2016	5.3	None	No changes to the meaning, language, or formatting of the technical content.
3/14/2017	5.3	None	No changes to the meaning, language, or formatting of the technical content.
10/3/2017	5.3	None	No changes to the meaning, language, or formatting of the technical content.
2/22/2018	5.3	None	No changes to the meaning, language, or formatting of the technical content.
3/23/2018	5.3	None	No changes to the meaning, language, or formatting of the technical content.

Table of Contents

1 Introduction

This document describes the level of support provided by Microsoft web browsers for the *Document Object Model (DOM) Level 2 HTML Specification Version 1.0* [DOM Level 2 - HTML], published 9 January 2003.

The [DOM Level 2 - HTML] 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 as defined in [RFC2119]. All statements of optional behavior use either MAY, SHOULD, or SHOULD NOT.

1.2 References

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

1.2.1 Normative References

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

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

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997, http://www.rfc-editor.org/rfc/rfc2119.txt

1.2.2 Informative References

None.

1.3 Microsoft Implementations

The following web browser versions implement some portion of the [DOM Level 2 - HTML] specification:

- 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

Each browser version may implement multiple document rendering modes. The modes vary from one to another in support of the standard. The following table lists the document modes supported by each browser version.

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
Internet Explorer 11	Quirks Mode IE7 Mode IE8 Mode IE9 Mode IE10 Mode IE11 Mode
Internet Explorer 11 for Windows 10	Quirks Mode IE7 Mode IE8 Mode IE9 Mode IE10 Mode IE11 Mode
Microsoft Edge	EdgeHTML Mode

For each variation presented in this document there is a list of the document modes and browser versions that exhibit the behavior described by the variation. All combinations of modes and versions that are not listed conform to the specification. For example, the following list for a variation indicates that the variation exists in three document modes in all browser versions that support these modes:

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

Note: "Standards Mode" in Internet Explorer 7 and "IE7 Mode" in Internet Explorer 8 refer to the same 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 2 - HTML] 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].)

The following table lists the sections of [DOM Level 2 - HTML] and whether they are considered normative or informative.

Sections	Normative/Informative
1	Normative
Appendix A-E	Informative

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].) 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.

2 Standards Support Statements

This section contains all variations, clarifications, and extensions for the Microsoft implementation of [DOM Level 2 - HTML].

- Section 2.1 describes normative variations from the MUST requirements of the specification.
- Section 2.2 describes clarifications of the MAY and SHOULD requirements.
- Section 2.3 describes extensions to the requirements.
- Section 2.4 considers error handling aspects of the implementation.
- Section 2.5 considers security aspects of the implementation.

2.1 Normative Variations

The following subsections describe normative variations from the MUST requirements of [DOM Level 2 - HTML].

2.1.1 [DOM Level 2 - HTML] Section 1.1, Introduction

V0024:

The specification states:

```
The interfaces found within this section are not mandatory. A DOM application may
use the hasFeature(feature, version) method of the DOMImplementation interface with
parameter values "HTML" and "2.0" (respectively) to determine whether or not this
module is supported by the implementation. In addition to the feature string
"HTML", the feature string "XHTML" (version string "2.0") can be used to check if
the implementation supports XHTML (this is equivalent to checking the features
"XML" and "HTML"). In order to fully support this module, an implementation must
also support the "Core" feature defined. Please refer to additional information
about conformance in the DOM Level 2 Core specification.
```

All Document Modes (Internet Explorer 7 and Internet Explorer 8)

The **hasFeature** method returns FALSE when the feature parameter is set to `"HTML"` and the version parameter is set to `"2.0"`.

2.1.2 [DOM Level 2 - HTML] Section 1.4, Misc Object Definitions

V0025:

The specification states:

```
interface HTMLOptionsCollection
namedItem
This method retrieves a Node using a name. It first searches for a Node with a
matching id attribute. If it doesn't find one, it then searches for a Node with a
matching name attribute, but only on those elements that are allowed a name
attribute. This method is case insensitive in HTML documents and case sensitive in
XHTML documents.

Parameters
name of type DOMString
The name of the Node to be fetched.
```

```
Return Value
Node
The Node with a name or id attribute whose value corresponds to the specified
string. Upon failure (e.g., no node with this name exists), returns null.

No Exceptions
```

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

XHTML documents are not supported by the **namedItem** method of the **HTMLOptionsCollection** interface.

IE9 Mode (All Versions)

The **namedItem** method of the **HTMLOptionsCollection** interface is not case-sensitive.

2.1.3 [DOM Level 2 - HTML] Section 1.5, Objects related to HTML documents

V0026:

The specification states:

```
Interface HTMLDocument
An HTMLDocument is the root of the HTML hierarchy and holds the entire content.
Besides providing access to the hierarchy, it also provides some convenience
methods for accessing certain sets of information from the document.

The following properties have been deprecated in favor of the corresponding ones
for the BODY element:


alinkColor
background
bgColor
fgColor
linkColor
vlinkColor
```

All Document Modes (All Versions)
The **background** attribute of the **HTMLDocument** interface is not supported.

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

Setting the **body** element of the **HTMLDocument** interface to a frameset is not supported.

V0027:

The specification states:

```
URL of type DOMString, readonly
The absolute URI of the document.
```

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

The uppercase **URL** attribute name of the **HTMLDocument** interface is implemented as a lowercase name (**url**).

V0029:

The specification states:

```
getElementsByName
With documents, this method returns the (possibly empty) collection of elements
whose name value is given by elementName. In XHTML 1.0 documents, this methods only
return the (possibly empty) collection of form controls with matching name. This
method is case sensitive.

Parameters
ElementName of type DOMString
The name attribute value for an element.

Return Value
NodeList
The matching elements.

No Exceptions
```

Quirks Mode and IE7 Mode (All Versions)

The **getElementsByName** method matches on the **name**, **id**, and **uniquename** attributes, instead of on only the **name** attribute. Only one matched element is included in the node list.

IE8 Mode and IE9 Mode (All Versions)

The **getElementsByName** method matches on the **name**, **id**, and **uniquename** attributes, instead of on only the **name** attribute. All matched elements are included in the node list.

Quirks Mode, IE7 Mode, IE8 Mode, IE9 Mode, and IE10 Mode (All Versions)

The **getElementsByName** method does not match expando attributes on an element. For example, **name** is not an attribute on **div** elements, therefore it does not return a match on `document.getElementsByName('testDiv')` when `<DIV name="testDiv"></DIV>`.

2.1.4 [DOM Level 2 - HTML] Section 1.6.5, Object definitions

V0002:

The specification states:

```
form of type HTMLFormElement, readonly
Returns the FORM element containing this control. Returns null if this control is
not within the context of a form.
```

IE7 Mode, IE8 Mode, IE9 Mode, IE10 Mode, and IE11 Mode (All Versions)

The **form** property of the **ISINDEX** element does not return NULL when the control is outside the context of a form.

V0003:

The specification states:

```
text of type DOMString
Document text color. See the text attribute definition in HTML 4.01. This attribute
is deprecated in HTML 4.01.
```

Quirks Mode, IE7 Mode, IE8 Mode, IE9 Mode, and IE10 Mode (All Versions)

The value of the **text** attribute is converted into a hexadecimal RGB value.

V0004:

The specification states:

```
add
Add a new element to the collection of OPTION elements for this SELECT. This method
is the equivalent of the appendChild method of the Node interface if the before
parameter is null. It is equivalent to the insertBefore method on the parent of
before in all other cases. This method may have no effect if the new element is not
an OPTION or an OPTGROUP.

Parameters
element of type HTMLElement
The element to add.

before of type HTMLElement
The element to insert before, or null for the tail of the list.

Exceptions
DOMException
NOT_FOUND_ERR: Raised if before is not a descendant of the SELECT element.

No Return Value
```

Quirks Mode and IE7 Mode (All Versions)

The **before** parameter of the **add** method of the **HTMLSelectElement** interface is not supported.

All Document Modes (All Versions)

The DOMException **NOT_FOUND_ERR** is not raised by the **add** method of the **HTMLSelectElement** interface.

V0005:

The specification states:

```
interface HTMLOptionElement : HTMLElement

selected of type boolean
Represents the current state of the corresponding form control, in an interactive
user agent. Changing this attribute changes the state of the form control, but does
not change the value of the HTML selected attribute of the element.
```

All Document Modes (All Versions)

Changing the value of the **selected** attribute of **HTMLOptionElement** results in an equivalent change to the value of the HTML **selected** attribute of the element.

V0006:

The specification states:

```
align of type DOMString
Aligns this object (vertically or horizontally) with respect to its surrounding
text. See the align attribute definition in HTML 4.01. This attribute is deprecated
in HTML 4.01.
```

All Document Modes (All Versions)

The value of the **align** attribute of the **HTMLInputElement** interface returns an empty string ("").

V0008:

The specification states:

```
interface HTMLInputElement : HTMLElement
value of type DOMString
When the type attribute of the element has the value "text", "file" or "password",
this represents the current contents of the corresponding form control, in an
interactive user agent. Changing this attribute changes the contents of the form
control, but does not change the value of the HTML value attribute of the element.
When the type attribute of the element has the value "button", "hidden", "submit",
"reset", "image", "checkbox" or "radio", this represents the HTML value attribute
of the element. See the value attribute definition in HTML 4.01.
```

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

Changing the **value** attribute of the **HTMLInputElement** interface to a `text`, `file`, or `password` input type results in an equivalent change to the HTML **value** attribute of the element. The **value** attribute is an empty string for an input type of `file`.

V0010:

The specification states:

```
value of type long
Reset sequence number when used in OL. See the value attribute definition in HTML
4.01. This attribute is deprecated in HTML 4.01.
```

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

The **value** attribute of the **HTMLLIElement** interface cannot be used to reset the sequence number of an **OL** element (ordered list).

V0011:

The specification states:

```
Interface HTMLPreElement
Preformatted text. See the PRE element definition in HTML 4.01.
IDL Definition
interface HTMLPreElement : HTMLElement {
attribute long            width;
};
```

IE7 Mode, IE8 Mode, IE9 Mode, IE10 Mode, and IE11 Mode (All Versions)

The **width** attribute of the **HTMLPreElement** interface is of type `string`.

V0012:

The specification states:

```
color of type DOMString
Font color. See the color attribute definition in HTML 4.01. This attribute is
deprecated in HTML 4.01.
```

IE11 Mode and EdgeHTML Mode (All Versions)

The **color** attribute of the **HTMLBaseFontElement** interface returns the RGB hexadecimal value for a color string (for example, the color string `blue` is converted to `#0000FF`).

V0013:

The specification states:

```
isMap of type boolean
Use server-side image map. See the ismap attribute definition in HTML 4.01.
```

All Document Modes (All Versions)

The **isMap** attribute of the **HTMLImageElement** interface always returns a value of `true`, even when the value is set to `false`.

V0014:

The specification states:

```
interface HTMLObjectElement : HTMLElement
codeBase of type DOMString
Base URI for classid, data, and archive attributes. See the codebase attribute
definition in HTML 4.01.
```

All Document Modes (All Versions)

If the **codeBase** attribute of the **OBJECT** element is not specified in HTML, the **codeBase** attribute of the **HTMLObjectElement** interface returns an empty string. Otherwise, if the **codeBase** attribute is specified in HTML, **codeBase** returns the base URI of the current document appended with the value specified for the **codeBase** attribute.

V0015:

The specification states:

```
interface HTMLObjectElement : HTMLElement
codeType of type DOMString
Content type for data downloaded via classid attribute. See the codetype attribute
definition in HTML 4.01.
```

All Document Modes (All Versions)

If the **codetype** attribute of the **object** element is not specified in HTML, or the attribute is declared without an assigned value, the **codeType** attribute of **HTMLObjectElement** returns an empty string as a default value instead of the value of the **type** attribute for the **object** element.

V0016:

The specification states:

```
form of type HTMLFormElement, readonly
Returns the FORM element containing this control. Returns null if this control is
not within the context of a form.
```

Quirks Mode, IE7 Mode, and IE8 Mode (All Versions)
Returns the form element even if the control is not within the context of a form.

V0017:

The specification states:

```
object of type DOMString, modified in DOM Level 2
The value of the "object" attribute. See the object attribute definition in HTML
4.01. This attribute is deprecated in HTML 4.01.
```

IE17 Mode, IE8 Mode, IE9 Mode, IE10 Mode, and IE11 Mode (All Versions)

The **object** attribute of the **HTMLAppletElement** interface is not supported.

V0018:

The specification states:

```
ch of type DOMString
Alignment character for cells in a column. See the char attribute definition in
HTML 4.01.
```

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

The **ch** attribute of the **TR** element is not supported. The value of the **ch** attribute can be set, but it does not change the alignment of text within a cell.

V0019:

The specification states:

```
chOff of type DOMString
Offset of alignment character. See the charoff attribute definition in HTML 4.01.
```

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

The **chOff** attribute of the **TR** element is not supported. The value of the **chOff** attribute can be set, but it does not change the offset of the alignment.

V0020:

The specification states:

```
deleteCell modified in DOM Level 2
Delete a cell from the current row.

Parameters
index of type long
The index of the cell to delete, starting from 0. If the index is -1 the last cell
in the row is deleted.

Exceptions
DOMException INDEX_SIZE_ERR: Raised if the specified index is greater than or equal
to the number of cells or if the index is a negative number other than -1.

No Return Value
```

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

If the specified index is greater than or equal to the number of cells or if the index is a negative number other than –1, the **deleteCell** method of the **HTMLTableRowElement** interface throws a JScript **Error** object with a description of "Invalid argument" and an error number of 0x80070057 (-2147024809).

V0021:

The specification states:

```
insertCell modified in DOM Level 2
Insert an empty TD cell into this row. If index is -1 or equal to the number of
cells, the new cell is appended.
Parameters
index of type long
The place to insert the cell, starting from 0.

Return Value
HTMLElement
The newly created cell.

Exceptions
DOMException INDEX_SIZE_ERR: Raised if the specified index is greater than the
number of cells or if the index is a negative number other than -1.
```

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

If the specified index is greater than or equal to the number of cells or if the index is a negative number other than –1, the **insertCell** method of the **HTMLTableRowElement** interface throws a JScript **Error** object with a description of "Invalid argument" and an error number of 0x80070057 (-2147024809).

V0022:

The specification states:

```
rowSpan of type long
Number of rows spanned by cell. See the rowspan attribute definition in HTML 4.01.
```

Quirks Mode and IE7 Mode (All Versions)

Setting the value of the **rowSpan** attribute of the **HTMLTableCellElement** interface to a number greater than the actual number of rows in the table results in the value being clipped to the actual number of rows.

V0023:

The specification states:

```
contentDocument of type Document, readonly, introduced in DOM Level 2
The document this frame contains, if there is any and it is available, or null
otherwise.
```

Quirks Mode and IE7 Mode (All Versions)

The **contentDocument** attribute is not supported on the **HTMLIFrameElement** interface.

IE8 Mode (Internet Explorer 8)

When the **HTMLIFrameElement** object is empty, the **contentDocument** attribute returns the document this frame contains. When the **src** attribute of the **HTMLIFrameElement** is set to a different domain, access to the **contentDocument** attribute is denied.

V0030:

The specification states:

```
version of type DOMString
Version information about the document's DTD. See the version attribute definition
in HTML 4.01. This attribute is deprecated in HTML 4.01.
```

All Document Modes (All Versions)

The **version** attribute of the **HTMLHtmlElement** interface returns an empty string ("").

V0031:

The specification states:

```
media of type DOMString
Designed for use with one or more target media. See the media attribute definition
in HTML 4.01.
```

All Document Modes (All Versions)

The default value for the **media** attribute of the **HTMLLinkElement** is the empty string instead of the expected value of `screen`.

V0032:

The specification states:

```
acceptCharset of type DOMString
List of character sets supported by the server. See the accept-charset attribute
definition in HTML 4.01.
```

All Document Modes (All Versions)

If the value of the **acceptCharset** attribute of the **HTMLFormElement** interface is not set, a default value of `UNKNOWN` is returned.

V0035:

The specification states:

```
value of type DOMString
The current form control value. See the value attribute definition in HTML 4.01.
```

Quirks and IE7 Mode (All Versions)

The **value** attribute of the **HTMLButtonElement** interface returns the contents of the element.

V0036:

The specification states:

```
focus
```

```
Gives keyboard focus to this element.
No Parameters
No Return Value
No Exceptions
```

All Document Modes (Internet Explorer 7)

The **focus** method of the **HTMLAnchorElement** interface is not supported.

V0037:

The specification states:

```
Interface HTMLObjectElement
Generic embedded object.
Note: In principle, all properties on the object element are read-write but in some
environments some properties may be read-only once the underlying object is
instantiated. See the OBJECT element definition in HTML 4.01.

IDL Definition
interface HTMLObjectElement : HTMLElement {
readonly attribute HTMLFormElement form;
attribute DOMString       code;
attribute DOMString       align;
attribute DOMString       archive;
attribute DOMString       border;
attribute DOMString       codeBase;
attribute DOMString       codeType;
attribute DOMString       data;
attribute boolean         declare;
attribute DOMString       height;
attribute long            hspace;
attribute DOMString       name;
attribute DOMString       standby;
attribute long            tabIndex;
attribute DOMString       type;
attribute DOMString       useMap;
attribute long            vspace;
attribute DOMString       width;
// Introduced in DOM Level 2:readonly
attribute Document        contentDocument;
};
```

Quirks Mode and IE7 Mode (All Versions)

The **contentDocument** attribute of the **HTMLObjectElement** interface is undefined.

Quirks Mode and IE7 Mode (Internet Explorer 7 and Internet Explorer 8)

The **vspace** and **hspace** attributes are of type `string`.

V0038:

The specification states:

```
align of type DOMStringAligns this object (vertically or horizontally) with respect
to its surrounding text. See the align attribute definition in HTML 4.01. This
attribute is deprecated in HTML 4.01.
```

All Document Modes (All Versions)
The **align** attribute of the **HTMLObjectElement** interface is undefined. Its value defaults to `bottom`.

V0039:

The specification states:

```
contentDocument of type Document, readonly, introduced in DOM Level 2
The document this object contains, if there is any and it is available, or null
otherwise.
```

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

The **contentDocument** attribute is not supported on the **HTMLObjectElement** interface.

V0040:

The specification states:

```
Interface HTMLParamElement
Parameters fed to the OBJECT element. See the PARAM element definition in HTML
4.01.
IDL Definition
interface HTMLParamElement : HTMLElement {
attribute DOMString       name;
attribute DOMString       type;
attribute DOMString       value;
attribute DOMString       valueType;
};
```

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

All attributes of the **HTMLParamElement** interface are defined only when they are defined in the markup. For an undefined attribute, **HTMLParamElement** returns `undefined`.

V0041:

The specification states:

```
valueType of type DOMString
Information about the meaning of the value attribute value. See the valuetype
attribute definition in HTML 4.01.
```

Quirks Mode and IE7 Mode (All Versions)

The **valueType** attribute of the **HTMLParamElement** interface returns `undefined`, not the default value `data`. If the **valuetype** attribute of a **PARAM** element is not specified in HTML, the **valueType** attribute of the **HTMLParamElement** interface returns `undefined` as the default value.

V0042:

The specification states:

```
Interface HTMLAppletElement
An embedded Java applet. See the APPLET element definition in HTML 4.01. This
element is deprecated in HTML 4.01.

IDL Definition
interface HTMLAppletElement : HTMLElement {
attribute DOMString       align;
attribute DOMString       alt;
attribute DOMString       archive;
attribute DOMString       code;
attribute DOMString       codeBase;
attribute DOMString       height;

// Modified in DOM Level 2:
```

```
attribute long            hspace;
attribute DOMString       name;
// Modified in DOM Level 2:
attribute DOMString       object;
// Modified in DOM Level 2:
attribute long            vspace;
attribute DOMString       width;
};
```

Quirks Mode, IE7 Mode, IE8 Mode, IE9 Mode, IE10 Mode, and IE11 Mode (All Versions)

The **object** attribute of the **HTMLAppletElement** interface is of type `object`.

V0043:

The specification states:

```
noHref of type boolean
Specifies that this area is inactive, i.e., has no associated action. See the
nohref attribute definition in HTML 4.01.
```

IE7 Mode, IE8 Mode, IE9 Mode, IE10 Mode, and IE11 Mode (All Versions)
The **noHref** attribute of the **HTMLAreaElement** interface is read-only.

V0044:

The specification states:

```
 Interface HTMLTableElement
The create* and delete* methods on the table allow authors to construct and modify
tables. [HTML 4.01] specifies that only one of each of the CAPTION, THEAD, and
TFOOT elements may exist in a table. Therefore, if one exists, and the
createTHead() or createTFoot() method is called, the method returns the existing
THead or TFoot element. See the TABLE element definition in HTML 4.01.

IDL Definition

interface HTMLTableElement : HTMLElement {
// Modified in DOM Level 2:
attribute HTMLTableCaptionElement caption;
// raises(DOMException) on setting
// Modified in DOM Level 2:
attribute HTMLTableSectionElement tHead;
// raises(DOMException) on setting
// Modified in DOM Level 2:
attribute HTMLTableSectionElement tFoot;
// raises(DOMException) on setting
readonly attribute HTMLCollection  rows;
readonly attribute HTMLCollection  tBodies;
attribute DOMString       align;
attribute DOMString       bgColor;
attribute DOMString       border;
attribute DOMString       cellPadding;
attribute DOMString       cellSpacing;
attribute DOMString       frame;
attribute DOMString       rules;
attribute DOMString       summary;
attribute DOMString       width;
HTMLElement        createTHead();
void               deleteTHead();
HTMLElement        createTFoot();
void               deleteTFoot();
HTMLElement        createCaption();
void               deleteCaption();

// Modified in DOM Level 2:
HTMLElement        insertRow(in long index)raises(DOMException);
// Modified in DOM Level 2:
void               deleteRow(in long index)raises(DOMException);
};
```

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

The following methods are not supported on the **HTMLTableElement** interface:

- **createTHead**
- **deleteTHead**
- **createTFoot**
- **deleteTFoot**
- **createCaption**
- **deleteCaption**
- **insertRow**
- **deleteRow**

V0045:

The specification states:

```
    caption of type HTMLTableCaptionElement, modified in DOM Level 2

    Returns the table's CAPTION, or void if none exists.

    Exceptions on setting
    DOMExceptionHIERARCHY_REQUEST_ERR: if the element is not a CAPTION.
```

IE7 Mode and IE8 Mode (All Versions)
The **caption** attribute of the **HTMLTableElement** interface is read-only. If this attribute is set, an error object with an error message of "`Member not found`" and an HRESULT of `-2147352573` is returned.

V0046:

The specification states:

```
    tBodies of type HTMLCollection, readonly

    Returns a collection of the table bodies (including implicit ones).
```

Quirks Mode, IE7 Mode, and IE8 Mode (All Versions)
If a **TABLE** element contains more than one **THEAD** or **TFOOT** element, the **tBodies** attribute of the **HTMLTableElement** interface includes these additional **THEAD** and **TFOOT** elements in its collection of table bodies.

V0047:

The specification states:

```
    tFoot of type HTMLTableSectionElement, modified in DOM Level 2
    Returns the table's TFOOT, or null if none exists.

    Exceptions on setting
    DOMException HIERARCHY_REQUEST_ERR: if the element is not a TFOOT.
```

Quirks Mode, IE7 Mode, IE8 Mode, IE9 Mode, and IE10 Mode (All Versions)
If an object of a type other than HTMLTableSectionElement is assigned to the **tFoot** property, an error object with an error message of "Member not found" and an HRESULT of `-2147352573` is returned.
IE11 Mode (All Versions)
If an object of a type other than HTMLTableSectionElement is assigned to the **tFoot** property, then "Error: No such interface supported" is thrown.

V0048:

The specification states:

```
    tHead of type HTMLTableSectionElement, modified in DOM Level 2

    Returns the table's THEAD, or null if none exists.

    Exceptions on setting
    DOMException HIERARCHY_REQUEST_ERR: if the element is not a THEAD.
```

Quirks Mode, IE7 Mode, IE8 Mode, IE9 Mode, and IE10 Mode (All Versions)
If an object of a type other than HTMLTableSectionElement is assigned to the **tHead** property, an error object with an error message of "Member not found" and an HRESULT of `-2147352573` is returned.
IE11 Mode (All Versions)
If an object of a type other than HTMLTableSectionElement is assigned to the **tHead** property, then "Error: No such interface supported" is thrown.

V0049:

The specification states:

```
    deleteRow modified in DOM Level 2
    Delete a table row.

    Parameters
    index of type long
    The index of the row to be deleted. This index starts from 0 and is relative
    to the logical order (not document order) of all the rows contained inside
    the table. If the index is -1 the last row in the table is deleted.

    Exceptions
    DOMException INDEX_SIZE_ERR: Raised if the specified index is greater
    than or equal to the number of rows or if the index is a negative number
    other than -1.

    No Return Value
```

Quirks Mode, IE7 Mode, and IE8 Mode (All Versions)
If the specified index is greater than or equal to the number of rows or if the index is a negative number other than –1, the **deleteRow** method of the **HTMLTableElement** interface throws a **JSErrors** exception with an error message of "`Invalid argument`" and an HRESULT of `–2147024809 is returned.`

V0052:

The specification states:

```
insertRow modified in DOM Level 2
Insert a new empty row in the table. The new row is inserted immediately before and
in the same section as the current indexth row in the table. If index is -1 or
equal to the number of rows, the new row is appended. In addition, when the table
is empty the row is inserted into a TBODY which is created and inserted into the
table.

Note: A table row cannot be empty according to [HTML 4.01].Parameters index of type
longThe row number where to insert a new row. This index starts from 0 and is
relative to the logical order (not document order) of all the rows contained inside
the table.

Return Value HTMLElement
The newly created row.

Exceptions
DOMException INDEX_SIZE_ERR: Raised if the specified index is greater than
the number of rows or if the index is a negative number other than -1.
```

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

If the specified index is greater than the number of rows or if the index is a negative number other than –1, the **insertRow** method of the **HTMLTableElement** interface throws a **JSError** exception with an error message of `"Invalid argument"` and an HRESULT of `-2147024809`.

V0053:

The specification states:

```
align of type DOMString
Horizontal alignment of data in cells. See the align attribute for HTMLTheadElement
for details.
```

All Document Modes (All Versions)

If the value of the **align** attribute for a **TBODY, THEAD,** or **TFOOT** element is set to `char or justify`, the **align** attribute of the **HTMLTableSectionElement** interface returns an empty string.

V0054:

The specification states:

```
ch of type DOMStringAlignment character for cells in a column. See the char
attribute definition in HTML 4.01.
```

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

The **ch** attribute of the **HTMLTableSectionElement** interface returns an empty string.

V0055:

The specification states:

```
chOff of type DOMStringOffset of alignment character. See the charoff attribute
definition in HTML 4.01.
```

Quirks Mode, IE7 Mode, IE8 Mode, IE9 Mode, IE10 Mode, and IE11 Mode (All Versions)

The **chOff** attribute of the **HTMLTableSectionElement** interface returns an empty string.

V0056:

The specification states:

```
deleteRow modified in DOM Level 2

Delete a row from this section.

Parameters
index of type long
The index of the row to be deleted, or -1 to delete the last row. This index starts
 from 0 and is relative only to the rows contained inside this section, not all the
 rows in the table.

Exceptions
DOMException
INDEX_SIZE_ERR: Raised if the specified index is greater than or equal to the
number of rows or if the index is a negative number other than -1.
```

Quirks Mode, IE7 Mode, and IE8 Mode (All Versions)
If the specified index is greater than or equal to the number of rows or if the index is a negative number other than −1, the **deleteRow** method of the **HTMLTableSectionElement** interface throws a **JSError** exception with an error message of "`Invalid argument`" and an HRESULT of `-2147024809` is returned.

V0057:

The specification states:

```
insertRow modified in DOM Level 2

Insert a row into this section. The new row is inserted immediately before the
current indexth row in this section. If index is -1 or equal to the number of rows
in this section, the new row is appended.

Parameters
index of type long
The row number where to insert a new row. This index starts from 0 and is relative
only to the rows contained inside this section, not all the rows in the table.

Return Value
HTMLElement The newly created row.
```

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

If the specified index is greater than the number of rows or if the index is a negative number other than −1, the **insertRow** method of the **HTMLTableSectionElement** interface throws a **JSError** exception with an error message of "`Invalid argument`" and an HRESULT of `-2147024809` is returned.

V0058:

The specification states:

```
align of type DOMStringHorizontal alignment of data in cell. See the align
attribute definition in HTML 4.01.
```

All Document Modes (All Versions)
If the value of the **align** attribute for a **TR** or **TD** element is set to `char` or `justify`, the **align** attribute of the **HTMLTableCellElement** interface returns an empty string.

V0059:

The specification states:

```
ch of type DOMStringAlignment character for cells in a column. See the char
attribute definition in HTML 4.01.
```

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

The **ch** attribute of the **HTMLTableCellElement** interface has no association with the **char** attribute.

V0060:

The specification states:

```
chOff of type DOMStringOffset of alignment character. See the charoff attribute
definition in HTML 4.01.
```

Quirks and IE7 Mode (All Versions)

The **chOff** attribute of the **HTMLTableCellElement** interface has no association with the **charoff** attribute.

V0061:

The specification states:

```
useMap of type DOMString
Use client-side image map. See the usemap attribute definition in HTML 4.01.
```

All Document Modes (All Versions)
The **useMap** attribute of the **HTMLObjectElement** interface is not supported for a **MAP** element that is a child of an **OBJECT** element.

2.2 Clarifications

The following subsections describe clarifications of the MAY and SHOULD requirements of [DOM Level 2 - HTML].

2.2.1 [DOM Level 2 - HTML] Section 1.5, Objects related to HTML documents

C0001:

The specification states:

```
referrer of type DOMString, readonly
Returns the URI of the page that linked to this page. The value is an empty string
if the user navigated to the page directly (not through a link, but, for example,
via a bookmark).
```

All Document Modes (Internet Explorer 8 and Internet Explorer 9)

The **referrer** attribute of the **HTMLDocument** interface returns an absolute, fully-qualified URI with file name, if present.

All Document Modes (Internet Explorer 7)

The **referrer** attribute of the **HTMLDocument** interface returns an absolute URI without file name, if present.

2.2.2 [DOM Level 2 - HTML] Section 1.6.3, Exposing Element Type Names

C0002:

The specification states:

```
If the document is an HTML 4.01 document the element type names exposed through a
property are in uppercase. For example, the body element type name is exposed
through the tagName property as BODY. If the document is an XHTML 1.0 document the
element name is exposed as it is written in the XHTML file. This means that the
element type names are exposed in lowercase for XHTML documents since the XHTML 1.0
DTDs defines element type names as lowercase, and XHTML, being derived from XML, is
case sensitive.
```

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

XHTML is not supported.

2.2.3 [DOM Level 2 - HTML] Section 1.6.5, Object definitions

C0003:

The specification states:

```
profile of type DOMString
URI designating a metadata profile. See the profile attribute definition in HTML
4.01.
```

Quirks and IE7 Mode (All Versions)

The **profile** attribute that is returned is the same as what the author has set in the **HEAD** element markup.

IE8 Mode and IE9 Mode (All Versions)

The **profile** attribute of the **HTMLHeadElement** interface returns a URI with a trailing slash if missing. For example, "`http://www.contoso.com`" is returned as "`http://www.contoso.com/`"

C0005:

The specification states:

```
aLink of type DOMString
Color of active links (after mouse-button down, but before mouse-button up). See
the alink attribute definition in HTML 4.01. This attribute is deprecated in HTML
4.01.
```

All Document Modes (All Versions)

If the value of the **alink** attribute for a **LINK** element is specified as a color string (such as `black`), the value is converted and stored as a hexadecimal RGB color value in the **aLink** attribute of the **HTMLBodyElement** interface.

C0006:

The specification states:

```
background of type DOMString
URI of the background texture tile image. See the background attribute definition
in HTML 4.01. This attribute is deprecated in HTML 4.01.
```

Quirks and IE7 Mode (All Versions)

The **background** attribute of the **HTMLBodyElement** interface is read as a relative path.

IE8 Mode and IE9 Mode (All Versions)

When the value of the **background** attribute for the **BODY** element is specified in HTML as a relative URI, the value is converted and stored as an absolute URI in the **background** attribute of the **HTMLBodyElement** interface.

C0007:

The specification states:

```
bgColor of type DOMString
Document background color. See the bgcolor attribute definition in HTML 4.01. This
attribute is deprecated in HTML 4.01.
```

Quirks Mode, IE7 Mode, IE8 Mode, IE9 Mode, IE10 Mode, and IE11 Mode (All Versions)

If the value of the **bgcolor** attribute for a **BODY** element is specified as a color string (such as `"black"`), the value is converted and stored as a hexadecimal RGB color value in the **bgColor** attribute of the **HTMLBodyElement** interface.

C0008:

The specification states:

```
link of type DOMString
Color of links that are not active and unvisited. See the link attribute definition
in HTML 4.01. This attribute is deprecated in HTML 4.01.
```

Quirks Mode, IE7 Mode, IE8 Mode, IE9 Mode, IE10 Mode, and IE11 Mode (All Versions)

If the value of the **link** attribute for a **BODY** element is specified in HTML as a color string (such as `black`), the value is converted and stored as a hexadecimal RGB color value in the **link** attribute of the **HTMLBodyElement** interface.

C0009:

The specification states:

```
vLink of type DOMString
Color of links that have been visited by the user. See the vlink attribute
definition in HTML 4.01. This attribute is deprecated in HTML 4.01.
```

Quirks Mode, IE7 Mode, IE8 Mode, IE9 Mode, IE10 Mode, and IE11 Mode (All Versions)

If the value of the **vlink** attribute for a **BODY** element is specified in HTML as a color string (such as `black`), the value is converted and stored as a hexadecimal RGB color value in the **vLink** attribute of the **HTMLBodyElement** interface.

C0010:

The specification states:

```
action of type DOMString
Server-side form handler. See the action attribute definition in HTML 4.01.
```

Quirks and IE7 Mode (All Versions)

The value of the **action** attribute of the **HTMLFormElement** interface is preserved and is not converted into an absolute URI.

IE8 Mode and IE9 Mode (All Versions)

When the value of the **action** attribute for the **FORM** element is specified in HTML as a relative URI, the value is converted and stored as an absolute URI in the **action** attribute of the **HTMLFormElement** interface.

C0011:

The specification states:

```
length of type unsigned long, modified in DOM Level 2
The number of options in this SELECT.

Exceptions on setting
DOMException   NOT_SUPPORTED_ERR: if setting the length is not allowed by the
implementation.
```

All Document Modes (All Versions)

Setting the value of the **length** attribute is allowed. The number of options in this **SELECT** element is increased or decreased (added or removed from the end of the options collection) as required.

C0012:

The specification states:

```
selectedIndex of type long
The ordinal index of the selected option, starting from 0. The value -1 is returned
if no element is selected. If multiple options are selected, the index of the first
selected option is returned.
```

All Document Modes (All Versions)

By default, the first option of a **SELECT** element is selected (the **selected** property on the first **OPTION** element is set to TRUE). If the multiple attribute of the **SELECT** element is set, the default value of **selectedIndex** is -1. If the multiple attribute is not set, **selectedIndex** never returns -1.

C0013:

The specification states:

```
size of type long
Number of visible rows. See the size attribute definition in HTML 4.01.
```

All Document Modes (All Versions)

If no value is specified for the **size** attribute of the **SELECT** element (or the value is not a number) then the **size** attribute of the **HTMLSelectElement** interface is assigned the value `0`.

C0014:

The specification states:

```
tabIndex of type long
Index that represents the element's position in the tabbing order. See the tabindex
attribute definition in HTML 4.01.
```

Quirks and IE7 Mode (All Versions)

The **tabIndex** attribute of the **HTMLSelectElement** interface accepts any invalid value. The value of the attribute defaults to `0` if no value is specified in HTML.

IE8 Mode and IE9 Mode (All Versions)

If no value, or an invalid value, is specified for the **tabindex** attribute of a **SELECT** element, the **tabIndex** attribute of the **HTMLSelectElement** interface defaults to a value of `0`.

C0015:

The specification states:

```
Interface HTMLInputElement
Form control.
Note: Depending upon the environment in which the page is being viewed, the value
property may be read-only for the file upload input type. For the "password" input
type, the actual value returned may be masked to prevent unauthorized use.
```

All Document Modes (All Versions)

The **value** attribute of the **HTMLInputElement** interface is read-only when its value is set to `file`. When its value is set to `password`, the password value is not masked.

All Document Modes (All Versions)

Setting the **type** attribute of the **HTMLInputElement** interface to an invalid value throws a JScript **Error** object with a description of "Invalid argument" and an error number of 0x80070057 (-2147024809).

C0016:

The specification states:

```
checked of type boolean
When the type attribute of the element has the value "radio" or "checkbox", this
represents the current state of the form control, in an interactive user agent.
Changes to this attribute change the state of the form control, but do not change
the value of the HTML checked attribute of the INPUT element. Note: During the
handling of a click event on an input element with a type attribute that has the
value "radio" or "checkbox", some implementations may change the value of this
property before the event is being dispatched in the document. If the default
action of the event is canceled, the value of the property may be changed back to
its original value. This means that the value of this property during the handling
of click events is implementation dependent.
```

All Document Modes (All Versions)

If the default action of the click event is canceled, the value of the **checked** attribute of the **HTMLInputElement** interface is changed back to its original value.
The value of the **checked** attribute of the **HTMLInputElement** interface changes before a click event is dispatched.

C0018:

The specification states:

```
shape of type DOMString
The shape of the active area. The coordinates are given by coords. See the shape
attribute definition in HTML 4.01.
```

IE8 Mode and IE9 Mode (All Versions)

If the value of the **shape** attribute for the **A** element is set to `default`, the **shape** attribute of the **HTMLAnchorElement** interface is set to `rect`.

C0019:

The specification states:

```
longDesc of type DOMString
URI designating a long description of this image or frame. See the longdesc
attribute definition in HTML 4.01.
```

IE8 Mode and IE9 Mode (All Versions)

When the value of the **longdesc** attribute for the **IMAGE** element is specified in HTML as a relative URI, the value is converted and stored as an absolute URI in the **longDesc** attribute of the **HTMLImageElement** interface.

C0020:

The specification states:

```
src of type DOMString
URI designating the source of this image. See the src attribute definition in HTML
4.01.
```

IE8 Mode and IE9 Mode (All Versions)

When the value of the **src** attribute for the **IMAGE** element is specified in HTML as a relative URI, the value is converted and stored as an absolute URI in the **src** attribute of the **HTMLImageElement**.

C0021:

The specification states:

```
data of type DOMString
A URI specifying the location of the object's data. See the data attribute
definition in HTML 4.01.
```

Quirks Mode and IE7 Mode (All Versions)
The **data** attribute of the **HTMLObjectElement** interface returns a URI as a relative path.
IE8 Mode and IE9 Mode (All Versions)

The **data** attribute of the **HTMLObjectElement** interface returns a URI as an absolute path.

C0022:

The specification states:

```
standby of type DOMString
Message to render while loading the object. See the standby attribute definition in
HTML 4.01.
```

IE7 Mode, IE8 Mode, IE9 Mode, IE10 Mode, and IE11 Mode (All Versions)

The **standby** attribute is not supported on the **HTMLObjectElement** interface.

C0024:

The specification states:

```
href of type DOMString
The URI of the linked resource. See the href attribute definition in HTML 4.01.
```

All Document Modes (All Versions)
A slash (/) is appended to the end of the URI if a file name is not specified.
IE8 Mode and IE9 Mode (All Versions)
A slash (/) is appended to the end of the **href** attribute of the **HTMLAreaElement** interface when the **href** attribute does not end with a slash.

C0025:

The specification states:

```
interface HTMLAreaElement : HTMLElement
shape of type DOMString.
The shape of the active area. The coordinates are given by coords. See the shape attribute
definition in HTML 4.01.
```

Quirks Mode and IE7 Mode (All Versions)

The value of the **shape** attribute of the **HTMLAreaElement** interface is converted to uppercase.

IE8 Mode and IE9 Mode (All Versions)

If the value of the **shape** attribute for the **AREA** element is set to `default` in markup, the **shape** attribute of the **HTMLAreaElement** interface is set to `rect`.

C0026:

The specification states:

```
src of type DOMString
URI designating an external script.
```

IE8 Mode and IE9 Mode (All Versions)

When the value of the **src** attribute for the **SCRIPT** element is specified in HTML as a relative URI, the value is converted and stored as an absolute URI in the **src** attribute of the **HTMLScriptElement** interface.

C0027:

The specification states:

```
bgColor of type DOMString
```

```
Cell background color. See the bgcolor attribute definition in HTML 4.01. This attribute is
deprecated in HTML 4.01.
```

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

If the value of the **bgcolor** attribute for a **TABLE** element is specified as a color string (such as `black`), the value is converted and stored as a hexadecimal RGB color value in the **bgColor** attribute of the **HTMLTableElement** interface.

C0030:

The specification states:

```
bgColor of type DOMString
Cell background color. See the bgcolor attribute
definition in HTML 4.01. This attribute is deprecated in HTML 4.01.
```

Quirks Mode, IE7 Mode, and IE8 Mode (All Versions)
If the value of the **bgcolor** attribute for a **TH** or **TD** element is specified as a color string (such as `black`), the value is converted and stored as a hexadecimal RGB color value in the **bgColor** attribute of the **HTMLTableCellElement** interface.

C0031:

The specification states:

```
height of type DOMStringCell height. See the height attribute definition in HTML
4.01. This attribute is deprecated in HTML 4.01.
```

All Document Modes (All Versions)
The **height** attribute of the **HTMLTableCellElement** interface is serialized with the `"px"` suffix removed if present.

C0033:

The specification states:

```
longDesc of type DOMString
URI [IETF RFC 2396] designating a long description of
this image or frame. See the longdesc attribute definition in HTML 4.01.
```

IE8 Mode and IE9 Mode (All Versions)

When the value of the **longdesc** attribute for the **IFRAME** element is specified in HTML as a relative URI, the value is converted and stored as an absolute URI in the **longDesc** attribute of the **HTMLIFrameElement** interface.

C0034:

The specification states:

```
src of type DOMString
A URI [IETF RFC 2396] designating the initial frame contents.
 See the src attribute definition in HTML 4.01.
```

IE8 Mode and IE9 Mode (All Versions)

When the value of the **src** attribute for the **IFRAME** element is specified in HTML as a relative URI, the value is converted and stored as an absolute URI in the **src** attribute of the **HTMLIFrameElement** interface.

2.3 Extensions

The following subsections describe extensions to the requirements of [DOM Level 2 - HTML].

2.3.1 [DOM Level 2 - HTML] Section 1.6.5, Object definitions

E0001:

The specification states:

```
form of type HTMLFormElement, readonly
Returns the FORM element containing this control. Returns null if this control is
not within the context of a form.
```

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

The value of the **form** attribute of the **HTMLTextAreaElement** interface can be set.

E0002:

The specification states:

```
contentDocument of type Document, readonly, introduced in DOM Level 2
The document this frame contains, if there is any and it is available, or null
otherwise.
```

Quirks and IE7 Mode (All Versions)
The value of the **contentDocument** attribute of the **HTMLFrameElement** interface can be set.

2.4 Error Handling

There are no additional error handling considerations.

2.5 Security

There are no additional security considerations.

3 Change Tracking

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

4 Index