APPENDIX B

SVG Elements Reference

**a**
Denotes a hyperlink, analogous to the `a` element in HTML. All enclosed nodes will link to the URI specified in the XLink `href` attribute. Use the `target` attribute to specify a window to open the new page in. The SVG user agent needs to support only simple XLink links.

**Example**
```
<a xlink:href="somepage.svg">
  <circle cx="45" cy="45" r="25" fill="red" />
</a>
```

**Parents**
a, defs, g, glyph, marker, mask, missing-glyph, pattern, svg, switch, symbol, text, textPath, tspan

**Children**
a, altGlyphDef, animate, animateColor, animateMotion, animateTransform, circle, clipPath, color-profile, cursor, defs, desc, ellipse, filter, font, font-face, g, image, line, linearGradient, marker, mask, metadata, path, pattern, polygon, polyline, radialGradient, rect, script, set, style, svg, switch, symbol, text, title, use, view

**Attributes**
alignment-baseline, baseline-shift, class, clip, clip-path, clip-rule, color, color-interpolation, color-interpolation-filters, color-profile, color-rendering, cursor, direction, display, dominant-baseline, enable-background, externalResourcesRequired, fill, fill-opacity, fill-rule, filter, flood-color, flood-opacity, font-family, font-size,

**altGlyph**
Maps the characters inside the `altGlyph` element to an alternative glyph or glyph set—for example, a mathematical symbol. `altGlyph` references (XLink `href` attribute) either a glyph or `altGlyphDef` element.

**Parents**
text, textPath, tspan

**Attributes**

**altGlyphDef**
Contains either one or more `glyphRef` elements (references a glyph to use) or a set of `altGlyphItem` elements. In the latter case, the user agent should choose the first `altGlyphItem` element where all references in the contained `glyphRef` elements point to an existing glyph.
Parents
a, defs, g, glyph, marker, mask, missing-glyph, pattern, svg, symbol

Children
altGlyphItem, glyphRef

Attributes
id, xml:base

altGlyphItem
Contains one or more glyphRef elements. The first altGlyphItem in an altGlyphDef, where all glyphRef elements point to available glyphs, is chosen to represent an alternative glyph.

Parents
altGlyphDef

Children
glyphRef

Attributes
id, xml:base

animate
Animates an attribute on the parent or referenced (XLink href attribute) element based on a list of values in the values attribute or an interpolation of the value in the from attribute to the value in the to attribute. The animation can be controlled by events or timing (begin and end attributes).

Example
<circle cx="67" cy="86" r="32" fill="#542354">
  <animate attributeType="XML" attributeName="cx" from="67" to="145"
  ➤dur="2s" repeatCount="indefinite" />
</circle>

Parents
a, circle, clipPath, defs, ellipse, feBlend, feColorMatrix, feComposite, feConvolveMatrix, feDiffuseLighting, feDisplacementMap, feDistantLight, feFlood,
feFuncA, feFuncB, feFuncG, feFuncR, feGaussianBlur, feImage, feMergeNode, feMorphology, feOffset, fePointLight, feSpecularLighting, feSpotLight, feTile, feTurbulence, filter, g, glyph, image, line, linearGradient, marker, mask, missing-glyph, path, pattern, polygon, polyline, radialGradient, rect, stop, svg, switch, symbol, text, textPath, tref, tspan, use

**Children**
desc, metadata, title

**Attributes**
accumulate, additive, attributeName, attributeType, begin, by, calcMode, dur, end, externalResourcesRequired, fill, from, id, keySplines, keyTimes, max, min, onbegin, onend, onrepeat, repeatCount, repeatDur, requiredExtensions, requiredFeatures, restart, systemLanguage, to, values, xlink:actuate, xlink:arcrole, xlink:href, xlink:role, xlink:show, xlink:title, xlink:type, xml:base, xmlns:xlink

**animateColor**
Animates the color of the parent or referenced (XLink href attribute) element. The animation values are set as in the animate element.

**Example**
```
<circle cx="67" cy="86" r="32" fill="#542354">
  <animateColor attributeName="fill" from="#542354" to="red" dur="2s" repeatCount="indefinite" />
</circle>
```

**Parents**
a, circle, clipPath, defs, ellipse, feDiffuseLighting, feFlood, feSpecularLighting, g, glyph, image, line, marker, mask, missing-glyph, path, pattern, polygon, polyline, rect, stop, svg, switch, symbol, text, textPath, tref, tspan, use

**Children**
desc, metadata, title

**Attributes**
accumulate, additive, attributeName, attributeType, begin, by, calcMode, dur, end, externalResourcesRequired, fill, from, id, keySplines, keyTimes, max, min, onbegin, onend, onrepeat, repeatCount, repeatDur, requiredExtensions, requiredFeatures, restart, systemLanguage, to, values, xlink:actuate, xlink:arcrole, xlink:href, xlink:role, xlink:show, xlink:title, xlink:type, xml:base, xmlns:xlink
animateMotion

Animates the parent or referenced (XLink href attribute) element along a path defined using the path attribute or the mpath subelement.

Example

```xml
<path id="myPath" d="M100,200 C100,100 250,100 250,200 S400,300 400,200"
   fill="none" stroke="black" />

<rect x="23" y="45" width="40" height="40" fill="yellow">
   <animateMotion dur="6s" repeatCount="1">
      <mpath xlink:href="#myPath" />
   </animateMotion>
</rect>
```

Parents

a, circle, clipPath, defs, ellipse, g, glyph, image, line, marker, mask, missing-glyph, path, pattern, polygon, polyline, rect, svg, switch, symbol, text, use

Children

desc, metadata, mpath, title

Attributes

accumulate, additive, begin, by, calcMode, dur, end, externalResourcesRequired, fill, from, id, keyPoints, keySplines, keyTimes, max, min, onbegin, onend, onrepeat, origin, path, repeatCount, repeatDur, requiredExtensions, requiredFeatures, restart, rotate, systemLanguage, to, values, xlink:actuate, xlink:arcrole, xlink:href, xlink:role, xlink:show, xlink:title, xlink:type, xml:base, xmlns:xlink

animateTransform

Animates a transformation (translation, scaling, rotation, or skewing is specified in the type attribute). The animation values are set as in the animate element.

Example

```xml
<rect x="200" y="200" width="40" height="40" fill="#FF00FF">
   <animateTransform attributeName="transform" attributeType="XML"
      type="rotate" dur="6s" repeatCount="infinite"
      from="0 220 220" to="360 220 220" />
</rect>
```
Parents
a, circle, clipPath, defs, ellipse, feImage, g, glyph, image, line, linearGradient, marker, mask, missing-glyph, path, pattern, polygon, polyline, radialGradient, rect, svg, switch, symbol, text, use

Children
desc, metadata, title

Attributes
accumulate, additive, attributeName, attributeType, begin, by, calcMode, dur, end, externalResourcesRequired, fill, from, id, keySplines, keyTimes, max, min, onbegin, onend, onrepeat, repeatCount, repeatDur, requiredExtensions, requiredFeatures, restart, systemLanguage, to, type, values, xlink:actuate, xlink:arcrole, xlink:href, xlink:role, xlink:show, xlink:title, xlink:type, xml:base, xmlns:xlink

circle
Defines a circular shape with a center (cx and cy attributes) and radius (r attribute).

Example
<circle cx="100" cy="100" r="45" stroke="blue" fill="none" />

Parents
a, clipPath, defs, g, glyph, marker, mask, missing-glyph, pattern, svg, switch, symbol

Children
animate, animateColor, animateMotion, animateTransform, desc, metadata, set, title

Attributes
class, clip-path, clip-rule, color, color-interpolation, color-rendering, cursor, cx, cy, display, externalResourcesRequired, fill, fill-opacity, fill-rule, filter, id, image-rendering, mask, onactivate, onclick, onfocusin, onfocusout, onload, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, opacity, pointer-events, r, requiredExtensions, requiredFeatures, shape-rendering, stroke, stroke-dasharray, stroke-dashoffset, stroke-linecap, stroke-linejoin, stroke-miterlimit, stroke-opacity, stroke-width, style, systemLanguage, text-rendering, transform, visibility, xml:base, xml:lang, xml:space
**clipPath**

The subelements of the clipPath element define a shape used to specify a clip area on any container or graphical shape, using the clip-path property.

**Parents**
a,defs,g,glyph,marker,mask,missing-glyph,pattern,svg,symbol

**Children**
animate,animateColor,animateMotion,animateTransform,circle,desc,ellipse,line,metadata,path,polygon,polyline,rect,set,text,title,use

**Attributes**

**color-profile**

Creates a name (name attribute) of a color identifier in SVG, which points to a color in an International Color Consortium (ICC) profile located locally (local attribute) or at a URI (XLink href attribute).

**Parents**
a,defs,g,glyph,marker,mask,missing-glyph,pattern,svg,symbol

**Children**
desc,metadata,title

**Attributes**
**cursor**

Describes a user-defined cursor by linking to an image (XLink `href` attribute). PNG images are recommended as the source of the cursor because the PNG format supports transparency. The coordinates of the hot-spot are specified by the `x` and `y` attributes.

**Example**

```xml
<cursor x="3" y="4" xlink:href="my_cursor.png" />
```

**Parents**

`a`, `defs`, `g`, `glyph`, `marker`, `mask`, `missing-glyph`, `pattern`, `svg`, `symbol`

**Children**

`desc`, `metadata`, `title`

**Attributes**


**definition-src**

Provides a URI (XLink `href` attribute) to a font definition.

**Parents**

`font-face`

**Attributes**


**defs**

Defines a group of elements for reference. The `defs` element behaves like a `g` element that is not rendered. Instead, it provides a method for defining elements, such as gradients, patterns, markers, or symbols, that can be referenced and used by other elements.

**Example**

```xml
<defs>
  <rect id="myRect" x="56" y="56" width="15" height="48" fill="#FF0000" />
</defs>
```
Parents
a, defs, g, glyph, marker, mask, missing-glyph, pattern, svg, symbol

Children
a, altGlyphDef, animate, animateColor, animateMotion, animateTransform, circle, clipPath, color-profile, cursor, defs, desc, ellipse, filter, font, font-face, g, image, line, linearGradient, marker, mask, metadata, path, pattern, polygon, polyline, radialGradient, rect, script, set, style, svg, switch, symbol, text, title, use, view

Attributes

desc
Specifies a text description for an element. The viewer implementation can use the description for aural or other nongraphical presentation of the content. The desc element can also contain other XML data under a separate namespace.

Example
<polygon points="23,45 78,98 234,54" fill="yellow">
  <title>A yellow polygon</title>
  <desc>This is a yellow polygon primitive with three points</desc>
</polygon>
Parents

a, animate, animateColor, animateMotion, animateTransform, circle, clipPath,
color-profile, cursor, defs, ellipse, filter, font, font-face, g, glyph, image, line,
linearGradient, marker, mask, missing-glyph, mpath, path, pattern, polygon, polyline,
radialGradient, rect, set, svg, switch, symbol, text, textPath, tref, tspan, use, view

Attributes

class, content, id, style, xml:base, xml:lang, xml:space

ellipse

Describes an ellipse primitive with a center (cx and cy attributes) and two radii (rx and ry attributes).

Example

<ellipse cx="300" cy="300" rx="200" ry="100" stroke="red" stroke-width="5"
fill="blue" />

Parents

a, clipPath, defs, g, glyph, marker, mask, missing-glyph, pattern, svg, switch, symbol

Children

animate, animateColor, animateMotion, animateTransform, desc, metadata, set, title

Attributes

class, clip-path, clip-rule, color, color-interpolation, color-rendering, cursor, cx,
cy, display, externalResourcesRequired, fill, fill-opacity, fill-rule, filter, id,
image-rendering, mask, onactivate, onclick, onfocusin, onfocusout, onload, onmousedown,
onmousemove, onmouseout, onmouseover, onmouseup, opacity, pointer-events, requiredExtensions,
requiredFeatures, rx, ry, shape-rendering, stroke, stroke-dasharray, stroke-dashoffset,
stroke-linecap, stroke-linejoin, stroke-miterlimit, stroke-opacity, stroke-width, style,
systemLanguage, text-rendering, transform, visibility, xml:base, xml:lang, xml:space

feBlend

Blends two objects (in and in2 attributes) according to the method specified in the mode attribute.
Parents
filter

Children
animate, set

Attributes
color-interpolation-filters, height, id, in, in2, mode, result, width, x, xml:base, y

feColorMatrix
Transforms the color of each pixel in an image (in attribute) depending on the type of modification (type attribute) and a matrix (values attribute).

Parents
filter

Children
animate, set

Attributes
color-interpolation-filters, height, id, in, result, type, values, width, x, xml:base, y

feComponentTransfer
Transforms the color of each pixel in an image (in attribute) with the functions specified in the subelements: feFuncA, feFuncB, feFuncG, and feFuncR.

Parents
filter

Children
feFuncA, feFuncB, feFuncG, feFuncR

Attributes
color-interpolation-filters, height, id, in, result, width, x, xml:base, y
feComposite
Combines two images (in and in2 attributes) using a Porter-Duff operation by the method specified in the operator attribute.

Parents
filter

Children
animate, set

Attributes
color-interpolation-filters, height, id, in, in2, k1, k2, k3, k4, operator, result, width, x, xml:base, y

feConvolveMatrix
Modifies an image (in attribute) by calculating the value for each pixel as a function of its neighboring pixels, based on a convolution matrix. The feConvolveMatrix element can be used to create, for example, blurring and edge-finding effects.

Parents
filter

Children
animate, set

Attributes
bias, color-interpolation-filters, divisor, edgeMode, height, id, in, kernelMatrix, kernelUnitLength, order, preserveAlpha, result, targetX, targetY, width, x, xml:base, y

feDiffuseLighting
Lights an image (in attribute) based on the alpha-channel as a bump map and light sources in the feDistantLight, fePointLight, and feSpotLight child elements.

Parents
filter
Children
animate, animateColor, feDistantLight, fePointLight, feSpotLight, set

Attributes
class, color, color-interpolation, color-interpolation-filters, color-rendering,
diffuseConstant, height, id, in, kernelUnitLength, lighting-color, result, style,
surfaceScale, width, x, xml:base, y

feDisplacementMap
Spatially displaces an image (in attribute) based on another image (in2 attribute).

Parents
filter

Children
animate, set

Attributes
color-interpolation-filters, height, id, in, in2, result, scale, width, x,
xChannelSelector, xml:base, y, yChannelSelector

feDistantLight
Used in the feDiffuseLighting and feSpecularLighting elements to describe a parallel light source defined by the angle toward the XY (azimuth attribute) and YZ (elevation attribute) planes.

Parents
feDiffuseLighting, feSpecularLighting

Children
animate, set

Attributes
azimuth, elevation, id, xml:base

feFlood
Fills the rectangle specified by the x, y, width, and height attributes with a color (flood-color attribute) and opacity (flood-opacity attribute) in an image (in attribute).
Parents
filter

Children
animate, animateColor, set

Attributes
class, color, color-interpolation, color-interpolation-filters, color-rendering,
flood-color, flood-opacity, height, id, in, result, style, width, x, xml:base, y

feFuncA
Used in the feComponentTransfer element to change the alpha component of a pixel according to the type of function specified in the type attribute.

Parents
feComponentTransfer

Children
animate, set

Attributes
amplitude, exponent, id, intercept, offset, slope, tableValues, type, xml:base

feFuncB
Used in the feComponentTransfer element to change the blue component of a pixel according to the type of function specified in the type attribute.

Parents
feComponentTransfer

Children
animate, set

Attributes
amplitude, exponent, id, intercept, offset, slope, tableValues, type, xml:base
**feFuncG**

Used in the feComponentTransfer element to change the green component of a pixel according to the type of function specified in the type attribute.

**Parents**

feComponentTransfer

**Children**

animate, set

**Attributes**

amplitude, exponent, id, intercept, offset, slope, tableValues, type, xml:base

---

**feFuncR**

Used in the feComponentTransfer element to change the red component of a pixel according to the type of function specified in the type attribute.

**Parents**

feComponentTransfer

**Children**

animate, set

**Attributes**

amplitude, exponent, id, intercept, offset, slope, tableValues, type, xml:base

---

**feGaussianBlur**

Blurs an image (in attribute), using the Gaussian blur algorithm, by the amount specified in the stdDeviation attribute.

**Example**

```xml
<defs>
  <filter id="myFilter" filterUnits="userSpaceOnUse" x="0" y="0"
    width="2000" height="2000">
    <feGaussianBlur in="SourceGraphic" stdDeviation="1" />
  </filter>
</defs>
<circle cx="100" cy="100" r="30" fill="orange" filter="url(#myFilter)"/>
```
Parents
filter

Children
animate, set

Attributes
color-interpolation-filters, height, id, in, result, stdDeviation, width, x, xml:base, y

feImage
References (XLink href attribute) an external image or fragment of an SVG image to add to the filter.

Parents
filter

Children
animate, animateTransform, set

Attributes
alignment-baseline, baseline-shift, class, clip, clip-path, clip-rule, color,
color-interpolation, color-interpolation-filters, color-profile, color-rendering,
cursor, direction, display, dominant-baseline, enable-background,
externalResourcesRequired, fill, fill-opacity, fill-rule, filter, flood-color,
flood-opacity, font-family, font-size, font-size-adjust, font-stretch, font-style,
font-variant, font-weight, glyph-orientation-horizontal,
glyph-orientation-vertical, id, image-rendering, kerning, letter-spacing,
lighting-color, marker-end, marker-mid, marker-start, mask, opacity, overflow,
pointer-events, shape-rendering, stop-color, stop-opacity, stroke, stroke-dasharray,
stroke-dashoffset, stroke-linecap, stroke-linejoin, stroke-miterlimit,
stroke-opacity, stroke-width, style, text-anchor, text-decoration, text-rendering,
unicode-bidi, visibility, word-spacing, writing-mode, xlink:actuate, xlink:arcrole,
xlink:href, xlink:role, xlink:show, xlink:title, xlink:type, xml:base, xml:lang,
xml:space, xmlns:xlink
feMerge
Merges all images specified in the feMergeNode subelements into one image.

Parents
filter

Children
feMergeNode

Attributes
color-interpolation-filters, height, id, result, width, x, xml:base, y

feMergeNode
References an input image (in attribute) to merge using the feMerge element.

Parents
feMerge

Children
animate, set

Attributes
id, in, xml:base

feMorphology
Fattens or thins an image (in attribute) based on the values of the operator and radius attributes.

Parents
filter

Children
animate, set

Attributes
color-interpolation-filters, height, id, in, operator, radius, result, width, x, xml:base, y
feOffset
Offsets the image (in attribute) by the values specified in the dx and dy attributes. For example, the feOffset element is used when creating drop shadows.

Parents
filter

Children
animate, set

Attributes
color-interpolation-filters, dx, dy, height, id, in, result, width, x, xml:base, y

fePointLight
A light source defined by the position of the light center (x, y, and z attributes). Used in the feDiffuseLighting and feSpecularLighting elements.

Parents
feDiffuseLighting, feSpecularLighting

Children
animate, set

Attributes
id, x, xml:base, y, z

feSpecularLighting
Lights an image (in attribute) based on the alpha-channel as a bump map and light sources in the feDistantLight, fePointLight, and feSpotLight child elements. The result is added to the input image.

Parents
filter

Children
animate, animateColor, feDistantLight, fePointLight, feSpotLight, set
Attributes
class, color, color-interpolation, color-interpolation-filters, color-rendering, height, id, in, kernelUnitLength, lighting-color, result, specularConstant, specularExponent, style, surfaceScale, width, x, xml:base, y

feSpotLight
A light source defined as a light cone with an origin (x, y, and z attributes) and target (pointsAtX, pointsAtY, and pointsAtZ attributes). The size of the cone is set in the limitingConeAngle attribute. Used in the feDiffuseLighting and feSpecularLighting elements.

Parents
feDiffuseLighting, feSpecularLighting

Children
animate, set

Attributes
id, limitingConeAngle, pointsAtX, pointsAtY, pointsAtZ, specularExponent, x, xml:base, y, z

feTile
Tiles the input image (in attribute) onto the rectangle defined by the x, y, width, and height attributes.

Parents
filter

Children
animate, set

Attributes
color-interpolation-filters, height, id, in, result, width, x, xml:base, y

feTurbulence
Creates an image by using the Perlin turbulence algorithm. The feTurbulence element can be used to create organic textures such as clouds.
A filter is a set of graphical operations (filter primitives) that can be applied to any graphic or container element using the filter attribute. A filter primitive starts with an image (in attribute) and produces a result image (result attribute). The result image can be used in other filter primitives. This way, several filter primitives can be applied to an element in many different configurations. The filter modifies the rendering of the element to create effects such as shading, inversion, or blurring.

Example

```xml
<defs>
  <filter id="myFilter" filterUnits="userSpaceOnUse" x="0" y="0" width="2000" height="2000">
    <feGaussianBlur in="SourceGraphic" stdDeviation="1" />
  </filter>
</defs>

<circle cx="100" cy="100" r="30" fill="orange" filter="url(#myFilter)"/>
```

Parents

```
fILTER
```

Children

```
ANIMATE, SET
```

Attributes

```
BASEFREQUENCY, COLOR-INTERPOLATION-FILTERS, HEIGHT, ID, NUMOCTAVES, RESULT, SEED, STITCHTILES, TYPE, WIDTH, X, XML:BASE, Y
```

filter

A filter is a set of graphical operations (filter primitives) that can be applied to any graphic or container element using the filter attribute. A filter primitive starts with an image (in attribute) and produces a result image (result attribute). The result image can be used in other filter primitives. This way, several filter primitives can be applied to an element in many different configurations. The filter modifies the rendering of the element to create effects such as shading, inversion, or blurring.

Example

```xml
<defs>
  <filter id="myFilter" filterUnits="userSpaceOnUse" x="0" y="0" width="2000" height="2000">
    <feGaussianBlur in="SourceGraphic" stdDeviation="1" />
  </filter>
</defs>

<circle cx="100" cy="100" r="30" fill="orange" filter="url(#myFilter)"/>
```
cursor, direction, display, dominant-baseline, enable-background, 
externalResourcesRequired, fill, fill-opacity, fill-rule, filter, filterRes, 
filterUnits, flood-color, flood-opacity, font-family, font-size, font-size-adjust, 
font-stretch, font-style, font-variant, font-weight, glyph-orientation-horizontal, 
glyph-orientation-vertical, height, id, image-rendering, kerning, letter-spacing, 
lighting-color, marker-end, marker-mid, marker-start, mask, opacity, overflow, 
pointer-events, primitiveUnits, shape-rendering, stop-color, stop-opacity, stroke, 
stroke-dasharray, stroke-dashoffset, stroke-linecap, stroke-linejoin, 
stroke-miterlimit, stroke-opacity, stroke-width, style, text-anchor, 
text-decoration, text-rendering, unicode-bidi, visibility, width, word-spacing, 
writing-mode, x, xlink:actuate, xlink:arcrole, xlink:href, xlink:role, xlink:show, 
xlink:title, xlink:type, xml:base, xml:lang, xml:space, xmlns:xlink, y

font

Specifies an SVG font. The font is further described using several subelements (for 
example, font-face, glyph).

Example
<defs>
  <font id="myFont">
    <font-face font-family="My Minimal font"></font-face>
    <glyph unicode="o">
      <circle cx="0" cy="400" r="400" stroke="black" stroke-width="100" 
fill="none" />
    </glyph>
    <missing-glyph>
      <line x1="-400" y1="0" x2="400" y2="0" fill="none" stroke="black" 
stroke-width="100"/> 
    </missing-glyph>
  </font>
</defs>

<text x="100" y="100" font-family="My Minimal font">Some text</text>

Parents
a, defs, g, glyph, marker, mask, missing_glyph, pattern, svg, symbol

Children
desc, font-face, glyph, hkern, metadata, missing_glyph, title, vkern
Attributes

font-face
Describes a font using several font characteristics (for example, font-family, font-style, font-weight attributes).

Example
See the font element example.

Parents
a, defs, font, g, glyph, marker, mask, missing-glyph, pattern, svg, symbol

Children
definition-src, desc, font-face-src, metadata, title

Attributes

font-face-format
A comma-separated list of formats available at the URI specified in the font-face-uri element. By using this element, the user agent can avoid downloading fonts of unsupported formats.
**Parents**
font-face-url

**Attributes**
id, string, xml:base

**font-face-name**
Identifies a locally installed font by its name.

**Parents**
font-face-src

**Attributes**
id, name, xml:base

**font-face-src**
References a font description as described by the font-face-format, font-face-name, and font-face-uri subelements.

**Parents**
font-face

**Children**
font-face-name, font-face-uri

**Attributes**
id, xml:base

**font-face-uri**
Contains the URI to a downloadable external font description.

**Parents**
font-face-src

**Children**
font-face-format
Attributes

foreignObject
Embeds any XML data into an SVG document and allows the user agent to use some external XML processor to render the data. Current transformations and CSS are used to position and format the XML. The processing and rendering of the data are not certain to be supported by the user agent. Therefore, foreignObject is used as a child node to a switch element, which supplies alternatives if the user agent cannot handle the foreignObject data.

Example
<switch>
  <foreignObject x="45" y="70" width="200" height="50" requiredExtensions="http://domain.com/SVGExtensions/MathML">
    <math xmlns="http://www.w3.org/1998/Math/MathML" >
      <apply>
        <minus/>
        <ci>a</ci>
      </apply>
      <apply>
        <plus/>
        <ci>b</ci>
        <ci>c</ci>
      </apply>
    </math>
  </foreignObject>
  <text x="45" y="70" font-family="Arial" font-size="12">a - (b + c)</text>
</switch>

Parents
switch

Attributes
alignment-baseline, baseline-shift, class, clip, clip-path, clip-rule, color, color-interpolation, color-interpolation-filters, color-profile, color-rendering,
**g**

Defines a group of one or more elements. The `g` element is used to structure a document and, if described using `title` and `desc` elements, can enhance accessibility.

**Example**

```xml
<g id="myGroup" transform="translate(34,30) rotate(45)">
  <title>Two connected circles</title>
  <circle cx="10" cy="10" r="8" fill="none" stroke="black" />
  <circle cx="40" cy="10" r="8" fill="none" stroke="black" />
  <line x1="10" y1="10" x2="40" y2="10" stroke="red" />
</g>
```

**Parents**

`a`, `defs`, `g`, `glyph`, `marker`, `mask`, `missing-glyph`, `pattern`, `svg`, `switch`, `symbol`

**Children**

`a`, `altGlyphDef`, `animate`, `animateColor`, `animateMotion`, `animateTransform`, `circle`, `clipPath`, `color-profile`, `cursor`, `desc`, `ellipse`, `filter`, `font`, `font-face`, `g`, `image`, `line`, `linearGradient`, `marker`, `mask`, `metadata`, `path`, `pattern`, `polygon`, `polyline`, `radialGradient`, `rect`, `script`, `set`, `style`, `svg`, `switch`, `symbol`, `text`, `title`, `use`, `view`

**Attributes**


glyph
Defines the shape and properties for a single glyph within a font. The shape can be defined either as a path (d attribute) or by enclosed subelements.

Example
See the font element example.

Parents
font

Children
a, altGlyphDef, animate, animateColor, animateMotion, animateTransform, circle, clipPath, color-profile, cursor, defs, desc, ellipse, filter, font, font-face, g, image, line, linearGradient, marker, mask, metadata, path, pattern, polygon, polyline, radialGradient, rect, script, set, style, svg, switch, symbol, text, title, use, view

Attributes
**glyphRef**
References a glyph element using the XLink href attribute or by font selection with several other attributes (for example, glyphRef, format).

**Parents**
altGlyphDef, altGlyphItem

**Attributes**

**hkern**
Specifies a kerning value (k attribute) for a pair of Unicode character sets (u1, g1, u2, and g2 attributes) or glyphs for horizontal text.

**Parents**
font

**Attributes**
g1, g2, id, k, u1, u2, xml:base

**image**
Inserts a bitmap (GIF, JPG, or PNG) or SVG image using the XLink href attribute. The image can be external or Base64-encoded inline data (data protocol). If the image is an SVG file, the image element will establish a new viewport, and the included SVG structure will be accessible using the Document Object Model (DOM).

**Example**
<image x="70" y="100" width="200" height="200" xlink:href="somepage.svg" />

**Parents**
a, defs, g, glyph, marker, mask, missing-glyph, pattern, svg, switch, symbol

**Children**
animate, animateColor, animateMotion, animateTransform, desc, metadata, set, title
**Attributes**

**line**
Draws a line from one point (x1 and y1 attributes) to another (x2 and y2 attributes).

**Example**
```html
<line x1="60" y1="34" x2="400" y2="100" stroke="#646464"
  stroke-dasharray="9,7" fill="none" />
```

**Parents**
a, clipPath,defs,g, glyph, marker, mask, missing-glyph, pattern, svg, switch, symbol

**Children**
animate, animateColor, animateMotion, animateTransform, desc, metadata, set, title

**Attributes**
class, clip-path, clip-rule, color, color-interpolation, color-rendering, cursor, display, externalResourcesRequired, fill, fill-opacity, fill-rule, filter, id, image-rendering, marker-end, marker-mid, marker-start, mask, onactivate, onclick, onfocusin, onfocusout, onload, onmousedown, onmousemove, onmouseout, onmouseover, opacity, pointer-events, requiredExtensions, requiredFeatures, shape-rendering, stroke, stroke-dasharray, stroke-dashoffset, stroke-linecap, stroke-linejoin, stroke-miterlimit, stroke-opacity, stroke-width, style, systemLanguage, text-rendering, transform, visibility, x1, x2, xml:base, xml:lang, xml:space, y1, y2

**linearGradient**
Defines a linear gradient by using stop subelements that define positions and colors within the gradient. The gradient can be modified using a gradient vector (x1, y1, x2, y2 attributes) or transformed (gradientTransform attribute). Using the XLink href attribute, a gradient can inherit the properties of another gradient.
Example
<defs>
  <linearGradient id="myLinearGradient">
    <stop offset="0%" stop-color="red" />
    <stop offset="100%" stop-color="yellow" />
  </linearGradient>
  <radialGradient id="myRadialGradient">
    <stop offset="0%" stop-color="red" />
    <stop offset="100%" stop-color="yellow" />
  </radialGradient>
</defs>
<circle cx="250" cy="250" r="40" fill="url(#myLinearGradient)" />
<circle cx="350" cy="250" r="40" fill="url(#myRadialGradient)" />

Parents
a, defs, g, glyph, marker, mask, missing-glyph, pattern, svg, symbol

Children
animate, animateTransform, desc, metadata, set, stop, title

Attributes
class, color, color-interpolation, color-rendering, externalResourcesRequired,
gradientTransform, gradientUnits, id, spreadMethod, stop-color, stop-opacity, style,
x1, x2, xlink:actuate, xlink:arcrole, xlink:href, xlink:role, xlink:show, xlink:title,
xlink:type, xml:base, xmlns:xlink, y1, y2

marker
Defines a shape that can be used as an arrowhead or marker on a line, path, polygon, or polyline. The marker symbol is referenced using the marker, marker-start, marker-end, or marker-mid properties. The marker shape is rendered only when referenced.

Example
<defs>
  <marker id="myMarker" viewBox="0 0 10 10" refX="0" refY="5"
    markerUnits="strokeWidth" markerWidth="4" markerHeight="3" orient="auto">
    <path d="M 0 0 L 10 5 L 0 10 z" />
  </marker>
</defs>
<line x1="50" y1="100" x2="300" y2="100" stroke="black" stroke-width="8"
    fill="none" marker-end="url(#myMarker)" />
Parents
a, defs, g, glyph, marker, mask, missing-glyph, pattern, svg, symbol

Children
a, altGlyphDef, animate, animateColor, animateMotion, animateTransform, circle, clipPath, color-profile, cursor, defs, desc, ellipse, filter, font, font-face, g, image, line, linearGradient, marker, mask, metadata, path, pattern, polygon, polyline, radialGradient, rect, script, set, style, svg, switch, symbol, text, title, use, view

Attributes

mask
Defines a shape that can be used as a mask on other graphics using the mask property. A mask can use transparency.

Parents
a, defs, g, glyph, marker, mask, missing-glyph, pattern, svg, symbol

Children
a, altGlyphDef, animate, animateColor, animateMotion, animateTransform, circle, clipPath, color-profile, cursor, defs, desc, ellipse, filter, font, font-face, g, image, line, linearGradient, marker, mask, metadata, path, pattern, polygon, polyline, radialGradient, rect, script, set, style, svg, switch, symbol, text, title, use, view
Attributes
alignment-baseline, baseline-shift, class, clip, clip-path, clip-rule, color,
color-interpolation, color-interpolation-filters, color-profile, color-rendering,
cursor, direction, display, dominant-baseline, enable-background,
externalResourcesRequired, fill, fill-opacity, fill-rule, filter, flood-color,
flood-opacity, font-family, font-size, font-size-adjust, font-stretch, font-style,
font-variant, font-weight, glyph-orientation-horizontal,
glyph-orientation-vertical, height, id, image-rendering, kerning, letter-spacing,
lighting-color, marker-end, marker-mid, marker-start, mask, maskContentUnits,
maskUnits, opacity, overflow, pointer-events, requiredExtensions, requiredFeatures,
shape-rendering, stop-color, stop-opacity, stroke, stroke-dasharray,
stroke-dashoffset, stroke-linecap, stroke-linejoin, stroke-miterlimit,
stroke-opacity, stroke-width, style, systemLanguage, text-anchor, text-decoration,
text-rendering, unicode-bidi, visibility, width, word-spacing, writing-mode, x,
xml:base, xml:lang, xml:space, y

metadata
Creates a container for any metadata (such as Resource Description Framework data) from
a different namespace.

Example
<metadata>
  <md:myMetaData xmlns:md="http://example.com/ns/metadata">
    <md:summary>This is a summary of this SVG file</md:summary>
  </md:myMetaData>
</metadata>

Parents
a, animate, animateColor, animateMotion, animateTransform, circle, clipPath,
color-profile, cursor, defs, ellipse, filter, font, font-face, g, glyph, image, line,
linearGradient, marker, mask, missing-glyph, mpath, path, pattern, polygon, polyline,
radialGradient, rect, set, svg, switch, symbol, text, textPath, tref, tspan, use, view

Attributes
id, xml:base
missing-glyph
Specifies the graphics to use when a nonexistent glyph from an SVG font is used.

Example
See the font element example.

Parents
font

Children
a, altGlyphDef, animate, animateColor, animateMotion, animateTransform, circle, clipPath, color-profile, cursor, defs, desc, ellipse, filter, font, font-face, g, image, line, linearGradient, marker, mask, metadata, path, pattern, polygon, polyline, radialGradient, rect, script, set, style, svg, switch, symbol, text, title, use, view

Attributes

mpath
Used within the animateMotion element to reference (XLink href attribute) a path to use for animation.

Example
<path id="myPath" d='M100,200 C100,100 250,100 250,200 S400,300 400,200' fill="none" stroke="black" />
<rect x="23" y="45" width="40" height="40" fill="yellow">
  <animateMotion dur="6s" repeatCount="1">
<path xlink:href="#myPath" />
</animateMotion>
</rect>

Parents
animateMotion

Children
desc, metadata, title

Attributes

path
Describes a path. The d attribute contains a set of path commands (move to, quadratic and cubic Bezier curves, elliptic arcs, lines, and close curve) that, for example, can move the drawing point and draw a line or curve.

Example
<path d="M100,200 C100,100 250,100 250,200 S400,300 400,200" fill="none"
  stroke="black" />

Parents
a, clipPath, defs, g, glyph, marker, mask, missing-glyph, pattern, svg, switch, symbol

Children
animate, animateColor, animateMotion, animateTransform, desc, metadata, set, title

Attributes
class, clip-path, clip-rule, color, color-interpolation, color-rendering, cursor, d, display, externalResourcesRequired, fill, fill-opacity, fill-rule, filter, id, image-rendering, marker-end, marker-mid, marker-start, mask, onactivate, onclick, onfocusin, onfocusout, onload, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, opacity, pathLength, pointer-events, requiredExtensions, requiredFeatures, shape-rendering, stroke, stroke-dasharray, stroke-dashoffset, stroke-linecap, stroke-linejoin, stroke-miterlimit, stroke-opacity, stroke-width, style, systemLanguage, text-rendering, transform, visibility, xml:base, xml:lang, xml:space
pattern
Defines a pattern that can be used to fill or stroke other shapes. The pattern element defines a rectangular area (x, y, width, and height attributes) with a pattern (consisting of subelement shapes) that is tiled to fill or stroke the referencing shape.

Example
<defs>
  <pattern id="myPattern" x="0" y="0" width="100" height="100"
  patternUnits="userSpaceOnUse">
    <rect x="0" y="0" width="50" height="50"/>
    <rect x="50" y="50" width="50" height="50"/>
  </pattern>
</defs>
<rect x="0" y="0" width="400" height="400" fill="url(#myPattern)"
  stroke="red"/>

Parents
a, defs, g, glyph, marker, mask, missing-glyph, pattern, svg, symbol

Children
a, altGlyphDef, animate, animateColor, animateMotion, animateTransform, circle, clipPath, color-profile, cursor, defs, desc, ellipse, filter, font, font-face, g, image, line, linearGradient, marker, mask, metadata, path, pattern, polygon, polyline, radialGradient, rect, script, set, style, svg, switch, symbol, text, title, use, view

Attributes
alignment-baseline, baseline-shift, class, clip, clip-path, clip-rule, color, color-interpolation, color-interpolation-filters, color-profile, color-rendering, cursor, direction, display, dominant-baseline, enable-background, externalResourcesRequired, fill, fill-opacity, fill-rule, filter, flood-color, flood-opacity, font-family, font-size, font-size-adjust, font-stretch, font-style, font-variant, font-weight, glyph-orientation-horizontal, glyph-orientation-vertical, height, id, image-rendering, kerning, letter-spacing, lighting-color, marker-end, marker-mid, marker-start, mask, opacity, overflow, patternContentUnits, patternTransform, patternUnits, pointer-events, preserveAspectRatio, requiredExtensions, requiredFeatures, shape-rendering, stop-color, stop-opacity, stroke, stroke-dasharray, stroke-dashoffset, stroke-linecap, stroke-linejoin, stroke-miterlimit, stroke-opacity, stroke-width, style, systemLanguage, text-anchor, text-decoration, text-rendering, unicode-bidi, viewBox, visibility, width, word-spacing, writing-mode, x, xlink:actuate,
polygon
A polygon is a closed shaped made up of straight connected lines (points attribute).

Example
<polygon points="34,56 78,78 96,23" fill="blue" stroke="#FF00FF" stroke-width="15"/>
<polyline points="34,56 78,78 96,23" fill="blue" stroke="#FF00FF" stroke-width="15" transform="translate(0,100)"/>

Parents
a, clipPath, defs, g, glyph, marker, mask, missing-glyph, pattern, svg, switch, symbol

Children
animate, animateColor, animateMotion, animateTransform, desc, metadata, set, title

Attributes
class, clip-path, clip-rule, color, color-interpolation, color-rendering, cursor, display, externalResourcesRequired, fill, fill-opacity, fill-rule, filter, id, image-rendering, marker-end, marker-mid, marker-start, mask, onactivate, onclick, onfocusin, onfocusout, onload, onomousedown, onmouseover, onmouseout, onmouseup, opacity, pointer-events, points, requiredExtensions, requiredFeatures, shape-rendering, stroke, stroke-dasharray, stroke-dashoffset, stroke-linecap, stroke-linejoin, stroke-miterlimit, stroke-opacity, stroke-width, style, systemLanguage, text-rendering, transform, visibility, xml:base, xml:lang, xml:space

polyline
A polyline is a shape consisting of a set of straight connected lines (points attribute), usually not closed.

Example
<polyline points="34,56 78,78 96,23" fill="blue" stroke="#FF00FF" stroke-width="15" transform="translate(0,100)"/>
Parents
a, clipPath, defs, g, glyph, marker, mask, missing-glyph, pattern, svg, switch, symbol

Children
animate, animateColor, animateMotion, animateTransform, desc, metadata, set, title

Attributes
class, clip-path, clip-rule, color, color-interpolation, color-rendering, cursor,
display, externalResourcesRequired, fill, fill-opacity, fill-rule, filter, id,
image-rendering, marker-end, marker-mid, marker-start, mask, onactivate, onclick,
onfocusin, onfocusout, onload, onmousedown, onmousemove, onmouseout, onmouseover, onmouseout, opacity, pointer-events, points, requiredExtensions, requiredFeatures,
shape-rendering, stroke, stroke-dasharray, stroke-dashoffset, stroke-linecap,
stroke-linejoin, stroke-miterlimit, stroke-opacity, stroke-width, style,
systemLanguage, text-rendering, transform, visibility, xml:base, xml:lang, xml:space

radialGradient
Defines a radial gradient by using stop subelements that define positions and colors
within the gradient. The gradient can be modified using a gradient vector (x1, y1, x2, y2
attributes) or transformed (gradientTransform attribute). Using the XLink href attribute,
a gradient can inherit the properties of another gradient.

Example
<defs>
  <linearGradient id="myLinearGradient">
    <stop offset="0%" stop-color="red" />
    <stop offset="100%" stop-color="yellow" />
  </linearGradient>
  <radialGradient id="myRadialGradient">
    <stop offset="0%" stop-color="red" />
    <stop offset="100%" stop-color="yellow" />
  </radialGradient>
</defs>

<circle cx="250" cy="250" r="40" fill="url(#myLinearGradient)" />
<circle cx="350" cy="250" r="40" fill="url(#myRadialGradient)" />
Parents
a, defs, g, glyph, marker, mask, missing-glyph, pattern, svg, symbol

Children
animate, animateTransform, desc, metadata, set, stop, title

Attributes

rect
Defines a rectangle that has a position (x and y attributes) and size (width and height attributes). Rounded corners are created by setting the rx and ry attributes.

Example
<rect x="32" y="75" width="70" height="70" fill="#454545" />

Parents
a, clipPath, defs, g, glyph, marker, mask, missing-glyph, pattern, svg, switch, symbol

Children
animate, animateColor, animateMotion, animateTransform, desc, metadata, set, title

Attributes
class, clip-path, clip-rule, color, color-interpolation, color-rendering, cursor, display, externalResourcesRequired, fill, fill-opacity, fill-rule, filter, height, id, image-rendering, mask, onactivate, onclick, onfocusin, onfocusout, onload, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, opacity, pointer-events, requiredExtensions, requiredFeatures, rx, ry, shape-rendering, stroke, stroke-dasharray, stroke-dashoffset, stroke-linecap, stroke-linejoin, stroke-miterlimit, stroke-opacity, stroke-width, style, systemLanguage, text-rendering, transform, visibility, width, x, xml:base, xml:lang, xml:space, y
script
Denotes a script, usually ECMAScript/JavaScript. The script can be either as enclosed text or linked to an external file (XLink href attribute). This element is analogous to the script element in HTML. Placing inline scripts within a CDATA element is recommended.

Example
<script type="text/ecmascript" xlink:href="myScriptFile.es" /></script>

Parents
a, defs, g, glyph, marker, mask, missing-glyph, pattern, svg, symbol

Attributes

set
Changes any animatable attribute in the parent or referenced (XLink href attribute) element to the value specified in the to attribute at a certain time or on an event.

Example
<circle cx='350' cy='250' r='40' fill='red'>
    <set attributeName='fill' attributeType='XML' to='green' begin='1s' />
</circle>

Parents
a, circle, clipPath, defs, ellipse, feBlend, feColorMatrix, feComposite, feConvolveMatrix, feDiffuseLighting, feDisplacementMap, feDistantLight, feFlood, feFuncA, feFuncB, feFuncG, feFuncR, feGaussianBlur, feImage, feMergeNode, feMorphology, feOffset, fePointLight, feSpecularLighting, feSpotLight, feTile, feTurbulence, filter, g, glyph, image, line, linearGradient, marker, mask, missing-glyph, path, pattern, polygon, polyline, radialGradient, rect, stop, svg, switch, symbol, text, textPath, tref, tspan, use

Children
desc, metadata, title
Attributes
attributeName, attributeType, begin, dur, end, externalResourcesRequired, fill, id,
max, min, onbegin, onend, onrepeat, repeatCount, repeatDur, requiredExtensions,
requiredFeatures, restart, systemLanguage, to, xlink:actuate, xlink:arcrole,
xlink:href, xlink:role, xlink:show, xlink:title, xlink:type, xml:base, xmlns:xlink

stop
Used within linearGradient or radialGradient elements. Defines a position (offset
attribute) and color (stop-color and stop-opacity attributes) within the gradient.

Example
<defs>
  <linearGradient id="myLinearGradient">
    <stop offset="0%" stop-color="red" />
    <stop offset="100%" stop-color="yellow" />
  </linearGradient>
  <radialGradient id="myRadialGradient">
    <stop offset="0%" stop-color="red" />
    <stop offset="100%" stop-color="yellow" />
  </radialGradient>
</defs>

  <circle cx='250' cy='250' r='40' fill="url(#myLinearGradient)" />
  <circle cx='350' cy='250' r='40' fill="url(#myRadialGradient)" />

Parents
linearGradient, radialGradient

Children
animate, animateColor, set

Attributes
class, color, color-interpolation, color-rendering, id, offset, stop-color,
stop-opacity, style, xml:base
**style**
Defines an internal cascading style sheet (CSS). The `style` element is analogous to the HTML `style` element. Placing the style definitions with a `CDATA` element is recommended.

**Example**
```xml
<style type="text/css"><![CDATA[
circle {
    fill: red;
}
circle.stroked{
    stroke: green;
    stroke-width:8px;
}
]]></style>
```

```xml
<circle cx="250" cy="250" r="40" />
<circle cx="350" cy="250" r="40" class="stroked" />
```

**Parents**
a, defs, g, glyph, marker, mask, missing-glyph, pattern, svg, symbol

**Attributes**
id, media, title, type, xml:base, xml:space

**svg**
Defines an SVG document or document fragment. When a document is defined, the correct SVG namespace has to be declared on the element using the `xmlns` attribute. An internal `svg` element (defining a document fragment) can be used to set a new viewport for its enclosed elements.

**Example**
```xml
<svg width="100" height="100" viewBox="0 0 100 100" xmlns="http://www.w3.org/2000/svg">
    <polygon points="34,56 78,78 96,23" fill="none" stroke="#FF00FF"/>
</svg>
```

**Parents**
a, defs, g, glyph, marker, mask, missing-glyph, pattern, svg, switch, symbol
Children
a, altGlyphDef, animate, animateColor, animateMotion, animateTransform, circle, clipPath, color-profile, cursor, defs, desc, ellipse, filter, font, font-face, g, image, line, linearGradient, marker, mask, metadata, path, pattern, polygon, polyline, radialGradient, rect, script, set, style, svg, switch, symbol, text, title, use, view

Attributes
alignment-baseline, baseline-shift, class, clip, clip-path, clip-rule, color, color-interpolation, color-interpolation-filters, color-profile, color-rendering, contentScriptType, contentStyleType, cursor, direction, display, dominant-baseline, enable-background, externalResourcesRequired, fill, fill-opacity, fill-rule, filter, flood-color, flood-opacity, font-family, font-size, font-size-adjust, font-stretch, font-style, font-variant, font-weight, glyph-orientation-horizontal, glyph-orientation-vertical, height, id, image-rendering, kerning, letter-spacing, lighting-color, marker-end, marker-mid, marker-start, mask, onabort, onactivate, onclick, onerror, onfocusin, onfocusout, onload, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, onresize, onscroll, onunload, opacity, overflow, pointer-events, preserveAspectRatio, requiredExtensions, requiredFeatures, shape-rendering, stop-color, stop-opacity, stroke, stroke-dasharray, stroke-dashoffset, stroke-linecap, stroke-linejoin, stroke-miterlimit, stroke-opacity, stroke-width, style, systemLanguage, text-anchor, text-decoration, text-rendering, unicode-bidi, version, viewBox, visibility, width, word-spacing, writing-mode, x, xml:base, xml:lang, xml:space, xmlns, y, zoomAndPan

switch
Provides the user agent with alternative data if a certain fragment (usually a foreignObject) cannot be rendered (based on the requiredFeatures and requiredExtensions attributes). It can also be used to provide alternative text based on the user's language (systemLanguage attribute).

Example
<s>switch>
   <foreignObject x="45" y="70" width="200" height="50"
   requiredExtensions="http://domain.com/SVGExtensions/MathML">
      <math xmlns="http://www.w3.org/1998/Math/MathML">
         <apply>
            <minus/>
            <ci>a</ci>
         </apply>
         <apply>
            <plus/>
            <ci>b</ci>
         </apply>
      </math>
   </foreignObject>
</s>
Parents
a, defs, g, glyph, marker, mask, missing-glyph, pattern, svg, switch, symbol

Children
a, animate, animateColor, animateMotion, animateTransform, circle, desc, ellipse, foreignObject, g, image, line, metadata, path, polygon, polyline, rect, set, svg, switch, text, title, use

Attributes

symbol
Defines a symbol that is not rendered directly but can be referenced using a use element. A symbol can fit into the viewport defined by the use element using the viewBox and preserveAspectRatio attributes.
Example

<defs>
  <symbol id="mySymbol" viewBox="0 0 20 20">
    <line x1="5" y1="5" x2="15" y2="15" fill="none" stroke="red"
         stroke-width="1"/>
    <line x1="15" y1="5" x2="5" y2="15" fill="none" stroke="red"
         stroke-width="1"/>
    <rect x="2" y="2" width="16" height="16" rx="3" fill="none"
         stroke="red" stroke-width="2"/>
  </symbol>
</defs>

<use x="5" y="5" width="100" height="100" xlink:href="#mySymbol"/>

Parents
a, defs, g, glyph, marker, mask, missing-glyph, pattern, svg, symbol

Children
a, altGlyphDef, animate, animateColor, animateMotion, animateTransform, circle,
clipPath, color-profile, cursor, defs, desc, ellipse, filter, font, font-face, g, image,
line, linearGradient, marker, mask, metadata, path, pattern, polygon, polyline,
radialGradient, rect, script, set, style, svg, switch, symbol, text, title, use, view

Attributes
alignment-baseline, baseline-shift, class, clip, clip-path, clip-rule, color,
color-interpolation, color-interpolation-filters, color-profile, color-rendering,
cursor, direction, display, dominant-baseline, enable-background,
externalResourcesRequired, fill, fill-opacity, fill-rule, filter, flood-color,
flood-opacity, font-family, font-size, font-size-adjust, font-stretch, font-style,
font-variant, font-weight, glyph-orientation-horizontal,
glyph-orientation-vertical, id, image-rendering, kerning, letter-spacing,
lighting-color, marker-end, marker-mid, marker-start, mask, onactivate, onclick,
onfocusin, onfocusout, onload, onmouseenter, onmousemove, onmouseout, onmouseup,
onmousemove, opacity, overflow, pointer-events, preserveAspectRatio, shape-rendering,
stop-color, stop-opacity, stroke, stroke-dasharray, stroke-dashoffset,
stroke-linecap, stroke-linejoin, stroke-miterlimit, stroke-opacity, stroke-width,
style, text-anchor, text-decoration, text-rendering, unicode-bidi, viewBox,
visibility, word-spacing, writing-mode, xml:base, xml:lang, xml:space
text
The graphical element for displaying text. The text can be formatted using several attributes and cascading style sheets (CSS). To create multiline text or inline formatting, use the tspan element.

Example
<text x="34" y="78" font-family="Arial" font-size="12" >This is a simple text with a <tspan fill="red">red</tspan> word</text>

Parents
a, clipPath, defs, g, glyph, marker, mask, missing-glyph, pattern, svg, switch, symbol

Children
a, altGlyph, animate, animateColor, animateMotion, animateTransform, desc, metadata, set, textPath, title, tref, tspan

Attributes
alignment-baseline, baseline-shift, class, clip-path, clip-rule, color, color-interpolation, color-rendering, cursor, direction, display, dominant-baseline, dx, dy, externalResourcesRequired, fill, fill-opacity, fill-rule, filter, font-family, font-size, font-size-adjust, font-stretch, font-style, font-variant, font-weight, glyph-orientation-horizontal, glyph-orientation-vertical, id, image-rendering, kerning, lengthAdjust, letter-spacing, mask, onactivate, onclick, onfocusin, onfocusout, onload, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, opacity, pointer-events, requiredExtensions, requiredFeatures, rotate, shape-rendering, stroke, stroke-dasharray, stroke-dashoffset, stroke-linecap, stroke-linejoin, stroke-miterlimit, stroke-opacity, stroke-width, style, systemLanguage, text-anchor, text-decoration, text-rendering, textLength, transform, unicode-bidi, visibility, word-spacing, writing-mode, x, xml:base, xml:lang, xml:space, y

textPath
The text content of the textPath element is rendered to follow a referenced path (XLink href attribute). The behavior can be modified using the method, spacing, and startOffset attributes.
Example
<defs>
  <path id="myPath" d="M100,200 C100,100 250,100 250,200 S400,300 400,200"/>
</defs>

<text x="34" y="208" font-family="Arial" font-size="14" >
  <textPath xlink:href="#myPath">This text is following a path</textPath>
</text>

Parents
text

Children
a, altGlyph, animate, animateColor, desc, metadata, set, title, tref, tspan

Attributes
alignment-baseline, baseline-shift, class, clip-path, clip-rule, color,
color-interpolation, color-rendering, cursor, direction, display, dominant-baseline,
externalResourcesRequired, fill, fill-opacity, fill-rule, filter, font-family,
font-size, font-size-adjust, font-stretch, font-style, font-variant, font-weight,
glyph-orientation-horizontal, glyph-orientation-vertical, id, image-rendering,
kerning, lengthAdjust, letter-spacing, mask, method, onactivate, onclick, onfocusin,
onfocusout, onload, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup,
opacity, pointer-events, requiredExtensions, requiredFeatures, shape-rendering,
spacing, startOffset, stroke, stroke-dasharray, stroke-dashoffset, stroke-linecap,
stroke-linejoin, stroke-miterlimit, stroke-opacity, stroke-width, style,
systemLanguage, text-anchor, text-decoration, text-rendering, textLength,
unicode-bidi, visibility, word-spacing, xlink:actuate, xlink:arcrole, xlink:href,
xlink:role, xlink:show, xlink:title, xlink:type, xml:base, xml:lang, xml:space,
xmlns:xlink

title
Specifies a text title for an element. Authors are recommended to include a title element as the first child node of the outermost svg element (compare to the HTML title element). The user agent can use the title for a window caption or tooltip when the user mouses over the content. The title element can also contain other XML data under a separate namespace.
Example
<svg xmlns="http://www.w3.org/2000/svg" version="1.1" width="100%" height="100%">
  <polygon points="23,45 78,98 234,54" fill="yellow">
    <title>A yellow polygon</title>
    <desc>This is a polygon primitive with three points, filled with yellow color</desc>
  </polygon>
</svg>

Parents
a, animate, animateColor, animateMotion, animateTransform, circle, clipPath, color-profile, cursor, defs, ellipse, filter, font, font-face, g, glyph, image, line, linearGradient, marker, mask, missing-glyph, mpath, path, pattern, polygon, polyline, radialGradient, rect, set, svg, switch, symbol, text, textPath, tref, tspan, use, view

Attributes
class, content, id, style, xml:base, xml:lang, xml:space

tref
Works like the tspan element but does not contain any text data in itself; instead, the tref element references another element containing text.

Example
<text id="myText" x="34" y="78">
  This a text
</text>

<text x="34" y="108" font-family="Arial" font-size="14">
  And here it is again: <tref xlink:href="#myText" />
</text>

Parents
text, textPath, tspan

Children
animate, animateColor, desc, metadata, set, title

Attributes
alignment-baseline, baseline-shift, class, clip-path, clip-rule, color, color-interpolation, color-rendering, cursor, direction, display, dominant-baseline, dx, dy, externalResourcesRequired, fill, fill-opacity, fill-rule, filter, font-family,
tspan
Defines an inline text element, analogous to the span element in HTML. tspan can be used to format or create multiline texts.

Example

<text x="34" y="78" font-family="Arial" font-size="12">This is a simple text with a <tspan fill="red">red</tspan> word</text>

Parents
text, textPath, tspan

Children
a, altGlyph, animate, animateColor, desc, metadata, set, title, tref, tspan

Attributes
alignment-baseline, baseline-shift, class, clip-path, clip-rule, color,
color-interpolation, color-rendering, cursor, direction, display, dominant-baseline,
dx, dy, externalResourcesRequired, fill, fill-opacity, fill-rule, filter, font-family,
font-size, font-size-adjust, font-stretch, font-style, font-variant, font-weight,
glyph-orientation-horizontal, glyph-orientation-vertical, id, image-rendering,
kerning, lengthAdjust, letter-spacing, mask, onactivate, onclick, onfocusin,
onfocusout, onload, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup,
opacity, pointer-events, requiredExtensions, requiredFeatures, rotate,
shape-rendering, stroke, stroke-dasharray, stroke-dashoffset, stroke-linecap,
stroke-linejoin, stroke-miterlimit, stroke-opacity, stroke-width, style,
systemLanguage, text-anchor, text-decoration, text-rendering, textLength,
unicode-bidi, visibility, word-spacing, x, xlink:actuate, xlink:arcrole, xlink:href,
xlink:role, xlink:show, xlink:title, xlink:type, xml:base, xml:lang, xml:space,
xmlns:xlink, y
**use**

Most elements can be reused using the `use` element. The `use` element references a different graphical, `g`, `svg`, `symbol`, or `use` element using the XLink `href` attribute. The referenced element is rendered at the position (`x` and `y` attributes) and size (`width` and `height` attributes) defined in the `use` element.

**Example**

```xml
<defs>
  <rect id="myRect" x="56" y="56" width="15" height="48" fill="#FFF000" />
</defs>

<use x="20" y="35" xlink:href="#myRect" />
```

**Parents**

`a`, `clipPath`, `defs`, `g`, `glyph`, `marker`, `mask`, `missing-glyph`, `pattern`, `svg`, `switch`, `symbol`

**Children**

`animate`, `animateColor`, `animateMotion`, `animateTransform`, `desc`, `metadata`, `set`, `title`

**Attributes**

view
Defines a viewport in the SVG file. A view can be used like a named anchor in HTML.

Example
<view id="myView" viewBox="100 200 400 200" />

<a xlink:href="#myView">
  <ellipse cx="300" cy="300" rx="200" ry="100" stroke="red" fill="blue" />
</a>

Parents
a, defs, g, glyph, marker, mask, missing-glyph, pattern, svg, symbol

Children
desc, metadata, title

Attributes
externalResourcesRequired, id, preserveAspectRatio, viewBox, viewTarget, xml:base, zoomAndPan

vkern
Specifies a kerning value (k attribute) for a pair of Unicode character sets (u1 and u2 attributes) or glyphs (g1 and g2 attributes) for vertical text.

Parents
font

Attributes
g1, g2, id, k, u1, u2, xml:base