

[MS-SVG]: Internet Explorer Scalable Vector Graphics (SVG) 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's Open Specification Promise (available here: <http://www.microsoft.com/interop/osp>) or the Community Promise (available here: <http://www.microsoft.com/interop/cp/default.mspx>). 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 iplq@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
09/08/2010	0.1	New	Released new document.
10/13/2010	0.2	Minor	Clarified the meaning of the technical content.
02/10/2011	1.0	Minor	Clarified the meaning of the technical content.
02/28/2011	1.1	Major	Significantly changed the technical content.
03/23/2011	1.2	Minor	Clarified the meaning of the technical content.

Table of Contents

1	Introduction	5
1.1	Glossary	5
1.2	References.....	5
1.2.1	Normative References.....	5
1.2.2	Informative References	5
1.3	Microsoft Implementations.....	5
1.4	Standards Support Requirements	6
1.5	Notation	6
2	Standards Support Statements.....	7
2.1	Normative Variations.....	7
2.1.1	[SVG11] Section 4.3, Basic DOM interfaces	7
2.1.2	[SVG11] Section 5.10.2, The xml:lang and xml:space attributes	8
2.1.3	[SVG11] Section 6.11, The 'style' element.....	8
2.1.4	[SVG11] Section 7.7, The viewBox attribute	8
2.1.5	[SVG11] Section 7.11, Object bounding box units	8
2.1.6	[SVG11] Section 8.5, DOM interfaces.....	9
2.1.7	[SVG11] Section 10.7.3, Glyph orientation within a text run	9
2.1.8	[SVG11] Section 10.9.2, Baseline alignment properties.....	10
2.1.9	[SVG11] Section 10.11, Spacing properties	11
2.1.10	[SVG11] Section 10.14, Alternate glyphs.....	11
2.1.11	[SVG11] Section 10.15, White space handling.....	11
2.1.12	[SVG11] Section 10.17, Text Module	12
2.1.13	[SVG11] Section 11.6.2, The 'marker' element.....	12
2.1.14	[SVG11] Section 11.7.1, Color interpolation properties: 'color-interpolation' and 'color-interpolation-filters'	13
2.1.15	[SVG11] Section 11.7.2, The 'color-rendering' property.....	13
2.1.16	[SVG11] Section 11.7.4, The 'text-rendering' property.....	13
2.1.17	[SVG11] Section 11.7.5, The 'image-rendering' property.....	13
2.1.18	[SVG11] Section 11.15, DOM interfaces	13
2.1.19	[SVG11] Section 13.3, Patterns.....	14
2.1.20	[SVG11] Section 14.3.3, The 'overflow' and 'clip' properties.....	14
2.1.21	[SVG11] Section 14.3.5, Establishing a new clipping path.....	15
2.1.22	[SVG11] Section 15.2, An example.....	15
2.1.23	[SVG11] Section 16.7, Magnification and panning.....	15
2.1.24	[SVG11] Section 16.12, Cursor Module	16
2.1.25	[SVG11] Section 17.2.2, SVG fragment identifiers	16
2.1.26	[SVG11] Section 17.4, XLink Attribute Module	16
2.1.27	[SVG11] Section 17.5, ExternalResourcesRequired Attribute Module.....	17
2.1.28	[SVG11] Section 19, Animation	17
2.1.29	[SVG11] Section 23.6, Extensibility Module	17
2.1.30	[SVG11] Section B.5, Relationship with DOM2 events	18
2.1.31	[SVG11] Section C., IDL Definitions	18
2.2	Clarifications	18
2.2.1	[SVG11] Section 4.3, Basic DOM interfaces	18
2.2.2	[SVG11] Section 5.11, DOM interfaces.....	19
2.2.3	[SVG11] Section 6.18, Aural style sheets	20
2.2.4	[SVG11] Section 7.12, Geographic Coordinate Systems	20
2.2.5	[SVG11] Section 8.3.9, The grammar for path data	20
2.2.6	[SVG11] Section 10.5, The 'tspan' element	21

2.2.7	[SVG11] Section 10.9.1, Text alignment properties	22
2.2.8	[SVG11] Section 10.19, DOM interfaces	22
2.2.9	[SVG11] Section 11, Painting: Filling, Stroking and Marker Symbols	23
2.2.10	[SVG11] Section 11.4, Stroke Properties	23
2.2.11	[SVG11] Section 11.7.3, The 'shape-rendering' property	23
2.2.12	[SVG11] Section 13.3, Patterns	23
2.2.13	[SVG11] Section 14.3.3, The 'overflow' and 'clip' properties	24
2.2.14	[SVG11] Section 16.2, Complete list of supported events	24
2.2.15	[SVG11] Section B.1, SVG DOM Overview	25
2.3	Error Handling	25
2.4	Security	25
3	Change Tracking.....	26
4	Index	27

1 Introduction

This document describes the level of support provided by Windows® Internet Explorer® 9 for the *Scalable Graphics (VG) 1.1 Specification* [\[W3C-SVG1.1\]](#), W3C Recommendation published January 14, 2003 and edited in place April 30, 2009. Internet Explorer 9 is the only version of Windows® Internet Explorer® that supports SVG.

The [\[W3C-SVG1.1\]](#) 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

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. Please check the archive site, <http://msdn2.microsoft.com/en-us/library/E4BD6494-06AD-4aed-9823-445E921C9624>, as an additional source.

[DOM Level 2 - Core] W3C, "Document Object Model (DOM) Level 2 Core Specification Version 1.0", W3C Recommendation 13 November, 2000, <http://www.w3.org/TR/DOM-Level-2-Core/>

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

[W3C-SVG1.1] W3C, "Scalable Vector Graphics (SVG) 1.1 Specification", W3C Recommendation 14 January 2003, edited in place 30 April 2009, <http://www.w3.org/TR/2003/REC-SVG11-20030114/>

1.2.2 Informative References

[W3C-SVG1.1/2] Andersson, O., et al., "Scalable Vector Graphics (SVG) 1.1 (Second Edition)", W3C Working Draft 22 June 2010, <http://www.w3.org/TR/SVG11/>

1.3 Microsoft Implementations

Windows® Internet Explorer® 9 implements the following document modes. Note that IE9 mode is the only document mode that supports SVG.

- Quirks Mode
- IE7 Mode
- IE8 Mode
- IE9 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 "IE5 mode" and "Quirks mode" refer to the same document mode in Internet Explorer 9.

1.4 Standards Support Requirements

To conform to [\[W3C-SVG1.1\]](#), 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 [\[W3C-SVG1.1\]](#) and whether they are considered normative or informative.

Sections	Normative/Informative
1-3	Informative
4-23	Normative
Appendices A-C	Normative
Appendices D, E	Informative
Appendices F, G, O	Normative
Appendices H-N, P	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 a full list of variations, clarifications, and extension points in the Microsoft implementation of **[W3C-SVG1.1]**.

- 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

The following subsections detail the normative variations from MUST requirements in [\[W3C-SVG1.1\]](#).

2.1.1 [SVG11] Section 4.3, Basic DOM interfaces

V0001:

The specification defines the **insertItemBefore**, **replaceItem**, and **appendItem** methods of the **SVGStringList** interface as follows:

insertItemBefore

Inserts a new item into the list at the specified position. The first item is number 0. If **newItem** is already in a list, it is removed from its previous list before it is inserted into this list.

Parameters

in DOMString **newItem** The item which is to be inserted into the list.

replaceItem

Replaces an existing item in the list with a new item. If **newItem** is already in a list, it is removed from its previous list before it is inserted into this list.

Parameters

in DOMString **newItem** The item which is to be inserted into the list.

appendItem

Inserts a new item at the end of the list. If **newItem** is already in a list, it is removed from its previous list before it is inserted into this list.

Parameters

in SVGNumber **newItem** The item which is to be inserted into the list. The first item is number 0.

IE9 Mode (All Versions)

When the **insertItemBefore**, **replaceItem**, or **appendItem** methods are called and the **newItem** parameter is already in a list, it remains in its previous list and a copy is inserted into this list.

2.1.2 [SVG11] Section 5.10.2, The `xml:lang` and `xml:space` attributes

V0030:

IE9 Mode (All Versions)

The **xml:space** attribute is not supported in SVG.

2.1.3 [SVG11] Section 6.11, The 'style' element

V0002:

The specification states:

```
type = content-type
This attribute specifies the style sheet language of the element's contents. The style sheet
language is specified as a content type (e.g., "text/css"), as per [RFC2045]. Authors must
supply a value for this attribute; there is no default value.
```

```
Animatable: no.
```

IE9 Mode (All Versions)

The **type** attribute has a default value of `text/css`.

For more information, see section 6.10 of [\[W3C-SVG1.1/2\]](#).

2.1.4 [SVG11] Section 7.7, The `viewBox` attribute

V0032:

The specification states:

```
All elements that establish a new viewport (see elements that establish viewports), plus the
'marker', 'pattern' and 'view' elements have attribute viewBox. The value of the viewBox
attribute is a list of four numbers <min-x>, <min-y>, <width> and <height>, separated by
whitespace and/or a comma, which specify a rectangle in user space which should be mapped to
the bounds of the viewport established by the given element, taking into account attribute
preserveAspectRatio. If specified, an additional transformation is applied to all descendants
of the given element to achieve the specified effect.
```

IE9 Mode (All Versions)

A value of 0 on the **viewBox** height or width does not disable rendering of the element.

2.1.5 [SVG11] Section 7.11, Object bounding box units

V0003:

The specification states:

```
Element: 'filter' Attribute

Attribute: filterUnits="objectBoundingBox"
```


Effect: Indicates that the attributes which define the filter effects region (x, y, width, height) represent fractions or percentages of the bounding box of the element to which the filter is applied.

IE9 Mode (All Versions)

The **filter** element and the **filterUnits** attribute are not supported.

V0006:

The specification states:

Element: 'filter' Attribute

Attribute: primitiveUnits ="objectBoundingBox"

Effect: Indicates that the various length values within the filter primitives represent fractions or percentages of the bounding box of the element to which the filter is applied.

IE9 Mode (All Versions)

The **filter** element and the **primitiveUnits** attribute are not supported.

2.1.6 [SVG11] Section 8.5, DOM interfaces

V0035:

The specification states:

DOM attribute `normalizedPathSegList` provides normalized access to the static/base contents of the `d` attribute where all path data commands are expressed in terms of the following subset of `SVGPathSeg` types: `SVG_PATHSEG_MOVETO_ABS` (M), `SVG_PATHSEG_LINETO_ABS` (L), `SVG_PATHSEG_CURVETO_CUBIC_ABS` (C) and `SVG_PATHSEG_CLOSEPATH` (z).and two lists to access the current animated values of the `d` attribute:

IE9 Mode (All Versions)

The **normalizedPathSegList** attribute is not supported.

C0022:

The specification defines the **pathLength** attribute of the **SVGPathElement** interface as follows:

readonly attribute `SVGAnimatedNumber` `pathLength`;

IE9 Mode (All Versions)

The **pathLength** attribute is not supported.

2.1.7 [SVG11] Section 10.7.3, Glyph orientation within a text run

V0036:

The specification states:

In some cases, it is required to alter the orientation of a sequence of characters relative to the inline-progression-direction. The requirement is particularly applicable to vertical layouts of East Asian documents, where sometimes narrow-cell Latin text is to be displayed horizontally and other times vertically. Two properties control the glyph orientation relative to the reference orientation for each of the two possible inline-progression-directions. 'glyph-orientation-vertical' controls glyph orientation when the inline-progression-direction is vertical. 'glyph-orientation-horizontal' controls glyph orientation when the inline-progression-direction is horizontal.

IE9 Mode (All Versions)

The **glyph-orientation-horizontal** and **glyph-orientation-vertical** attributes are not supported.

2.1.1.8 [SVG11] Section 10.9.2, Baseline alignment properties

V0037:

The specification states:

```
'dominant-baseline'
Value:  auto | use-script | no-change | reset-size | ideographic | alphabetic |
        hanging | | mathematical | central | middle | text-after-edge | text-before-edge |
        inherit
Initial:  auto
Applies to:  text content elements
Inherited:  no
Percentages:  N/A
Media:  visual
Animatable:  yes
```

IE9 Mode (All Versions)

The **dominant-baseline** attribute is not supported.

C0023:

The specification states:

```
alignment-baseline'
Value:  auto | baseline | before-edge | text-before-edge | middle | central | after-edge |
        text-after-edge | ideographic | alphabetic | hanging | mathematical | inherit
Initial:  auto
Applies to:  'tspan', 'tref', 'altGlyph', 'textPath' elements
Inherited:  no
Percentages:  N/A
Media:  visual
Animatable:  yes
```

IE9 Mode (All Versions)

The **alignment-baseline** property is not supported.

V0038:

The specification states:

`'baseline-shift'`
Value: baseline | sub | super | <percentage> | <length> | inherit
Initial: baseline
Applies to: 'tspan', 'tref', 'altGlyph', 'textPath' elements
Inherited: no
Percentages: refers to the "line-height" of the 'text' element, which in the case of SVG is defined to be equal to the 'font-size'
Media: visual
Animatable:

IE9 Mode (All Versions)

The **baseline-shift** attribute is not supported.

2.1.9 [SVG11] Section 10.11, Spacing properties

V0039:

The specification states:

`'kerning'`
Value: auto | <length> | inherit
Initial: auto
Applies to: text content elements
Inherited: yes
Percentages: N/A
Media: visual
Animatable: yes

IE9 Mode (All Versions)

The **kerning** attribute is not supported.

2.1.10 [SVG11] Section 10.14, Alternate glyphs

V0040:

The specification states:

There are situations such as ligatures, special-purpose fonts (e.g., a font for music symbols) or alternate glyphs for Asian text strings where it is required that a different set of glyphs is used than the glyph(s) which normally corresponds to the given character data. The 'altGlyph' element provides control over the glyphs used to render particular character data.

IE9 Mode (All Versions)

The **altGlyph** element is not supported.

2.1.11 [SVG11] Section 10.15, White space handling

V0007:

The specification states (see also [SVG11] Section 5.10., The xml:lang and xml:space attributes):

SVG supports the standard XML attribute `xml:space` to specify the handling of white space characters within a given 'text' element's character data. The SVG user agent has special processing rules associated with this attribute as described below. These are behaviors that occur subsequent to XML parsing [XML10] and any construction of a Document Object Model [DOM2].

IE9 Mode (All Versions)

The **xml:space** attribute is not supported.

2.1.12 [SVG11] Section 10.17, Text Module

V0008:

The specification states:

Element `tref`:

Attributes:

`Core.attrib`, `Conditional.attrib`, `External.attrib`, `Style.attrib`, `XLinkRequired.attrib`, `x`, `y`, `dx`, `dy`, `rotate`, `textLength`, `lengthAdjust`, `GraphicalEvents.attrib`, `Paint.attrib`, `Paint.attrib`, `Font.attrib`, `Opacity.attrib`, `Graphics.attrib`, `Cursor.attrib`, `Filter.attrib`, `Mask.attrib`, `GraphicalEvents.attrib`, `Clip.attrib`, `TextContent.attrib`

Content Model:

`(#PCDATA | Description.class | Animation.class)*`

IE9 Mode (All Versions)

The **tref** element and its attributes are not supported.

2.1.13 [SVG11] Section 11.6.2, The 'marker' element

V0010:

The specification states:

Properties inherit into the 'marker' element from its ancestors; properties do not inherit from the element referencing the 'marker' element.

'marker' elements are never rendered directly; their only usage is as something that can be referenced using the 'marker', 'marker-start', 'marker-end' and 'marker-mid' properties. The 'display' property does not apply to the 'marker' element; thus, 'marker' elements are not directly rendered even if the 'display' property is set to a value other than none, and 'marker' elements are available for referencing even when the 'display' property on the 'marker' element or any of its ancestors is set to none.

Event attributes and event listeners attached to the contents of a 'marker' element are not processed; only the rendering aspects of 'marker' elements are processed.

IE9 Mode (All Versions)

Properties of a **marker** element inherit at the point of reference, not from the ancestors of the **marker** element.

2.1.14 [SVG11] Section 11.7.1, Color interpolation properties: 'color-interpolation' and 'color-interpolation-filters'

V0011:

The specification defines the **color-interpolation** and **color-interpolation-filters** properties.

IE9 Mode (All Versions)

The **color-interpolation** and **color-interpolation-filters** properties are not supported.

2.1.15 [SVG11] Section 11.7.2, The 'color-rendering' property

V0013:

The specification defines the **color-rendering** property.

IE9 Mode (All Versions)

The **color-rendering** property is not supported.

2.1.16 [SVG11] Section 11.7.4, The 'text-rendering' property

V0014:

The specification defines the **text-rendering** property.

IE9 Mode (All Versions)

The **text-rendering** property is not supported.

2.1.17 [SVG11] Section 11.7.5, The 'image-rendering' property

V0015:

The specification defines the **image-rendering** property.

IE9 Mode (All Versions)

The **image-rendering** property is not supported.

2.1.18 [SVG11] Section 11.15, DOM interfaces

V0017:

The specification states:

Definition group Marker Unit Types

Defined constants

SVG_MARKERUNITS_UNKNOWN The marker unit type is not one of predefined types. It is invalid to attempt to define a new value of this type or to attempt to switch an existing value to this type.

SVG_MARKERUNITS_USERSPACEONUSE The value of attribute markerUnits is 'userSpaceOnUse'.

SVG_MARKERUNITS_STROKEWIDTH The value of attribute markerUnits is 'strokeWidth'.

Attributes

readonly SVGAnimatedLength refX
Corresponds to attribute refX on the given 'marker' element.
readonly SVGAnimatedLength refY
Corresponds to attribute refY on the given 'marker' element.
readonly SVGAnimatedEnumeration markerUnits
Corresponds to attribute markerUnits on the given 'marker' element.
One of the Marker Units Types defined above.

IE9 Mode (All Versions)

The **markerUnits** attribute of the **SVGMarkerElement** interface is not supported.

The following constant values are not supported:

- SVG_MARKERUNITS_UNKNOWN
- SVG_MARKERUNITS_USERSPACEONUSE
- SVG_MARKERUNITS_STROKEWIDTH

2.1.19 [SVG11] Section 13.3, Patterns

V0049:

The specification states:

'pattern' elements are never rendered directly; their only usage is as something that can be referenced using the 'fill' and 'stroke' properties. The 'display' property does not apply to the 'pattern' element; thus, 'pattern' elements are not directly rendered even if the 'display' property is set to a value other than none, and 'pattern' elements are available for referencing even when the 'display' property on the 'pattern' element or any of its ancestors is set to none.

IE9 Mode (All Versions)

The **display** property affects **pattern** elements and references to those **pattern** elements.

2.1.20 [SVG11] Section 14.3.3, The 'overflow' and 'clip' properties

V0018:

The specification states:

'clip'
Value: <shape> | auto | inherit
Initial: auto
Applies to: elements which establish a new viewport, 'pattern' elements and 'marker' elements
Inherited: no
Percentages: N/A
Media: visual
Animatable: yes

The 'clip' property has the same parameter values as defined in [CSS2-clip]. Unitless values, which indicate current user coordinates, are permitted on the coordinate values on the

<shape>. The value of "auto" defines a clipping path along the bounds of the viewport created by the given element.

IE9 Mode (All Versions)

The **clip** property is not supported.

2.1.21 [SVG11] Section 14.3.5, Establishing a new clipping path

V0019:

The specification states:

A 'clipPath' element can contain 'path' elements, 'text' elements, basic shapes (such as 'circle') or a 'use' element. If a 'use' element is a child of a 'clipPath' element, it must directly reference 'path', 'text' or basic shape elements. Indirect references are an error (see Error processing).

IE9 Mode (All Versions)

The **clipPath** element allows indirect references. Indirect references are not an error.

2.1.22 [SVG11] Section 15.2, An example

V0020:

The specification includes Chapter 15, Filter Effects.

IE9 Mode (All Versions)

SVG filter effects, according to Chapter 15, Filter Effects, of [\[W3C-SVG1.1\]](#), are not supported.

2.1.23 [SVG11] Section 16.7, Magnification and panning

V0021:

The specification states:

The outermost 'svg' element in an SVG document fragment has attribute zoomAndPan, which takes the possible values of disable and magnify, with the default being magnify.

If disable, the user agent shall disable any magnification and panning controls and not allow the user to magnify or pan on the given document fragment.

If magnify, in environments that support user interactivity, the user agent shall provide controls to allow the user to perform a "magnify" operation on the document fragment.

If a zoomAndPan attribute is assigned to an inner 'svg' element, the zoomAndPan setting on the inner 'svg' element will have no effect on the SVG user agent.

Animatable: no.

IE9 Mode (All Versions)

The **zoomAndPan** attribute is not supported, except for the default magnify operation.

2.1.24 [SVG11] Section 16.12, Cursor Module

V0022:

The specification states:

Elements: `cursor`

Attributes: `Core.attrib`, `XLinkRequired.attrib`, `Conditional.attrib`, `External.attrib`, `x`, `y`

Content Model: `(Description.class)`

IE9 Mode (All Versions)

The **cursor** element and its attributes are not supported.

2.1.25 [SVG11] Section 17.2.2, SVG fragment identifiers

V0055:

The specification states:

If the SVG fragment identifier addresses specific SVG view (e.g., `MyDrawing.svg#svgView(viewBox(0,200,1000,1000))`), then the document fragment defined by the closest ancestor 'svg' element is displayed in the viewport using the SVG view specification provided by the SVG fragment identifier.

IE9 Mode (All Versions)

Parameters for **viewBox** elements in fragment specifications may be delimited by single spaces in addition to commas.

2.1.26 [SVG11] Section 17.4, XLink Attribute Module

V0023:

The specification states:

The XLink Attribute Module defines the `XLink.attrib`, `XLinkRequired.attrib`, `XLinkEmbed.attrib` and `XLinkReplace.attrib` attribute sets.

Attributes in Collection: `xlink:type`, `xlink:href`, `xlink:role`, `xlink:arcrole`, `xlink:title`, `xlink:show`, `xlink:actuate`

IE9 Mode (All Versions)

The following attributes of the XLink Attribute Module are not supported:

- **actuate**
- **arcrole**
- **role**
- **show**

- **title**
- **type**

XLink features that are related to fragment identifiers and XML Pointer Language (XPTR) are not supported.

For more information, see sections 17.1.2, 17.1.5, and 17.3 of [\[W3C-SVG1.1/2\]](#).

2.1.27 [SVG11] Section 17.5, ExternalResourcesRequired Attribute Module

V0024:

The specification states:

```
Collection Name:  External.attrib

Attributes in Collection: externalResourcesRequired
```

IE9 Mode (All Versions)

The **externalResourcesRequired** attribute is not supported.

2.1.28 [SVG11] Section 19, Animation

V0025:

The specification describes Synchronized Multimedia Integration Language (SMIL) Animation.

IE9 Mode (All Versions)

SMIL animation is not supported. In addition, no references to animation elements, properties, attributes, interfaces, or data types in other chapters of [\[W3C-SVG1.1\]](#) are supported.

For data types that can be animated, the default setting for the **animVal** attribute is **animVal** = **baseVal**.

2.1.29 [SVG11] Section 23.6, Extensibility Module

V0026:

The specification states:

```
Elements:  foreignObject

Attributes: Core.attrib, Conditional.attrib, External.attrib, Style.attrib,
Presentation.attrib, GraphicsElementEventAttrs, transform, x, y, width, height, content

Content Model:  #PCDATA
```

IE9 Mode (All Versions)

The **foreignObject** element is not supported.

2.1.30 [SVG11] Section B.5, Relationship with DOM2 events

V0060:

The specification states:

```
The SVG DOM supports the following mouse event types [ DOM2-MOUSEEVENTS]:
click
mousedown
mouseup
mouseover
mousemove
mouseout
clientX and clientY parameters for mouse events represent the mouse coordinates at which the
event occurred relative to the DOM Implementation's client area. relatedTarget is used to
identify a secondary EventTarget related to a UI event. Currently this attribute is used with
the mouseover event to indicate the EventTarget which the pointing device exited and with the
mouseout event to indicate the EventTarget which the pointing device entered.
```

IE9 Mode (All Versions)

The **title** attribute supports the **click** event even though it should not be supported.

2.1.31 [SVG11] Section C., IDL Definitions

C0024:

The specification defines the **currentView** attribute of the **SVGSVGElement** as follows:

```
readonly attribute SVGViewSpec currentView;
```

IE9 Mode (All Versions)

The **currentView** attribute is not supported.

2.2 Clarifications

The following subsections identify clarifications to recommendations made by [\[W3C-SVG1.1\]](#).

2.2.1 [SVG11] Section 4.3, Basic DOM interfaces

C0001:

The specification defines the **SVGLength** interface.

IE9 Mode (All Versions)

NO_MODIFICATION_ALLOWED_ERR exceptions are raised on **SVGLength** methods that can mutate a read-only object.

C0002:

The specification states:

```
getScreenCTM
```

Returns the transformation matrix from current user units (i.e., after application of the transform attribute, if any) to the parent user agent's notice of a "pixel". For display devices, ideally this represents a physical screen pixel. For other devices or environments where physical pixel sizes are not known, then an algorithm similar to the CSS2 definition of a "pixel" can be used instead.

No Parameters

Return value

SVGMatrix An SVGMatrix object that defines the given transformation matrix.

No Exceptions

IE9 Mode (All Versions)

Null is returned if the **getScreenCTM** element is not in the document tree.

For more information, see section 4.5.23 of [\[W3C-SVG1.1/2\]](#).

2.2.2 [SVG11] Section 5.11, DOM interfaces

C0003:

The specification defines the **getIntersectionList** and **getEnclosureList** methods of the **SVGSVGElement** interface as follows:

getIntersectionList

Returns the list of graphics elements whose rendered content intersects the supplied rectangle, honoring the 'pointer-events' property value on each candidate graphics element.

Parameters

in SVGRect rect The test rectangle. The values are in the initial coordinate system for the current 'svg' element.

in SVGElement referenceElement If not null, then only return elements whose drawing order has them below the given reference element.

Return value

NodeList A list of Elements whose content intersects the supplied rectangle.

No Exceptions

getEnclosureList

Returns the list of graphics elements whose rendered content is entirely contained within the supplied rectangle, honoring the 'pointer-events' property value on each candidate graphics element.

Parameters

in SVGRect rect The test rectangle. The values are in the initial coordinate system for the current 'svg' element.

in SVGElement referenceElement If not null, then only return elements whose drawing order has them below the given reference element.

Return value

NodeList A list of Elements whose content is enclosed by the supplied rectangle.

No Exceptions

IE9 Mode (All Versions)

The **NodeList** return value of the **getEnclosureList** and **getIntersectionList** methods is identical to the **NodeList** interface that is defined in [\[DOM Level 2 - Core\]](#), except that the interface is static and not live.

For more information, see section 5.11.2 of [\[W3C-SVG1.1/2\]](#).

2.2.3 [SVG11] Section 6.18, Aural style sheets

C0004:

The specification defines aural style sheets, which are optional.

IE9 Mode (All Versions)

Aural style sheets are not supported.

2.2.4 [SVG11] Section 7.12, Geographic Coordinate Systems

C0005:

The specification defines geographic coordinate systems, which are optional.

IE9 Mode (All Versions)

Geographic coordinate systems are not supported.

2.2.5 [SVG11] Section 8.3.9, The grammar for path data

V0027:

The specification states:

The following is the BNF for SVG paths.

```
svg-path:
    wsp* moveto-drawto-command-groups? wsp*
moveto-drawto-command-groups:
    moveto-drawto-command-group
    | moveto-drawto-command-group wsp* moveto-drawto-command-groups
moveto-drawto-command-group:
    moveto wsp* drawto-commands?
drawto-commands:
    drawto-command
    | drawto-command wsp* drawto-commands
drawto-command:
    closepath
    | lineto
    | horizontal-lineto
    | vertical-lineto
    | curveto
    | smooth-curveto
    | quadratic-bezier-curveto
```

```

    | smooth-quadratic-bezier-curve
    | elliptical-arc
moveto:
    ( "M" | "m" ) wsp* moveto-argument-sequence
moveto-argument-sequence:
    coordinate-pair
    | coordinate-pair comma-wsp? lineto-argument-sequence
closepath:
    ( "Z" | "z" )
lineto:
    ( "L" | "l" ) wsp* lineto-argument-sequence
lineto-argument-sequence:
    coordinate-pair
    | coordinate-pair comma-wsp? lineto-argument-sequence
horizontal-lineto:
    ( "H" | "h" ) wsp* horizontal-lineto-argument-sequence
horizontal-lineto-argument-sequence:
    coordinate
    | coordinate comma-wsp? horizontal-lineto-argument-sequence
vertical-lineto:
    ( "V" | "v" ) wsp* vertical-lineto-argument-sequence
vertical-lineto-argument-sequence:
    coordinate
    | coordinate comma-wsp? vertical-lineto-argument-sequence

```

IE9 Mode (All Versions)

White spaces and commas in the Backus-Naur Form (BNF) grammar are interchangeable. White spaces and commas can both delimit command groups.

2.2.6 [SVG11] Section 10.5, The 'tspan' element

C0006:

The specification defines the **rotate** attribute of the **tspan** element, with the following provisions for extra characters:

If more characters are provided than <number>s, then for each of these extra characters: (a) if an ancestor 'text' or 'tspan' element specifies a supplemental rotation for the given character via a rotate attribute, then the given supplemental rotation is applied to the given character, else (b) no supplemental rotation occurs.

If the attribute is not specified: (a) if an ancestor 'text' or 'tspan' element specifies a supplemental rotation for a given character via a rotate attribute, then the given supplemental rotation is applied to the given character (nearest ancestor has precedence), else (b) no supplemental rotation occurs.

IE9 Mode (All Versions)

If more characters are provided than <number> data types, for each of these extra characters the rotation value that is specified by the last number must be used.

If the **rotate** attribute is not specified, and if an ancestor **text** or **tspan** element specifies a supplemental rotation for a given character by using a **rotate** attribute, the given supplemental rotation is applied to the given character (the nearest ancestor has precedence). If there are more

characters than `<number>` data types specified in the ancestor's **rotate** attribute, for each of these extra characters the rotation value that is specified by the last number must be used.

2.2.7 [SVG11] Section 10.9.1, Text alignment properties

C0007:

The specification defines the **start** value of the **text-anchor** property as follows:

start
The rendered characters are aligned such that the start of the text string is at the initial current text position. For Latin or Arabic, which is usually rendered horizontally, this is comparable to left alignment. For Asian text with a vertical primary text direction, this is comparable to top alignment.

IE9 Mode (All Versions)

The rendered characters are aligned such that the start of the text string is at the initial current text position. For Latin script that has a left-to-right text orientation, this alignment is equivalent to left alignment. For scripts that have a right-to-left text orientation, such as Hebrew and Arabic, this alignment is equivalent to right alignment. For Asian scripts that have a vertical text orientation, this alignment is equivalent to top alignment.

For more information, see section 10.9.1 of [\[W3C-SVG1.1/2\]](#).

2.2.8 [SVG11] Section 10.19, DOM interfaces

C0008:

The specification defines the **getSubStringLength** method of the **SVGTextContentElement** interface as follows:

getSubStringLength
The total sum of all of the advance values from rendering the specified substring of the characters, including the advance value on the glyphs (horizontal or vertical), the effect of properties 'kerning', 'letter-spacing' and 'word-spacing' and adjustments due to attributes `dx` and `dy` on 'tspan' elements. For non-rendering environments, the user agent shall make reasonable assumptions about glyph metrics.

Parameters
in unsigned long `charnum` The index of the first character in the substring, where the first character has an index of 0.
in unsigned long `nchars` The number of characters in the substring.

Return value
float The text advance distance.

Exceptions
`DOMException INDEX_SIZE_ERR`: Raised if the `charnum` is negative or if `charnum+nchars` is greater than or equal to the number of characters at this node.

IE9 Mode (All Versions)

The **nchars** parameter describes the number of characters in the substring. If **nchars** specifies more characters than are available, the substring consists of all characters starting with the **charnum** parameter until the end of the list of characters.

For more information, see section 10.17.1 of [\[W3C-SVG1.1/2\]](#).

2.2.9 [SVG11] Section 11, Painting: Filling, Stroking and Marker Symbols

C0009:

The specification includes links to "solid color" and "solid colors."

IE9 Mode (All Versions)

Solid color elements and properties are not implemented and do not exist in the specification.

2.2.10 [SVG11] Section 11.4, Stroke Properties

C0010:

The specification describes the **stroke** property as follows:

A subpath consisting of a moveto and lineto to the same exact location or a subpath consisting of a moveto and a closepath will be stroked only if the 'stroke-linecap' property is set to "round", producing a circle centered at the given point.

IE9 Mode (All Versions)

Any zero-length subpath is not stroked if the **stroke-linecap** property has a value of `butt`. The subpath is stroked if the **stroke-linecap** property has a value of `round` or `square`, producing respectively a circle or a square centered at the given point.

The following examples demonstrate zero-length subpaths:

- M 10,10 L 10,10
- M 20,20 h 0
- M 30,30 z
- M 40,40 c 0,0 0,0 0,0

For more information, see section F.5 of [\[W3C-SVG1.1/2\]](#).

2.2.11 [SVG11] Section 11.7.3, The 'shape-rendering' property

V0028:

The specification defines the **shape-rendering** property.

IE9 Mode (All Versions)

The **shape-rendering** property is not supported.

2.2.12 [SVG11] Section 13.3, Patterns

C0022:

The specification states:

SVG's user agent style sheet sets the 'overflow' property for 'pattern' elements to hidden, which causes a rectangular clipping path to be created at the bounds of the pattern tile. Unless the 'overflow' property is overridden, any graphics within the pattern which goes outside of the pattern rectangle will be clipped. Example pattern01 below shows the effect of clipping to the pattern tile.

IE9 Mode (All Versions)

The **overflow** property on **pattern** elements is always treated as if the value `hidden` is assigned to it; it cannot be overridden.

2.2.13 [SVG11] Section 14.3.3, The 'overflow' and 'clip' properties

C0011:

The specification describes the **overflow** property as follows:

- The initial value for 'overflow' as defined in [CSS2-overflow] is 'visible'; however, SVG's user agent style sheet overrides this initial value and set the 'overflow' property on elements that establish new viewports (e.g., 'svg' elements), 'pattern' elements and 'marker' elements to the value 'hidden'.

IE9 Mode (All Versions)

On the outermost **svg** element that is inline in HTML5, the initial value for the **overflow** property is `visible`.

2.2.14 [SVG11] Section 16.2, Complete list of supported events

V0029:

The specification states:

Event name: `SVGLoad`

Description: The event is triggered at the point at which the user agent has fully parsed the element and its descendants and is ready to act appropriately upon that element, such as being ready to render the element to the target device. Referenced external resources that are required must be loaded, parsed and ready to render before the event is triggered. Optional external resources are not required to be ready for the event to be triggered.

DOM2 name: `(same)`

DOM2 category: `none`

Event attribute name: `onload`

IE9 Mode (All Versions)

The **SVGLoad** event does not bubble.

For more information, see section 16.2 of [\[W3C-SVG1.1/2\]](#).

2.2.15 [SVG11] Section B.1, SVG DOM Overview

C0012:

The specification describes the SVG DOM and specific SVG DOM interfaces.

IE9 Mode (All Versions)

When an attribute is accessed that has not been specified in the document (for example, the **x** attribute is not provided in **x.baseVal**), a corresponding SVG DOM object is created and initialized by using the value that is used to render when the attribute is accessed. The returned object does not affect rendering until it is changed for the first time. After the first change, the object becomes live so that any changes to the corresponding attribute are immediately reflected in the object.

For more information, see section B.1.1 of [\[W3C-SVG1.1/2\]](#).

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-ES5] protocol document between the February 2011 and March 2011 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.
- Changes made for template compliance.
- Removal of a document from the documentation set.

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 revision 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.

Section	Tracking number (if applicable) and description	Major change (Y or N)	Change Type
2.1.6 [SVG11] Section 8.5, DOM Interfaces	Added section to discuss the pathLength attribute.	Y	New content added.
2.1.8 [SVG11] Section 10.9.2, Baseline alignment properties	Added section to discuss the alignment-baseline property.	Y	New content added.
2.1.31 [SVG11] Section C., IDL Definitions	Added section to discuss the currentView attribute.	Y	New content added.

4 Index

A

[Alternate glyphs](#) 11
[An example](#) 15
[Animation](#) 17
[Aural style sheets](#) 20

B

[Baseline alignment properties](#) 10
Basic DOM interfaces ([section 2.1.1](#) 7, [section 2.2.1](#) 18)

C

[Change tracking](#) 26
[Color interpolation properties: 'color-interpolation' and 'color-interpolation-filters'](#) 13
[Complete list of supported events](#) 24
[Cursor Module](#) 16

D

DOM interfaces ([section 2.1.6](#) 9, [section 2.1.18](#) 13, [section 2.2.2](#) 19, [section 2.2.8](#) 22)

E

Elements
 [:style](#) 8
 [:tspan](#) 21
[Establishing a new clipping path](#) 15
[Extensibility Module](#) 17
[ExternalResourcesRequired Attribute Module](#) 17

G

[Geographic Coordinate Systems](#) 20
[Glossary](#) 5
[Glyph orientation within a text run](#) 9

I

[IDL Definitions](#) 18
[Informative references](#) 5
[Introduction](#) 5

M

[Magnification and panning](#) 15

N

[Normative references](#) 5

O

[Object bounding box units](#) 8

P

[Painting: Filling - Stroking and Marker Symbols](#) 23
Patterns ([section 2.1.19](#) 14, [section 2.2.12](#) 23)
Properties
 [:clip](#) 14
 [:color-rendering](#) 13
 [:image-rendering](#) 13
 [:overflow](#) 24
 [:shape-rendering](#) 23
 [:text-rendering](#) 13

R

References
 [informative](#) 5
 [normative](#) 5
[Relationship with DOM2 events](#) 18

S

[Spacing properties](#) 11
[Stroke Properties](#) 23
[SVG DOM Overview](#) 25
[SVG fragment identifiers](#) 16

T

[Text alignment properties](#) 22
[Text Module](#) 12
[The 'color-rendering' property](#) 13
[The grammar for path data](#) 20
[The 'image-rendering' property](#) 13
[The 'marker' element](#) 12
The 'overflow' and 'clip' properties ([section 2.1.20](#) 14, [section 2.2.13](#) 24)
[The 'shape-rendering' property](#) 23
[The 'style' element](#) 8
[The 'text-rendering' property](#) 13
[The 'tspan' element](#) 21
[The viewBox attribute](#) 8
[The xml:lang and xml:space attributes](#) 8
[Tracking changes](#) 26

W

[White space handling](#) 11

X

[XLink Attribute Module](#) 16