[MS-DOM2S]:

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

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	5.1	None	No changes to the meaning, language, or formatting of the technical content.
3/22/2016	5.1	None	No changes to the meaning, language, or formatting of the technical content.
7/19/2016	5.2	Minor	Clarified the meaning of the technical content.
11/2/2016	5.2	None	No changes to the meaning, language, or formatting of the technical content.
3/14/2017	5.2	None	No changes to the meaning, language, or formatting of the technical content.
10/3/2017	5.2	None	No changes to the meaning, language, or formatting of the technical content.
2/22/2018	5.2	None	No changes to the meaning, language, or formatting of the technical content.
3/23/2018	5.2	None	No changes to the meaning, language, or formatting of the technical content.
8/28/2018	5.2	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 Style Specification Version 1.0* [DOM Level 2 - Style], published 13 November 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 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 - Style] World Wide Web Consortium, "Document Object Model (DOM) Level 2 Style Specification Version 1.0", W3C Recommendation 13 November 2000, http://www.w3.org/TR/2000/REC-DOM-Level-2-Style-20001113/

[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 web browser versions implement some portion of [DOM Level 2 - Style]:

- 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 - 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 [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 - Style].

- 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 - Style].

2.1.1   [DOM Level 2 - Style] Section 1.2, Style Sheet Interfaces

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**.

2.1.2   [DOM Level 2 - Style] Section 1.3, Document Extensions

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.

2.1.3   [DOM Level 2 - Style] Section 1.4, Association between a style sheet and a document.

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.

2.1.4 [DOM Level 2 - Style] Section 2.2, CSS Fundamental Interfaces

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 non-standard 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, IE10 Mode, and IE11 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, IE10 Mode, IE11 Mode, and EdgeHTML 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 stylesheets::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, IE10 Mode, IE11 Mode, and EdgeHTML 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.

2.1.5   [DOM Level 2 - Style] Section 2.2.1, Override and computed style sheet

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, IE10 Mode, and IE11 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.

2.1.6   [DOM Level 2 - Style] Section 2.2.2, Style sheet creation

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
```

All Document Modes (All Versions)

The **DOMImplementationCSS** interface is not supported.

2.1.7   [DOM Level 2 - Style] Section 2.3, CSS2 Extended Interface

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.
```

Quirks Mode, IE7 Mode, IE8 Mode, IE9 Mode, IE10 Mode, and IE11 Mode (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:

```
speakNumeral of type DOMString
See the speak-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 **speakNumeral** property is not supported.

V0060:

The specification states:

```
speakPunctuation of type DOMString
See the speak-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 **speakPunctuation** 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.

2.2   Clarifications

There are no clarifications of the MAY and SHOULD requirements of [DOM Level 2 - Style].

2.3   Extensions

The following subsection describes extensions to the requirements of [DOM Level 2 - Style].

2.3.1 [DOM Level 2 - Style] Section 2.3, CSS2 Extended Interface

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, IE10 Mode, and IE11 Mode (All Versions)

The following non-standard properties are supported:

- **accelerator**

- **layoutFlow**

- **layoutGrid**

- **layoutGridChar**

- **layoutGridLine**

- **layoutGridMode**

- **layoutGridType**

- **lineBreak**

- **msInterpolationMode**

- **pixelBottom**

- **pixelHeight**

- **pixelLeft**

- **pixelRight**

- **pixelTop**

- **pixelWidth**

- **posBottom**

- **posHeight**

- **posLeft**

- **posRight**

- **posTop**

- **posWidth**

- **scrollbar3dLightColor**

- **scrollbarArrowColor**

- **scrollbarBaseColor**

- **scrollbarDarkShadowColor**

- **scrollbarFaceColor**

- **scrollbarHighlightColor**

- **scrollbarShadowColor**

- **scrollbarTrackColor**

- **styleFloat**

- **textAutospace**

- **textDecorationBlink**

- **textDecorationLineThrough**

- **textDecorationNone**

- **textDecorationOverline**

- **textDecorationUnderline**

- **textJustifyTrim**

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

The following non-standard properties are supported:

- **backgroundPositionX**

- **backgroundPositionY**

- **imeMode**

- **overflow**

- **overflowX**

- **overflowY**

- **rubyAlign**

- **rubyOverhang**

- **rubyPosition**

- **textAlignLast**

- **textJustify**

- **textKashida**

- **textKashidaSpace**

- **textOverflow**

- **textUnderlinePosition**

- **wordbreak**

- **writingMode**

zoommode

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

The following non-standard properties are supported:

- **behavior**

IE8 Mode (All Versions)

The following non-standard properties are supported:

- **msBlockProgression**

- **msBoxSizing**

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

The following non-standard properties are supported:

- **accelerator**

- **backgroundPositionX**

- **backgroundPositionY**

- **imeMode**

- **layoutFlow**

- **layoutGrid**

- **layoutGridChar**

- **layoutGridLine**

- **layoutGridMode**

- **layoutGridType**

- **lineBreak**

- **msBlockProgression**

- **msBoxAlign**

- **msBoxDirection**

- **msBoxFlex**

- **msBoxFlexGroup**

- **msBoxLineProgression**

- **msBoxLines**

- **msBoxOrdinalGroup**

- **msBoxOrient**

- **msBoxPack**

- **msGetPropertyEnabled**

- **msInterpolationMode**

- **msPutPropertyEnabled**

- **msTransform**

- **msTransformOrigin**

- **overflowX**

- **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**

- **zoom**

IE10 Mode and IE 11 Mode (All Versions)

The following non-standard properties are supported:

- **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**

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

A

Association between a style sheet and a document. 8

C

Change tracking 35
CSS Fundamental Interfaces 9
CSS2 Extended Interface (section 2.1.7 16, section 2.3.1 28)

D

Document Extensions 8

G

Glossary 4

I

Informative references 4
Introduction 4

N

Normative references 4

O

Override and computed style sheet 15

R

References
    informative 4
    normative 4

S

Style sheet creation 16
Style Sheet Interfaces 7

T

Tracking changes 35