

## [MS-DOM2S]:

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

---

### Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation for protocols, file formats, languages, standards as well as overviews of the interaction among each of these technologies.
- **Copyrights.** This documentation is covered by Microsoft copyrights. Regardless of any other terms that are contained in the terms of use for the Microsoft website that hosts this documentation, you may make copies of it in order to develop implementations of the technologies described in the Open Specifications and may distribute portions of it in your implementations using these technologies or your documentation as necessary to properly document the implementation. You may also distribute in your implementation, with or without modification, any schema, IDL's, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications.
- **No Trade Secrets.** Microsoft does not claim any trade secret rights in this documentation.
- **Patents.** Microsoft has patents that may cover your implementations of the technologies described in the Open Specifications. Neither this notice nor Microsoft's delivery of the documentation grants any licenses under those or any other Microsoft patents. However, a given Open Specification may be covered by Microsoft [Open Specification Promise](#) or the [Community Promise](#). If you would prefer a written license, or if the technologies described in the Open Specifications are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting [iplg@microsoft.com](mailto:iplg@microsoft.com).
- **Trademarks.** The names of companies and products contained in this documentation may be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights.
- **Fictitious Names.** The example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted in this documentation are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

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

**Tools.** The Open Specifications do 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 are intended for use in conjunction with publicly available standard specifications and network programming art, and assumes that the reader either is familiar with the aforementioned material or has immediate access to it.

## Revision Summary

Date	Revision History	Revision Class	Comments
03/17/2010	0.1	New	Released new document.
03/26/2010	1.0	None	Introduced no new technical or language changes.
05/26/2010	1.2	None	Introduced no new technical or language changes.
09/08/2010	1.3	Major	Significantly changed the technical content.
10/13/2010	1.4	Minor	Clarified the meaning of the technical content.
02/10/2011	2.0	Minor	Clarified the meaning of the technical content.
02/22/2012	3.0	Major	Significantly changed the technical content.
07/25/2012	3.1	Minor	Clarified the meaning of the technical content.

# Table of Contents

<b>1 Introduction .....</b>	<b>4</b>
1.1 Glossary .....	4
1.2 References.....	4
1.2.1 Normative References.....	4
1.2.2 Informative References .....	4
1.3 Microsoft Implementations.....	4
1.4 Standards Support Requirements .....	5
1.5 Notation .....	5
<b>2 Standards Support Statements.....</b>	<b>7</b>
2.1 Normative Variations.....	7
2.2 Clarifications .....	7
2.2.1 [DOM Level 2 - Style] Section 1.2, Interface StyleSheet (introduced in DOM Level 2).....	7
2.2.2 [DOM Level 2 - Style] Section 1.3, Interface LinkStyle (introduced in DOM Level 2).....	8
2.2.3 [DOM Level 2 - Style] Section 1.4, HTML and Style Sheet Creation.....	9
2.2.4 [DOM Level 2 - Style] Section 2.2, Interface CSSStyleSheet (introduced in DOM Level 2) .....	9
2.2.5 [DOM Level 2 - Style] Section 2.2.1, Interface ViewCSS (introduced in DOM Level 2).....	16
2.2.6 [DOM Level 2 - Style] Section 2.2.2, Interface DOMImplementationCSS (introduced in DOM Level 2) .....	16
2.2.7 [DOM Level 2 - Style] Section 2.3, Interface CSS2Properties (introduced in DOM Level 2) .....	17
2.3 Error Handling.....	38
2.4 Security.....	38
<b>3 Change Tracking.....</b>	<b>39</b>
<b>4 Index .....</b>	<b>41</b>

# 1 Introduction

This document describes the level of support provided by Windows® Internet Explorer® for the *Document Object Model (DOM) Level 2 Style Specification Version 1.0* [\[DOM Level 2 - Style\]](#), W3C Recommendation November 13, 2000.

The [\[DOM Level 2 - Style\]](#) 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 described in [\[RFC2119\]](#). All statements of optional behavior use either MAY, SHOULD, or SHOULD NOT.

## 1.2 References

References to Microsoft Open Specifications documentation do not include a publishing year because links are to the latest version of the technical documents, which are updated frequently. References to other documents include a publishing year when one is available.

### 1.2.1 Normative References

We conduct frequent surveys of the normative references to assure their continued availability. If you have any issue with finding a normative reference, please contact [dochelp@microsoft.com](mailto:dochelp@microsoft.com). We will assist you in finding the relevant information. Please check the archive site, <http://msdn2.microsoft.com/en-us/library/E4BD6494-06AD-4aed-9823-445E921C9624>, as an additional source.

[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/>

[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 Microsoft products implement some portion of [\[DOM Level 2 - Style\]](#):

- Windows® Internet Explorer® 7
- Windows® Internet Explorer® 8
- Windows® Internet Explorer® 9
- Windows® Internet Explorer® 10

In addition, each version of Windows® Internet Explorer® implements multiple document modes, which can vary individually in their support of the standard. The following table lists the document modes available in each version of Internet Explorer.

Browser version	Document modes supported
Internet Explorer 7	Quirks mode Standards mode
Internet Explorer 8	Quirks mode IE7 mode IE8 mode
Internet Explorer 9	Quirks mode IE7 mode IE8 mode IE9 mode
Internet Explorer 10 (Beta)	Quirks mode IE7 mode IE8 mode IE9 mode IE10 Mode

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

**Note** "Standards mode" in Internet Explorer 7 and "IE7 mode" in Internet Explorer 8 refer to 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 [\[DOM Level 2 - Style\]](#) 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 [\[DOM Level 2 - Style\]](#) and whether they are considered normative or informative.

Sections	Normative/Informative
1-4	Normative
Appendix A-D	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 <a href="#">RFC2119</a> .) 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 a full list of variations and clarifications points in the Microsoft implementation of [\[DOM Level 2 - Style\]](#).

- Section [2.1](#) includes only those variations that violate a MUST requirement in the target specification.
- Section [2.2](#) describes further variations from MAY and SHOULD requirements.
- Section [2.3](#) identifies variations in error handling.
- Section [2.4](#) identifies variations that impact security.

### 2.1 Normative Variations

There are no variations from [\[DOM Level 2 - Style\]](#).

### 2.2 Clarifications

The following subsections identify clarifications to recommendations made by [\[DOM Level 2 - Style\]](#).

#### 2.2.1 [DOM Level 2 - Style] Section 1.2, Interface StyleSheet (introduced in DOM Level 2)

V0001:

The specification states:

```
href of type DOMString, readonly
If the style sheet is a linked style sheet, the value of its attribute is its
location. For inline style sheets, the value of this attribute is null.
```

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

The **href** attribute of the **StyleSheet** interface is not treated as read-only. It is possible to get and set the **href** attribute. If **href** is set to a new value, the new style sheet is read and applied to the document in place of the original style sheet.

The **href** attribute for an inline style sheet is returned as an empty string, instead of null.

V0002:

The specification states:

```
media of type MediaList, readonly
The intended destination media for style information. The media is often specified
in the ownerNode. If no media has been specified, the MediaList will be empty.
```

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

The **media** attribute of the **StyleSheet** interface has read/write access; it is of type string, not **MediaList**.

V0003:

The specification states:

```
ownerNode of type Node, readonly
The node that associates this style sheet with the document. For HTML, this may be
the corresponding LINK or STYLE element. For XML, it may be the linking processing
instruction. For style sheets that are included by other style sheets, the value of
this attribute is null.
```

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

The **ownerNode** attribute of the **StyleSheet** interface is not supported. Similar functionality is provided by the **owningElement** attribute.

V0004:

The specification states:

```
title of type DOMString, readonly
The advisory title. The title is often specified in the ownerNode.
```

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

The **title** attribute of the **StyleSheet** interface has read/write access.

V0005:

The specification states:

```
The MediaList interface provides the abstraction of an ordered collection of media,
without defining or constraining how this collection is implemented. An empty list
is the same as a list that contains the medium "all".
The items in the MediaList are accessible via an integral index, starting from 0.
```

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

The **MediaList** interface is not supported. The **media** attribute of the **StyleSheet** interface is of type string, not **MediaList**.

## 2.2.2 [DOM Level 2 - Style] Section 1.3, Interface LinkStyle (introduced in DOM Level 2)

V0006:

The specification states:

```
The LinkStyle interface provides a mechanism by which a style sheet can be
retrieved from the node responsible for linking it into a document. An instance of
the LinkStyle interface can be obtained using binding-specific casting methods on
an instance of a linking node (HTMLLinkElement, HTMLStyleElement or
ProcessingInstruction in DOM Level 2).
```

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

The **LinkStyle** interface is not supported.



### 2.2.3 [DOM Level 2 - Style] Section 1.4, HTML and Style Sheet Creation

V0007:

The specification states:

The underlying style sheet will be created after the element is inserted into the document and the type attribute is set in a way indicating that the element corresponds to a style sheet language interpreted by the user agent.

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

Inserting a style sheet by creating a new **STYLE** element is supported, but rules can be edited only through the **Rules** collection of a **styleSheet** object.

V0008:

The specification states:

HTML and Style Sheet Removal  
Removing a LINK HTML element or a STYLE HTML element removes the underlying style sheet from the style sheet collection associated with a document. Specifically, the removed style sheet is no longer applied to the presentation of the document.

*All Document Modes (Internet Explorer 7)*

When removing a **LINK** element or a **STYLE** element from a **document** object through a script, the underlying **styleSheet** is removed from the style sheet collection associated with a document (**document.styleSheets**). However, the style is still applied to the presentation of the document.

### 2.2.4 [DOM Level 2 - Style] Section 2.2, Interface CSSStyleSheet (introduced in DOM Level 2)

V0009:

The specification states:

The CSSStyleSheet interface is a concrete interface used to represent a CSS style sheet i.e., a style sheet whose content type is "text/css".

```
IDL Definition
// Introduced in DOM Level 2:
interface CSSStyleSheet : stylesheets::StyleSheet {
    readonly attribute CSSRule          ownerRule;
    readonly attribute CSSRuleList      cssRules;
    unsigned long          insertRule(in DOMString rule,
                                     in unsigned long index)
                                     raises(DOMException);
    void                   deleteRule(in unsigned long index)
                                     raises(DOMException);
};
```

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

The following standard attributes or methods are not supported:

- **cssRules**
- **ownerRule**
- **insertRule**
- **deleteRule**

The following additional attributes or methods are supported:

- **readonly**
- **id**
- **imports**
- **pages**
- **rules**
- **addImport**
- **addPageRule**
- **addRule**
- **removeImport**
- **removeRule**

*IE8 Mode (All Versions)*

The additional attributes **isAlternate** and **isPrefAlternate** are supported.

V0010:

The specification states:

The CSSRule interface is the abstract base interface for any type of CSS statement. This includes both rule sets and at-rules. An implementation is expected to preserve all rules specified in a CSS style sheet, even if the rule is not recognized by the parser. Unrecognized rules are represented using the CSSUnknownRule interface."

IDL Definition

// Introduced in DOM Level 2:

```
interface CSSRule {

    // RuleType
    const unsigned short UNKNOWN_RULE = 0;
    const unsigned short STYLE_RULE = 1;
    const unsigned short CHARSET_RULE = 2;
    const unsigned short IMPORT_RULE = 3;
    const unsigned short MEDIA_RULE = 4;
    const unsigned short FONT_FACE_RULE = 5;
    const unsigned short PAGE_RULE = 6;

    readonly attribute unsigned short type;
    attribute DOMString cssText;
```

```

// raises(DOMException) on setting

readonly attribute CSSStyleSheet parentStyleSheet;
readonly attribute CSSRule parentRule;
};

```

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

The following variations apply:

- The **CSSRule** interface is not supported.
- The **rule** object can be used to represent a style within a style sheet that consists of a selector and one or more declarations.
- The **page** object represents an @page rule within a **styleSheet** object.
- The **imports** object represents an @import rule within a **styleSheet** object.

#### *IE9 Mode and IE10 Mode (All Versions)*

The **CSSRule** interface is supported, but the **UNKNOWN\_RULE** and **CHARSET\_RULE** rule types are not supported.

Setting the **cssText** attribute does not raise a **DOMException** exception. The command to set the **cssText** attribute is ignored.

V0011:

The specification states:

The CSSStyleRule interface represents a single rule set in a CSS style sheet.

```

IDL Definition
// Introduced in DOM Level 2:
interface CSSStyleRule : CSSRule {
    attribute DOMString selectorText;
    // raises(DOMException) on setting

    readonly attribute CSSStyleDeclaration style;
};

```

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

The following variations apply:

- The **CSSStyleRule** interface does not support the inherited properties of the **CSSRule** interface.
- The **readOnly** property is an additional supported property.
- The **selectorText** property is read-only.

#### *IE9 Mode and IE10 Mode (All Versions)*

Setting **selectorText** does not raise a **DOMException**. The command is ignored.

V0012:

## The specification states:

The `CSSMediaRule` interface represents a `@media` rule in a CSS style sheet. A `@media` rule can be used to delimit style rules for specific media types.

```
IDL Definition
// Introduced in DOM Level 2:
interface CSSMediaRule : CSSRule {
    readonly attribute StyleSheet::MediaList media;
    readonly attribute CSSRuleList cssRules;
    unsigned long insertRule(in DOMString rule,
                             in unsigned long index)
        raises(DOMException);
    void deleteRule(in unsigned long index)
        raises(DOMException);
};
```

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

The **CSSMediaRule** interface is not supported.

V0013:

## The specification states:

The `CSSFontFaceRule` interface represents a `@font-face` rule in a CSS style sheet. The `@font-face` rule is used to hold a set of font descriptions.

```
IDL Definition
// Introduced in DOM Level 2:
interface CSSFontFaceRule : CSSRule {
    readonly attribute CSSStyleDeclaration style;
};
```

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

The **CSSFontFaceRule** interface is not supported.

V0014:

## The specification states:

The `CSSPageRule` interface represents a `@page` rule within a CSS style sheet. The `@page` rule is used to specify the dimensions, orientation, margins, etc. of a page box for paged media.

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

The **CSSPageRule** interface is not supported.

The **page** object represents an **@page** rule within a style sheet.

V0015:

## The specification states:

The `CSSImportRule` interface represents a `@import` rule within a CSS style sheet. The `@import` rule is used to import style rules from other style sheets.

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

The **CSSImportRule** interface is not supported.

The **imports** collection represents an **@import** rule within a style sheet.

V0016:

The specification states:

The `CSSCharsetRule` interface represents a `@charset` rule in a CSS style sheet. The value of the encoding attribute does not affect the encoding of text data in the DOM objects; this encoding is always UTF-16. After a stylesheet is loaded, the value of the encoding attribute is the value found in the `@charset` rule. If there was no `@charset` in the original document, then no `CSSCharsetRule` is created. The value of the encoding attribute may also be used as a hint for the encoding used on serialization of the style sheet. The value of the `@charset` rule (and therefore of the `CSSCharsetRule`) may not correspond to the encoding the document actually came in; character encoding information e.g. in an HTTP header, has priority (see CSS document representation) but this is not reflected in the `CSSCharsetRule`.

### *All Document Modes (All Versions)*

The **CSSCharsetRule** interface is not supported.

V0017:

The specification states:

The `CSSUnknownRule` interface represents an at-rule not supported by this user agent.

### *All Document Modes (All Versions)*

The **CSSUnknownRule** interface is not supported.

V0018:

The specification states:

The `CSSStyleDeclaration` interface represents a single CSS declaration block. This interface may be used to determine the style properties currently set in a block or to set style properties explicitly within the block. While an implementation may not recognize all CSS properties within a CSS declaration block, it is expected to provide access to all specified properties in the style sheet through the `CSSStyleDeclaration` interface. Furthermore, implementations that support a specific level of CSS should correctly handle CSS shorthand properties for that level. For a further discussion of shorthand properties, see the `CSS2Properties` interface. This interface is also used to provide a read-only access to the computed values of an element. See also the `ViewCSS` interface. Note: The CSS Object Model doesn't provide an access to the specified or actual values of the CSS cascade.

```

IDL Definition
// Introduced in DOM Level 2:
interface CSSStyleDeclaration {
    attribute DOMString      cssText;
                                // raises(DOMException) on setting

    DOMString      getPropertyValue(in DOMString propertyName);
    CSSValue      getPropertyCSSValue(in DOMString propertyName);
    DOMString      removeProperty(in DOMString propertyName)
                                raises(DOMException);
    DOMString      getPropertyPriority(in DOMString propertyName);
    void      setProperty(in DOMString propertyName,
                        in DOMString value,
                        in DOMString priority)
                                raises(DOMException);
    readonly attribute unsigned long      length;
    DOMString      item(in unsigned long index);
    readonly attribute CSSRule      parentRule;
};

```

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

The **cssText** attribute is the only attribute supported in the **CSSStyleDeclaration** interface.

#### *IE9 Mode and IE10 Mode (All Versions)*

The **getPropertyPriority** method returns an empty string for the following shorthand properties even if they have a priority (for example, "important"):

- **border-color**
- **border-style**
- **border-top**
- **border-right**
- **border-bottom**
- **border-left**
- **border-width**
- **font**
- **list-style**
- **margin**
- **outline**
- **padding**

The **getPropertyCSSValue** method is not supported.

V0019:

The specification states:

The `CSSValue` interface represents a simple or a complex value. A `CSSValue` object only occurs in a context of a CSS property.

*All Document Modes (All Versions)*

The **CSSValue** interface is not supported.

V0020:

The specification states:

The `CSSPrimitiveValue` interface represents a single CSS value. This interface may be used to determine the value of a specific style property currently set in a block or to set a specific style property explicitly within the block.

*All Document Modes (All Versions)*

The **CSSPrimitiveValue** interface is not supported.

V0021:

The specification states:

The `CSSValueList` interface provides the abstraction of an ordered collection of CSS values.

*All Document Modes (All Versions)*

The **CSSValueList** interface is not supported.

V0022:

The specification states:

The `RGBColor` interface is used to represent any RGB color value. This interface reflects the values in the underlying style property.

*All Document Modes (All Versions)*

The **RGBColor** interface is not supported.

V0023:

The specification states:

The `Rect` interface is used to represent any rect value. This interface reflects the values in the underlying style property.

*All Document Modes (All Versions)*

The **Rect** interface is not supported.

V0024:

The specification states:

The Counter interface is used to represent any counter or counters function value.  
This interface reflects the values in the underlying style property.

*All Document Modes (All Versions)*

The **Counter** interface is not supported.

## 2.2.5 [DOM Level 2 - Style] Section 2.2.1, Interface ViewCSS (introduced in DOM Level 2)

V0025:

The specification states:

Interface ViewCSS (introduced in DOM Level 2)  
This interface represents a CSS view. The getComputedStyle method provides a read only access to the computed values of an element.

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

The **ViewCSS** interface is not supported.

*IE9 Mode and IE10 Mode (All Versions)*

The **getComputedStyle** method incorrectly returns "0% 0%" when the style of the element to be computed has the value of `inherit` for the **background-position** property.

V0026:

The specification states:

Interface DocumentCSS (introduced in DOM Level 2)  
This interface represents a document with a CSS view.

*All Document Modes (All Versions)*

The **DocumentCSS** interface is not supported.

## 2.2.6 [DOM Level 2 - Style] Section 2.2.2, Interface DOMImplementationCSS (introduced in DOM Level 2)

V0027:

The specification states:

Interface DOMImplementationCSS (introduced in DOM Level 2).  
This interface allows the DOM user to create a CSSStyleSheet outside the context of a document. There is no way to associate the new CSSStyleSheet with a document in DOM Level 2



The **DOMImplementationCSS** interface is not supported.

## **2.2.7 [DOM Level 2 - Style] Section 2.3, Interface CSS2Properties (introduced in DOM Level 2)**

E0001:

The specification states:

```
Interface CSS2Properties (introduced in DOM Level 2)
The CSS2Properties interface represents a convenience mechanism for retrieving and
setting properties within a CSSStyleDeclaration. The attributes of this interface
correspond to all the properties specified in CSS2. Getting an attribute of this
interface is equivalent to calling the getPropertyValue method of the
CSSStyleDeclaration interface. Setting an attribute of this interface is equivalent
to calling the setProperty method of the CSSStyleDeclaration interface.
```

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

The **filter** property is an additional property that is supported.

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

The following additional properties are supported:

- **accelerator**
- **backgroundPositionX**
- **backgroundPositionY**
- **behavior**
- **imeMode**
- **layoutFlow**
- **layoutGrid**
- **layoutGridChar**
- **layoutGridLine**
- **layoutGridMode**
- **layoutGridType**
- **lineBreak**
- **msInterpolationMode**
- **overflow**
- **overflowY**
- **pixelBottom**

- **pixelHeight**
- **pixelLeft**
- **pixelRight**
- **pixelTop**
- **pixelWidth**
- **posBottom**
- **posHeight**
- **posLeft**
- **posRight**
- **posTop**
- **posWidth**
- **rubyAlign**
- **rubyOverhang**
- **rubyPosition**
- **scrollbar3dLightColor**
- **scrollbarArrowColor**
- **scrollbarBaseColor**
- **scrollbarDarkShadowColor**
- **scrollbarFaceColor**
- **scrollbarHighlightColor**
- **scrollbarShadowColor**
- **scrollbarTrackColor**
- **styleFloat**
- **textAlignLast**
- **textAutospace**
- **textDecorationBlink**
- **textDecorationLineThrough**
- **textDecorationNone**
- **textDecorationOverline**
- **textDecorationUnderline**

- **textJustify**
- **textJustifyTrim**
- **textKashida**
- **textKashidaSpace**
- **textOverflow**
- **textUnderlinePosition**
- **unicodeBidi**
- **wordbreak**
- **wordWrap**
- **writingMode**
- **zoom**

*IE8 Mode (All Versions)*

The following additional properties are supported:

- **msBlockProgression**
- **msBoxSizing**

*IE9 Mode and IE10 Mode (All Versions)*

The following additional properties are supported:

- **accelerator**
- **alignmentBaseline**
- **backgroundClip**
- **backgroundOrigin**
- **backgroundPositionX**
- **backgroundPositionY**
- **backgroundSize**
- **baselineShift**
- **behavior**
- **borderBottomLeftRadius**
- **borderBottomRightRadius**
- **borderRadius**
- **borderTopLeftRadius**

- **borderTopRightRadius**
- **boxShadow**
- **boxSizing**
- **clipPath**
- **clipRule**
- **dominantBaseline**
- **fill**
- **fillOpacity**
- **fillRule**
- **filter**
- **glyphOrientationHorizontal**
- **glyphOrientationVertical**
- **imeMode**
- **kerning**
- **layoutFlow**
- **layoutGrid**
- **layoutGridChar**
- **layoutGridLine**
- **layoutGridMode**
- **layoutGridType**
- **lineBreak**
- **marker**
- **markerEnd**
- **markerMid**
- **markerStart**
- **mask**
- **msBlockProgression**
- **msBoxAlign**
- **msBoxDirection**
- **msBoxFlex**

- **msBoxFlexGroup**
- **msBoxLineProgression**
- **msBoxLines**
- **msBoxOrdinalGroup**
- **msBoxOrient**
- **msBoxPack**
- **msGetPropertyEnabled**
- **msInterpolationMode**
- **msPutPropertyEnabled**
- **msTransform**
- **msTransformOrigin**
- **opacity**
- **overflowX**
- **overflowY**
- **pixelBottom**
- **pixelHeight**
- **pixelLeft**
- **pixelRight**
- **pixelTop**
- **pixelWidth**
- **pointerEvents**
- **posBottom**
- **posHeight**
- **posLeft**
- **posRight**
- **posTop**
- **posWidth**
- **rubyAlign**
- **rubyOverhang**
- **rubyPosition**

- **scrollbar3dLightColor**
- **scrollbarArrowColor**
- **scrollbarBaseColor**
- **scrollbarDarkShadowColor**
- **scrollbarFaceColor**
- **scrollbarHighlightColor**
- **scrollbarShadowColor**
- **scrollbarTrackColor**
- **stopColor**
- **stopOpacity**
- **stroke**
- **strokeDasharray**
- **strokeDashoffset**
- **strokeLinecap**
- **strokeLinejoin**
- **strokeMiterlimit**
- **strokeOpacity**
- **strokeWidth**
- **styleFloat**
- **textAlignLast**
- **textAnchor**
- **textAutospace**
- **textDecorationBlink**
- **textDecorationLineThrough**
- **textDecorationNone**
- **textDecorationOverline**
- **textDecorationUnderline**
- **textJustify**
- **textJustifyTrim**
- **textKashida**

- **textKashidaSpace**
- **textOverflow**
- **textUnderlinePosition**
- **wordBreak**
- **wordWrap**
- **writingMode**
- **zoom**

*IE10 Mode (All Versions)*

The following additional properties are supported:

- **breakAfter**
- **breakBefore**
- **breakInside**
- **colorInterpolationFilters**
- **columnFill**
- **columnGap**
- **columnRule**
- **columnRuleColor**
- **columnRuleStyle**
- **columnRuleWidth**
- **columns**
- **columnSpan**
- **columnWidth**
- **enableBackground**
- **floodColor**
- **floodOpacity**
- **lightingColor**
- **msAnimation**
- **msAnimationDelay**
- **msAnimationDirection**
- **msAnimationDuration**

- **msAnimationFillMode**
- **msAnimationIterationCount**
- **msAnimationName**
- **msAnimationPlayState**
- **msAnimationTimingFunction**
- **msBackfaceVisibility**
- **msContentZoomBoundary**
- **msContentZoomBoundaryMax**
- **msContentZoomBoundaryMin**
- **msContentZoomChaining**
- **msContentZooming**
- **msContentZoomSnap**
- **msContentZoomSnapPoints**
- **msContentZoomSnapType**
- **msFlowFrom**
- **msFlowInto**
- **msFontFeatureSettings**
- **msGridColumn**
- **msGridColumnAlign**
- **msGridColumns**
- **msGridColumnSpan**
- **msGridLayer**
- **msGridRow**
- **msGridRowAlign**
- **msGridRows**
- **msGridRowSpan**
- **msHighContrastAdjust**
- **msHyphenateLimitChars**
- **msHyphenateLimitLines**
- **msHyphenateLimitZone**



- **msHyphens**
- **msOverflowStyle**
- **msPerspective**
- **msPerspectiveOrigin**
- **msScrollBoundary**
- **msScrollBoundaryBottom**
- **msScrollBoundaryLeft**
- **msScrollBoundaryRight**
- **msScrollBoundaryTop**
- **msScrollChaining**
- **msScrollRails**
- **msScrollSnapPointsX**
- **msScrollSnapPointsY**
- **msScrollSnapType**
- **msScrollSnapX**
- **msScrollSnapY**
- **msTouchAction**
- **msTransformStyle**
- **msTransition**
- **msTransitionDelay**
- **msTransitionDuration**
- **msTransitionProperty**
- **msTransitionTimingFunction**
- **msUserSelect**
- **msWrapFlow**
- **msWrapMargin**
- **msWrapThrough**

V0028:

The specification states:

```
azimuth of type DOMString
```

See the `azimuth` property definition in CSS2.

Exceptions on setting

`DOMException`

`SYNTAX_ERR`: Raised if the new value has a syntax error and is unparsable.

`NO_MODIFICATION_ALLOWED_ERR`: Raised if this property is readonly.

### *All Document Modes (All Versions)*

The **`azimuth`** property of the **`CSS2Properties`** interface is not supported.

V0029:

The specification states:

`background` of type `DOMString`

See the `background` property definition in CSS2.

Exceptions on setting

`DOMException`

`SYNTAX_ERR`: Raised if the new value has a syntax error and is unparsable.

`NO_MODIFICATION_ALLOWED_ERR`: Raised if this property is readonly.

### *Quirks Mode (All Versions)*

Setting the **`background`** property to `normal` is the same as setting **`background`** to `none transparent scroll repeat 0% 0%`.

V0030:

The specification states:

`border` of type `DOMString`

See the `border` property definition in CSS2.

Exceptions on setting

`DOMException`

`SYNTAX_ERR`: Raised if the new value has a syntax error and is unparsable.

`NO_MODIFICATION_ALLOWED_ERR`: Raised if this property is readonly.

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

When setting the **`border`** property by using longhand (for example, `border-width:thick; border-style:dotted;`), **`border`** returns blank unless there is either a **`style`** or **`width`** property present.

### *IE7 Mode (Internet Explorer 7)*

The **`border`**, **`borderRight`**, **`borderLeft`**, **`borderTop`**, and **`borderBottom`** properties are always displayed as blank unless there is either a **`style`** or a **`width`** property present. This is regardless of how the property was set by using shorthand or by using longhand.

V0031:

The specification states:

```
borderSpacing of type DOMString
See the border-spacing property definition in CSS2.

Exceptions on setting
DOMException
SYNTAX_ERR: Raised if the new value has a syntax error and is unparsable.

NO_MODIFICATION_ALLOWED_ERR: Raised if this property is readonly.
```

*All Document Modes (Internet Explorer 7)*

The **borderSpacing** property is not supported.

*Quirks Mode and IE7 Mode (All Versions)*

Setting the **borderSpacing** property in the DOM has no effect on layout. The property can be read and set from the DOM but it will not be taken into consideration when laying out the page.

V0032:

The specification states:

```
captionSide of type DOMString
See the caption-side property definition in CSS2.

Exceptions on setting
DOMException
SYNTAX_ERR: Raised if the new value has a syntax error and is unparsable.

NO_MODIFICATION_ALLOWED_ERR: Raised if this property is readonly.
```

*All Document Modes (Internet Explorer 7)*

The **captionSide** property is not supported.

*Quirks Mode and IE7 Mode (All Versions)*

Setting the **captionSide** property in the DOM has no effect on layout. The property can be read and set from the DOM but it will not be taken into consideration when laying out the page.

V0033:

The specification states:

```
clip of type DOMString
See the clip property definition in CSS2.

Exceptions on setting
DOMException
SYNTAX_ERR: Raised if the new value has a syntax error and is unparsable.
NO_MODIFICATION_ALLOWED_ERR: Raised if this property is readonly.
```

*IE7 Mode (All Versions)*

The **clip** property is returned as an empty string instead of the correct value.

V0034:

The specification states:

```
cssFloat of type DOMString
See the float property definition in CSS2.

Exceptions on setting
DOMExceptionSYNTAX_ERR: Raised if the new value has a syntax error and is unparsable.
NO_MODIFICATION_ALLOWED_ERR: Raised if this property is readonly.
```

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

The **cssFloat** property is not supported. Similar functionality is provided by the **styleFloat** property.

V0035:

The specification states:

```
cue of type DOMString
See the cue property definition in CSS2.

Exceptions on setting
DOMExceptionSYNTAX_ERR: Raised if the new value has a syntax error and is unparsable.
NO_MODIFICATION_ALLOWED_ERR: Raised if this property is readonly.
```

*All Document Modes (All Versions)*

The **cue** property is not supported.

V0036:

The specification states:

```
cueAfter of type DOMString
See the cue-after property definition in CSS2.

Exceptions on setting
DOMExceptionSYNTAX_ERR: Raised if the new value has a syntax error and is unparsable.
NO_MODIFICATION_ALLOWED_ERR: Raised if this property is readonly.
```

*All Document Modes (All Versions)*

The **cueAfter** property is not supported.

V0037:

The specification states:

```
cueBefore of type DOMString
See the cue-before property definition in CSS2.
```

Exceptions on setting  
DOMException  
SYNTAX\_ERR: Raised if the new value has a syntax error and is unparsable.  
  
NO\_MODIFICATION\_ALLOWED\_ERR: Raised if this property is readonly.

#### *All Document Modes (All Versions)*

The **cueBefore** property is not supported.

V0038:

The specification states:

elevation of type DOMString  
See the elevation property definition in CSS2.  
  
Exceptions on setting  
DOMException  
SYNTAX\_ERR: Raised if the new value has a syntax error and is unparsable.  
  
NO\_MODIFICATION\_ALLOWED\_ERR: Raised if this property is readonly.

#### *All Document Modes (All Versions)*

The **elevation** property is not supported.

V0039:

The specification states:

fontSizeAdjust of type DOMString  
See the font-size-adjust property definition in CSS2.  
  
Exceptions on setting  
DOMException  
SYNTAX\_ERR: Raised if the new value has a syntax error and is unparsable.  
  
NO\_MODIFICATION\_ALLOWED\_ERR: Raised if this property is readonly.

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

The **fontSizeAdjust** property is not supported.

V0040:

The specification states:

fontStretch of type DOMString  
See the font-stretch property definition in CSS2.  
  
Exceptions on setting  
DOMException  
SYNTAX\_ERR: Raised if the new value has a syntax error and is unparsable.

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

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

The **fontStretch** property is not supported.

V0041:

The specification states:

```
markerOffset of type DOMString
See the marker-offset property definition in CSS2.

Exceptions on setting
DOMException
SYNTAX_ERR: Raised if the new value has a syntax error and is unparsable.

NO_MODIFICATION_ALLOWED_ERR: Raised if this property is readonly.
```

### *All Document Modes (All Versions)*

The **markerOffset** property is not supported.

V0042:

The specification states:

```
marks of type DOMString
See the marks property definition in CSS2.

Exceptions on setting
DOMException
SYNTAX_ERR: Raised if the new value has a syntax error and is unparsable.

NO_MODIFICATION_ALLOWED_ERR: Raised if this property is readonly.
```

### *All Document Modes (All Versions)*

The **marks** property is not supported.

V0043:

The specification states:

```
orphans of type DOMString
See the orphans property definition in CSS2.

Exceptions on setting
DOMException
SYNTAX_ERR: Raised if the new value has a syntax error and is unparsable.

NO_MODIFICATION_ALLOWED_ERR: Raised if this property is readonly.
```

### *All Document Modes (Internet Explorer 7)*

The **orphans** property is not supported.

*Quirks Mode and IE7 Mode (Internet Explorer 8 and Internet Explorer 9)*

Setting the **orphans** property in the DOM has no effect on the page layout. The property can be read and set from the DOM but it will not be taken into consideration when laying out the page.

V0044:

The specification states:

```
outline of type DOMString
See the outline property definition in CSS2.

Exceptions on setting
DOMException
SYNTAX_ERR: Raised if the new value has a syntax error and is unparsable.

NO_MODIFICATION_ALLOWED_ERR: Raised if this property is readonly.
```

*All Document Modes (Internet Explorer 7)*

The **outline** property is not supported.

*Quirks Mode and IE7 Mode (Internet Explorer 8 and Internet Explorer 9)*

The **outline** property is accessible from the DOM. However, changing its value has no effect on the page layout. The value of this DOM property is ignored.

V0045:

The specification states:

```
outlineColor of type DOMString
See the outline-color property definition in CSS2.

Exceptions on setting
DOMException
SYNTAX_ERR: Raised if the new value has a syntax error and is unparsable.

NO_MODIFICATION_ALLOWED_ERR: Raised if this property is readonly.
```

*All Document Modes (Internet Explorer 7)*

The **outlineColor** property is not supported.

*Quirks Mode and IE7 Mode (Internet Explorer 8 and Internet Explorer 9)*

Setting the **outlineColor** property in the DOM has no effect on the page layout. The property can be read and set from the DOM but it will not be taken into consideration when laying out the page.

V0046:

The specification states:

```
outlineStyle of type DOMString
```

See the `outline-style` property definition in CSS2.

Exceptions on setting

`DOMException`

`SYNTAX_ERR`: Raised if the new value has a syntax error and is unparsable.

`NO_MODIFICATION_ALLOWED_ERR`: Raised if this property is readonly.

#### *All Document Modes (Internet Explorer 7)*

The **`outlineStyle`** property is not supported.

#### *Quirks Mode and IE7 Mode (Internet Explorer 8 and Internet Explorer 9)*

Setting the **`outlineStyle`** property in the DOM has no effect on layout. The property can be read and set from the DOM but it will not be taken into consideration when laying out the page.

V0047:

The specification states:

`outlineWidth` of type `DOMString`

See the `outline-width` property definition in CSS2.

Exceptions on setting

`DOMException`

`SYNTAX_ERR`: Raised if the new value has a syntax error and is unparsable.

`NO_MODIFICATION_ALLOWED_ERR`: Raised if this property is readonly.

#### *All Document Modes (Internet Explorer 7)*

The **`outlineWidth`** property is not supported.

#### *Quirks Mode and IE7 Mode (Internet Explorer 8 and Internet Explorer 9)*

Setting the **`outlineWidth`** property in the DOM does not affect the layout. The property can be read and set from the DOM but it is not used during page layout.

V0048:

The specification states:

page of type `DOMString`

See the `page` property definition in CSS2.

Exceptions on setting

`DOMException`

`SYNTAX_ERR`: Raised if the new value has a syntax error and is unparsable.

`NO_MODIFICATION_ALLOWED_ERR`: Raised if this property is readonly.

#### *All Document Modes (All Versions)*

The **`page`** property is not supported.



V0049:

The specification states:

pause of type DOMString  
See the pause property definition in CSS2.

Exceptions on setting  
DOMException

SYNTAX\_ERR: Raised if the new value has a syntax error and is unparsable.

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

*All Document Modes (All Versions)*

The **pause** property is not supported.

V0050:

The specification states:

pauseAfter of type DOMString  
See the pause-after property definition in CSS2.

Exceptions on setting  
DOMException

SYNTAX\_ERR: Raised if the new value has a syntax error and is unparsable.

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

*All Document Modes (All Versions)*

The **pauseAfter** property is not supported.

V0051:

The specification states:

pauseBefore of type DOMString  
See the pause-before property definition in CSS2.

Exceptions on setting  
DOMException

SYNTAX\_ERR: Raised if the new value has a syntax error and is unparsable.

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

*All Document Modes (All Versions)*

The **pauseBefore** property is not supported.

V0052:

The specification states:

pitch of type DOMString  
See the pitch property definition in CSS2.

Exceptions on setting  
DOMException

SYNTAX\_ERR: Raised if the new value has a syntax error and is unparsable.

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

#### *All Document Modes (All Versions)*

The **pitch** property is not supported.

V0053:

The specification states:

pitchRange of type DOMString  
See the pitch-range property definition in CSS2.

Exceptions on setting  
DOMException

SYNTAX\_ERR: Raised if the new value has a syntax error and is unparsable.

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

#### *All Document Modes (All Versions)*

The **pitchRange** property is not supported.

V0054:

The specification states:

"playDuring of type DOMStringSee the play-during property definition in CSS2.Exceptions on setting DOMExceptionSYNTAX\_ERR: Raised if the new value has a syntax error and is unparsable.NO\_MODIFICATION\_ALLOWED\_ERR: Raised if this property is readonly."

#### *All Document Modes (All Versions)*

The **playDuring** property is not supported.

V0055:

The specification states:

richness of type DOMString  
See the richness property definition in CSS2.

Exceptions on setting  
DOMException SYNTAX\_ERR: Raised if the new value has a syntax error and is unparsable.

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

#### *All Document Modes (All Versions)*

The **richness** property is not supported.

V0056:

The specification states:

size of type DOMString  
See the size property definition in CSS2.

Exceptions on setting  
DOMExceptionSYNTAX\_ERR: Raised if the new value has a syntax error and is unparsable.  
NO\_MODIFICATION\_ALLOWED\_ERR: Raised if this property is readonly.

#### *All Document Modes (All Versions)*

The **size** property is not supported.

V0057:

The specification states:

speak of type DOMString  
See the speak property definition in CSS2.

Exceptions on setting  
DOMExceptionSYNTAX\_ERR: Raised if the new value has a syntax error and is unparsable.  
NO\_MODIFICATION\_ALLOWED\_ERR: Raised if this property is readonly.

#### *All Document Modes (All Versions)*

The **speak** property is not supported.

V0058:

The specification states:

speakHeader of type DOMString  
See the speak-header property definition in CSS2.

Exceptions on setting  
DOMExceptionSYNTAX\_ERR: Raised if the new value has a syntax error and is unparsable.  
NO\_MODIFICATION\_ALLOWED\_ERR: Raised if this property is readonly.

#### *All Document Modes (All Versions)*

The **speakHeader** property is not supported.

V0059:

The specification states:

speakingNumeral of type DOMString  
See the speaking-numeral property definition in CSS2.

Exceptions on setting  
DOMExceptionSYNTAX\_ERR: Raised if the new value has a syntax error and is unparsable.  
NO\_MODIFICATION\_ALLOWED\_ERR: Raised if this property is readonly.

#### *All Document Modes (All Versions)*

The **speakingNumeral** property is not supported.

V0060:

The specification states:

speakingPunctuation of type DOMString  
See the speaking-punctuation property definition in CSS2.

Exceptions on setting  
DOMExceptionSYNTAX\_ERR: Raised if the new value has a syntax error and is unparsable.  
NO\_MODIFICATION\_ALLOWED\_ERR: Raised if this property is readonly.

#### *All Document Modes (All Versions)*

The **speakingPunctuation** property is not supported.

V0061:

The specification states:

speechRate of type DOMString  
See the speech-rate property definition in CSS2.

Exceptions on setting  
DOMExceptionSYNTAX\_ERR: Raised if the new value has a syntax error and is unparsable.  
NO\_MODIFICATION\_ALLOWED\_ERR: Raised if this property is readonly.

#### *All Document Modes (All Versions)*

The **speechRate** property is not supported.

V0062:

The specification states:

stress of type DOMString  
See the stress property definition in CSS2.

Exceptions on setting  
DOMExceptionSYNTAX\_ERR: Raised if the new value has a syntax error and is unparsable.  
NO\_MODIFICATION\_ALLOWED\_ERR: Raised if this property is readonly.

#### *All Document Modes (All Versions)*

The **stress** property is not supported.

V0063:

The specification states:

textShadow of type DOMString  
See the text-shadow property definition in CSS2.

Exceptions on setting  
DOMExceptionSYNTAX\_ERR: Raised if the new value has a syntax error and is unparsable.  
NO\_MODIFICATION\_ALLOWED\_ERR: Raised if this property is readonly.

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

The **textShadow** property is not supported.

V0064:

The specification states:

voiceFamily of type DOMString  
See the voice-family property definition in CSS2.

Exceptions on setting  
DOMExceptionSYNTAX\_ERR: Raised if the new value has a syntax error and is unparsable.  
NO\_MODIFICATION\_ALLOWED\_ERR: Raised if this property is readonly.

*All Document Modes (All Versions)*

The **voiceFamily** property is not supported.

V0065:

The specification states:

volume of type DOMString  
See the volume property definition in CSS2.

Exceptions on setting  
DOMExceptionSYNTAX\_ERR: Raised if the new value has a syntax error and is unparsable.  
NO\_MODIFICATION\_ALLOWED\_ERR: Raised if this property is readonly.

*All Document Modes (All Versions)*

The **volume** property is not supported.

V0066:

The specification states:

whiteSpace of type DOMString  
See the white-space property definition in CSS2.

Exceptions on setting

DOMExceptionSYNTAX\_ERR: Raised if the new value has a syntax error and is unparsable.  
NO\_MODIFICATION\_ALLOWED\_ERR: Raised if this property is readonly.

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

Setting the **whiteSpace** property in the DOM has no effect on layout. The property can be read and set from the DOM but it will not be taken into consideration when laying out the page.

### **2.3 Error Handling**

There are no additional considerations for error handling.

### **2.4 Security**

There are no additional security considerations.

### 3 Change Tracking

This section identifies changes that were made to the [MS-DOM2S] protocol document between the February 2012 and July 2012 releases. Changes are classified as New, Major, Minor, Editorial, or No change.

The revision class **New** means that a new document is being released.

The revision class **Major** means that the technical content in the document was significantly revised. Major changes affect protocol interoperability or implementation. Examples of major changes are:

- A document revision that incorporates changes to interoperability requirements or functionality.
- An extensive rewrite, addition, or deletion of major portions of content.
- The removal of a document from the documentation set.
- Changes made for template compliance.

The revision class **Minor** means that the meaning of the technical content was clarified. Minor changes do not affect protocol interoperability or implementation. Examples of minor changes are updates to clarify ambiguity at the sentence, paragraph, or table level.

The revision class **Editorial** means that the language and formatting in the technical content was changed. Editorial changes apply to grammatical, formatting, and style issues.

The revision class **No change** means that no new technical or language changes were introduced. The technical content of the document is identical to the last released version, but minor editorial and formatting changes, as well as updates to the header and footer information, and to the revision summary, may have been made.

Major and minor changes can be described further using the following change types:

- New content added.
- Content updated.
- Content removed.
- New product behavior note added.
- Product behavior note updated.
- Product behavior note removed.
- New protocol syntax added.
- Protocol syntax updated.
- Protocol syntax removed.
- New content added due to protocol revision.
- Content updated due to protocol revision.
- Content removed due to protocol revision.
- New protocol syntax added due to protocol revision.

- Protocol syntax updated due to protocol revision.
- Protocol syntax removed due to protocol revision.
- New content added for template compliance.
- Content updated for template compliance.
- Content removed for template compliance.
- Obsolete document removed.

Editorial changes are always classified with the change type **Editorially updated**.

Some important terms used in the change type descriptions are defined as follows:

- **Protocol syntax** refers to data elements (such as packets, structures, enumerations, and methods) as well as interfaces.
- **Protocol revision** refers to changes made to a protocol that affect the bits that are sent over the wire.

The changes made to this document are listed in the following table. For more information, please contact [protocol@microsoft.com](mailto:protocol@microsoft.com).

Section	Tracking number (if applicable) and description	Major change (Y or N)	Change type
<a href="#">1 Introduction</a>	Updated document to remove beta tagging.	N	Content updated.



## 4 Index

### C

[Change tracking](#) 39

### G

[Glossary](#) 4

### H

[HTML and Style Sheet Creation](#) 9

### I

[Informative references](#) 4

[Interface CSS2Properties \(introduced in DOM Level 2\)](#) 17

[Interface CSSStyleSheet \(introduced in DOM Level 2\)](#) 9

[Interface DOMImplementationCSS \(introduced in DOM Level 2\)](#) 16

[Interface LinkStyle \(introduced in DOM Level 2\)](#) 8

[Interface StyleSheet \(introduced in DOM Level 2\)](#) 7

[Interface ViewCSS \(introduced in DOM Level 2\)](#) 16

[Introduction](#) 4

### N

[Normative references](#) 4

### R

References

[informative](#) 4

[normative](#) 4

### T

[Tracking changes](#) 39