AN APPLIED MATERIALS COMPANY

NEMA| GFX Extensions

TSVG Supported Elements List

Version v22.03

Part Number: D-TSVG-PVG

June 25, 2025

Disclaimer

This document is written in good faith with the intent to assist the readers in the use of the product. Circuit diagrams and other information relating to Think Silicon S.A products are included as a means of illustrating typical applications. Although the information has been checked and is believed to be accurate, no responsibility is assumed for inaccuracies. Information contained in this document is subject to continuous improvements and developments.

Think Silicon S.A products are not designed, intended, authorized or warranted for use in any life support or other application where product failure could cause or contribute to personal injury or severe property damage. Any and all such uses without prior written approval of Think Silicon S.A. will be fully at the risk of the customer.

Think Silicon S.A. disclaims and excludes any and all warranties, including without limitation any and all implied warranties of merchantability, fitness for a particular purpose, title, and infringement and the like, and any and all warranties arising from any course or dealing or usage of trade.

This document may not be copied, reproduced, or transmitted to others in any manner. Nor may any use of information in this document be made, except for the specific purposes for which it is transmitted to the recipient, without the prior written consent of Think Silicon S.A. This specification is subject to change at anytime without notice.

Think Silicon S.A. is not responsible for any errors contained herein. In no event shall Think Silicon S.A. be liable for any direct, indirect, incidental, special, punitive, or consequential damages; or for loss of data, profits, savings or revenues of any kind; regardless of the form of action, whether based on contract; tort; negligence of Think Silicon S.A or others; strict liability; breach of warranty; or otherwise; whether or not any remedy of buyers is held to have failed of its essential purpose, and whether or not Think Silicon S.A. has been advised of the possibility of such damages.

Questions or comments may be directed to:
Think Silicon S.A
Suite B8
Patras Science Park
Rion Achaias 26504, Greece
web: http://www.think-silicon.com
email:info@think-silicon.com
Tel: +30 2610 911543
Fax: +30 2610 911544

Contents

1 Overview

This document contains the supported elements from Tiny SVG 1.2 profile, as well as descriptions of the elements and the their respective attributes. These elements are used in files converted from SVG to TSVG format, for use with the NEMA| GFX library.

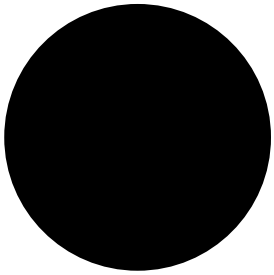
2 Supported elements

2.1 `circle` element

The `circle` SVG element is a basic shape, used to draw circles on a center point and a radius.

Example

```
<svg viewBox="0 0 100 100" xmlns="http://www.w3.org/2000/svg">
  <circle cx="50" cy="50" r="50"/>
</svg>
```



Supported attributes

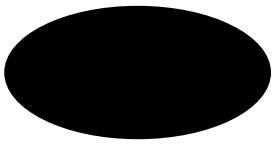
- cx
- cy
- r
- id
- fill
- fill-opacity
- fill-rule
- stroke
- stroke-opacity
- stroke-width
- stroke-dasharray
- stroke-dashoffset
- style
- transform

2.2 `ellipse` element

The `ellipse` SVG element is a basic shape, used to draw ellipses on a center coordinate, and both their x and y radius

Example

```
<svg viewBox="0 0 200 100" xmlns="http://www.w3.org/2000/svg">
  <ellipse cx="100" cy="50" rx="100" ry="50"/>
</svg>
```



Supported attributes

- cx
- cy
- rx
- ry
- id
- fill
- fill-opacity
- fill-rule
- stroke
- stroke-opacity
- stroke-width
- stroke-dasharray
- stroke-dashoffset
- style
- transform

2.3 g element

The g SVG element is a container used to group other SVG elements.

Transformations applied to the g element are performed on its child elements, and its attributes are inherited by its children.

Supported attributes

- id
- fill

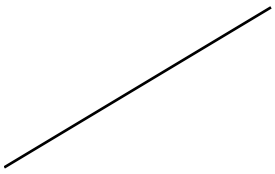
- fill-opacity
- fill-rule
- stroke
- stroke-opacity
- stroke-width
- stroke-linecap
- stroke-linejoin
- stroke-miterlimit
- stroke-dasharray
- stroke-dashoffset
- style
- transform

2.4 `line` element

The `line` element is an SVG basic shape used to create a line connecting two points.

Example

```
<svg viewBox="0 0 100 100" xmlns="http://www.w3.org/2000/svg">
  <line x1="0" y1="80" x2="100" y2="20" stroke="black"/>
</svg>
```



Supported attributes

- x1
- y1
- x2
- y2
- id
- fill
- fill-opacity
- fill-rule
- stroke

- stroke-opacity
- stroke-width
- stroke-linecap
- stroke-linejoin
- stroke-miterlimit
- stroke-dasharray
- stroke-dashoffset
- style
- transform

2.5 `linearGradient` element

The `linearGradient` element lets authors define linear gradients to apply to other SVG elements.

Supported attributes

- x1
- y1
- x2
- y2
- gradientUnits
- id

2.6 `path` element

The `path` SVG element is the generic element to define a shape. All the basic shapes can be created with a path element.

Example

```
<svg viewBox="0 0 100 100" xmlns="http://www.w3.org/2000/svg">
  <path d="M 10,30
           A 20,20 0,0,1 50,30
           A 20,20 0,0,1 90,30
           Q 90,60 50,90
           Q 10,60 10,30 z"/>
</svg>
```

Supported attributes

- d
- class
- id
- fill
- fill-opacity
- fill-rule
- stroke
- stroke-opacity
- stroke-width
- stroke-linecap
- stroke-linejoin
- stroke-miterlimit
- stroke-dasharray
- stroke-dashoffset
- style
- transform

2.7 `polygon` element

The `polygon` element defines a closed shape consisting of a set of connected straight line segments. The last point is connected to the first point.

Example

```
<svg viewBox="0 0 200 100" xmlns="http://www.w3.org/2000/svg">
  <polygon points="100,100 150,25 150,75 200,0"
             fill="none" stroke="black" />
</svg>
```

Supported attributes

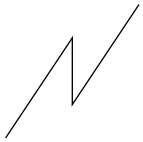
- points
- id
- fill
- fill-opacity
- fill-rule
- stroke
- stroke-opacity
- stroke-width
- stroke-linecap
- stroke-linejoin
- stroke-miterlimit
- stroke-dasharray
- stroke-dashoffset
- style
- transform

2.8 `polyline` element

The `polyline` element is an SVG basic shape that creates straight lines connecting several points. Typically a `polyline` is used to create open shapes as the last point doesn't have to be connected to the first point.

Example

```
<svg viewBox="0 0 200 100" xmlns="http://www.w3.org/2000/svg">
  <polyline points="100,100 150,25 150,75 200,0"
            fill="none" stroke="black" />
</svg>
```



Supported attributes

- points
- id
- fill
- fill-opacity
- fill-rule
- stroke
- stroke-opacity
- stroke-width
- stroke-linecap
- stroke-linejoin
- stroke-miterlimit
- stroke-dasharray
- stroke-dashoffset
- style
- transform

2.9 `radialGradient` element

The `radialGradient` element lets authors define radial gradients to apply to other SVG elements.

Supported attributes

- cx
- cy
- r
- gradientUnits
- id

2.10 `stop` element

The `stop` element defines a color and its position to use on a gradient. This element is always a child of a `linearGradient` or `radialGradient` element.

Supported attributes

- stop-color
- stop-opacity
- id

2.11 `rect` element

The `rect` element is a basic SVG shape that draws rectangles, defined by their position, width, and height. The rectangles may have their corners rounded.

Example

```
<svg viewBox="0 0 220 100" xmlns="http://www.w3.org/2000/svg">
  <!-- Simple rectangle -->
  <rect width="100" height="100" />

  <!-- Rounded corner rectangle -->
  <rect x="120" width="100" height="100" rx="15" />
</svg>
```

Supported attributes

- x
- y
- rx
- ry
- height
- width
- id
- fill
- fill-opacity
- fill-rule
- stroke
- stroke-opacity
- stroke-width

- stroke-dasharray
- stroke-dashoffset
- style
- transform

2.12 `text` element

The `text` element draws a graphics element consisting of text.

Supported attributes

- x
- y
- id
- fill
- fill-opacity
- stroke
- stroke-opacity
- stroke-width
- transform

2.13 `use` element

The `use` element takes nodes from within the SVG document, and duplicates them somewhere else.

Supported attributes

- x
- y
- id
- xlink:href
- href
- fill
- fill-opacity
- fill-rule
- stroke
- stroke-opacity
- stroke-width
- stroke-linecap
- stroke-linejoin
- stroke-miterlimit
- stroke-dasharray
- stroke-dashoffset

- transform

3 Supported attributes

3.1 cx

For `circle/ellipse` elements

Description	The x-axis coordinate of a circle or ellipse.
Value type	length
Default value	0

For `radialGradient` element

Description	The x coordinate of the end circle of the radial gradient.
Value type	length-percent
Default value	50%

3.2 cy

For `circle/ellipse` elements

Description	The y-axis coordinate of a circle or ellipse.
Value type	length
Default value	0

For `radialGradient` element

Description	The y coordinate of the end circle of the radial gradient.
Value type	length-percentage
Default value	50%

3.3 x1

For `line` element

Description	Defines the x-axis coordinate of the line starting point.
Value type	length | number

Default value	0

For `linearGradient` element

Description	Defines the x coordinate of the starting point of the vector gradient along which the linear gradient is drawn.
Value type	length-percentage | number
Default value	0%

3.4 y1

For `line` element

Description	Defines the y-axis coordinate of the line starting point.
Value type	length | number
Default value	0

For `linearGradient` element

Description	Defines the y coordinate of the starting point of the vector gradient along which the linear gradient is drawn.
Value type	length-percentage | number
Default value	0%

3.5 x2

For `line` element

Description	Defines the x-axis coordinate of the line ending point.
Value type	length | number
Default value	0

For `linearGradient` element

Description	Defines the x coordinate of the ending point of the vector gradient along which the linear gradient is drawn.
Value type	length-percentage | number
Default value	100%

3.6 y2

For `line` element

Description	Defines the y-axis coordinate of the line ending point.
Value type	length | number
Default value	0

For `linearGradient` element

Description	Defines the y coordinate of the ending point of the vector gradient along which the linear gradient is drawn.
Value type	length-percentage | number
Default value	100%

3.7 x

Description	Defines the x coordinate of a rect, of a use element or of the starting point of a text baseline.
Value type	length | coordinate
Default value	0

3.8 y

Description	Defines the y coordinate of a rect, of a use element or of the starting point of a text baseline.
Value type	length | coordinate

Default value	0

3.9 `r`

For `circle` element

Description	The radius of the circle. A value lower or equal to zero disables rendering of the circle.
Value type	length
Default value	0

For `radialGradient` element

Description	The radius of the end circle of the radial gradient. The gradient will be drawn such that the 100% `stop` is mapped to the perimeter of the end circle.
Value type	length
Default value	50%

3.10 `rx`

For `ellipse` element

Description	The radius of the ellipse on the x axis.
Value type	`auto` | length
Default value	`auto`

For `rect` element

Description	The horizontal corner radius of the rect. Defaults to `ry` if it is specified.
Value type	`auto` | length
Default value	`auto`

3.11 `ry`

For `ellipse` element

Description	The radius of the ellipse on the y axis.
Value type	`auto` | length
Default value	`auto`

For `rect` element

Description	The vertical corner radius of the rect. Defaults to `rx` if it is specified.
Value type	`auto` | length
Default value	`auto`

3.12 `height`

Description	The height of the rect.
Value type	`auto` | length | percentage
Default value	`auto`

3.13 `width`

Description	The width of the rect.
Value type	`auto` | length | percentage
Default value	`auto`

3.14 `d`

Description	This attribute defines the shape of the path.
Value type	string
Default value	''

3.15 `points`

Description	This attribute defines the list of points (pairs of x,y absolute coordinates) required to draw a polygon or a polyline.
Value type	number+
Default value	""

3.16 `class`

Description	Assigns a class name or set of class names to an element. You may assign the same class name or names to any number of elements, however, multiple class names must be separated by whitespace characters.
Value type	list-of-class-names
Default value	''

3.17 `id`

Description	This attribute assigns a unique name to an element.
Value type	id
Default value	None

3.18 `href`

Description	This attribute defines a link to a resource as a reference URL. The exact meaning of that link depends on the context of each element using it.
Value type	url
Default value	None

3.19 `xlink:href`

Description	This attribute defines a reference to a resource as a reference IRI. The exact meaning of that link depends on the context of each element using it.
Value type	iri
Default value	None

3.20 `fill`

Description	For shapes and text, `fill` is a presentation attribute that defines the color used to paint the element.
Value type	Available color formats are: • rgb/rgba format • #RGB, #RRGGBB, #RGBA, #RRGGBBAA format • predefined color tags included here
Default value	black

3.21 `fill-opacity`

Description	This attribute is a presentation attribute defining the opacity of the paint server (color, gradient, pattern, etc.) applied to a shape.
Value type	[0-1]
Default value	1

3.22 `fill-rule`

Description	This attribute is a presentation attribute defining the algorithm to use to determine the inside part of a shape.
Value type	nonzero | evenodd
Default value	nonzero

3.23 `gradientUnits`

The `gradientUnits` attribute defines the coordinate system used for attributes specified on the gradient elements.

For `linearGradient` element

Description	For linearGradient element, `gradientUnits` defines the coordinate system used for the attributes x1, y1, x2, and y2.
Value type	`userSpaceOnUse` | `objectBoundingBox`
Default value	`objectBoundingBox`

For `radialGradient` element

Description	For radialGradient element, `gradientUnits` defines the coordinate system used for the attributes cx, cy, and r.
Value type	`userSpaceOnUse` | `objectBoundingBox`
Default value	`objectBoundingBox`

3.24 `stop-color`

Description	This attribute defines the color of the gradient stop.
Value type	`currentcolor`|color
Default value	black

3.25 `stop-opacity`

Description	This attribute defines the opacity of the gradient stop.
Value type	opacity
Default value	1

3.26 `stroke`

Description	This attribute is a presentation attribute defining the color (or any SVG paint server like gradients or patterns) used to paint the outline of the shape.
Value type	Available color formats are: • rgb/rgba format • #RGB, #RRGGBB, #RGBA, #RRGGBBAA format • predefined color tags included here
Default value	none

3.27 `stroke-opacity`

Description	This attribute is a presentation attribute defining the opacity of the paint server (color, gradient, pattern, etc.) applied to the stroke of a shape.
Value type	[0-1]
Default value	1

3.28 `stroke-width`

Description	This attribute is a presentation attribute defining the width of the stroke to be applied to the shape.
Value type	length
Default value	1px

3.29 `stroke-linecap`

Description	This attribute is a presentation attribute defining the shape to be used at the end of open subpaths when they are stroked.
Value type	`butt` | `round` | `square`
Default value	butt

3.30 `stroke-linejoin`

Description	This attribute is a presentation attribute defining the shape to be used at the join of two segments of a stroked path.
Value type	`bevel` | `miter` | `round`
Default value	miter

3.31 `stroke-miterlimit`

Description	This attribute is a presentation attribute defining a limit in the ratio of the miter length to the stroke-width used to fraw a miter join. When the miter is exceeded the join is substituted with a bevel join.
Value type	`number`
Default value	4

3.32 `stroke-dasharray`

Description	This attribute is a presentation attribute defining the on off patterns used at the shape outline.
Value type	none | `dasharray`
Default value	none

3.33 `stroke-dashoffset`

Description	This attribute is a presentation attribute defining an initial offset at the associated dash array.
Value type	`length`
Default value	0

3.34 `style`

Description	This attribute allows to style an element using CSS declarations.

Value type	CSS declarations
Available declarations	All supported SVG presentation attributes included in this document can be used as CSS declarations inside the style attribute.

3.35 `transform`

Description	This attribute defines a list of transform definitions
Value type	id
Default value	None

Transform functions

matrix
The `matrix(a,b,c,d,e,f)` transform function specifies a transformation in the form of a transformation matrix of six values.

translate
The `translate(x,[y])` transform function moves the object by x and y. If y is not provided, it is assumed to be 0.

scale
The `scale(x,[y])` transform function specifies a scale operation by x and y. If y is not provided, it is assumed to be equal to x.

rotate
The `rotate(a,[x,y])` transform function specifies a rotation by a degrees about a given point. If optional parameters x and y are not supplied, the rotation is about the origin of the current user coordinate system. If the optional parameters are supplied, the rotation is about the point (x,y).

skewX
The `skewX(a)` transform function specifies a skew transformation along the x axis by a degrees.

skewY
The `skewY(a)` function specifies a skew transformation along the y axis by a degrees.