

svg_options — Options for exporting to Scalable Vector Graphics

[Description](#)[Syntax](#)[Options](#)[Remarks and examples](#)[Also see](#)

Description

These *svg_options* are used with `graph export` when creating a Scalable Vector Graphics file; see [\[G-2\] graph export](#).

Syntax

<i>svg_options</i>	Description
<code>baselineshift(on off)</code>	whether to use SVG <code>baseline-shift</code> attribute for subscript or superscript; default is <code>off</code>
<code>ignorefont(on off)</code>	whether to ignore graph fonts used for text; default is <code>off</code>
<code>bgfill(on off)</code>	whether to use background fill; default is <code>on</code>
<code>nbspace(on off)</code>	whether to use Unicode character for no-break space instead of spaces in some strings; default is <code>on</code>
<code>clipstroke(on off)</code>	whether to use clipping paths to simulate stroke alignment; default is <code>on</code>
<code>scalestrokewidth(on off)</code>	whether to manually scale stroke widths; default is <code>off</code>
<code>verbose</code>	whether to output all default attributes and classes
<code>pathprefix(prefix)</code>	prefix to use when naming SVG paths
<code>width(#px #in)</code>	width of graph in pixels or inches
<code>height(#px #in)</code>	height of graph in pixels or inches
<code>fontface(fontname)</code>	default font to use
<code>fontfacesans(fontname)</code>	font to use for text in <code>{stSans}</code> “font”
<code>fontfaceserif(fontname)</code>	font to use for text in <code>{stSerif}</code> “font”
<code>fontfacemono(fontname)</code>	font to use for text in <code>{stMono}</code> “font”
<code>fontfacesymbol(fontname)</code>	font to use for text in <code>{stSymbol}</code> “font”

where *fontname* may be a valid font name or `default` to restore the default setting.

Current default values may be listed by typing

```
. graph set svg
```

and default values may be set by typing

```
. graph set svg name value
```

where *name* is the name of an *svg_option*, omitting the parentheses.

Options

`baselineshift(on)` and `baselineshift(off)` specify whether to use the SVG attribute `baseline-shift` for displaying subscripts and superscripts. As of February 4, 2016, IE, Microsoft Edge, and Firefox do not support this attribute, but Chrome and Safari do. When `baselineshift(off)` is specified, Stata instead uses the SVG `dy` attribute to display subscripts and superscripts. However, Chrome and Safari may not properly render subscripts and superscripts when using the `dy` attribute if there are leading or trailing spaces in the string and the `nbsp(off)` option is specified. The default is `baselineshift(off)`.

`ignorefont(on)` and `ignorefont(off)` specify whether to output the SVG `font-family` attribute. When `ignorefont(on)` is specified, no font information is written to the SVG file, so the SVG renderer must determine which font to use when displaying text. The default is `ignorefont(off)`.

`bgfill(on)` and `bgfill(off)` specify whether to use the background fill. When `bgfill(off)` is specified, no background fill is written to the SVG file, so the SVG background is transparent. The default is `bgfill(on)`.

`nbsp(on)` and `nbsp(off)` specify whether to use the Unicode character for no-break space (U+00A0) in place of spaces in a string. By default, SVG renderers ignore leading and trailing spaces in strings. The SVG `xml:space` attribute can be used to preserve leading and trailing spaces in strings, but both IE and Microsoft Edge ignore that attribute. When `nbsp(on)` is specified, Stata first looks at a string to see whether it has any leading or trailing spaces. If so, it replaces all the spaces in the string with the Unicode no-break space character. When `nbsp(off)` is specified, Stata uses the `xml:space` attribute when a string contains leading or trailing spaces. `nbsp()` is a tradeoff between making the XML data within the SVG file more readable and the SVG file itself more compact (`off`) versus greater compatibility among web browsers (`on`). The default is `nbsp(on)`.

`clipstroke(on)` and `clipstroke(off)` specify whether to use clipping paths to simulate stroke alignment for polygons. Stata allows a closed object such as an ellipse, a rectangle, or a polygon to be stroked on the inside, center, or outside of the object's outline. Stata uses the SVG `stroke-alignment` property to support stroke alignment for polygons. But popular web browsers such as Internet Explorer, Apple Safari, and Google Chrome do not support the `stroke-alignment` property (as of March 15, 2016). When `clipstroke(on)` is specified, Stata simulates an inside or outside stroke alignment for polygons by first defining the polygon as its own clipping path, then stroking the center of the polygon's outline with twice the stroke width. The half of the doubled stroke that is not contained within the clipping path will not be visible. The default is `clipstroke(on)`.

`scalestrokewidth(on)` and `scalestrokewidth(off)` specify whether to manually scale the stroke widths in an SVG file. Most SVG renderers use the SVG file's viewbox to scale shapes and stroke widths. While Adobe Illustrator does scale shapes according to an SVG file's viewbox, it fails to do so for the stroke widths, which causes the strokes to appear extremely thick. When `scalestrokewidth(on)` is specified, Stata manually scales every stroke width so that SVG files appear correctly in Adobe Illustrator but incorrectly in every other known SVG renderer. The default is `scalestrokewidth(off)`.

`verbose` specifies whether to output all default attributes and classes. For example, the default text alignment value for the SVG attribute `text-anchor` is `start`. Stata only outputs the `text-anchor` attribute if the text alignment is centered or end aligned. When `verbose` is specified, Stata will always output the `text-anchor` attribute even if the text uses the default start text alignment. Stata will also output attributes such as the writing direction, the type of point cloud if applicable, and the type of marker symbol being displayed.

`pathprefix(prefix)` specifies the prefix to use when naming SVG paths. A path is a collection of lines and curves that define a shape. A path in an SVG image can be named so that it can be used multiple times in the image. However, if multiple SVG images are included in an HTML document and they share common path names, the browser can become confused about which path to use. To prevent this path name collision, Stata uses random strings to create unique path names. If you would prefer to create stable path names, specify `pathprefix()`. Stata will use your path prefix and an index that is incremented for each path to create stable path names. Use the same guidelines as naming variables when specifying a path prefix.

`width(#px|#in)` specifies the width of the graph in pixels or inches. The default units are pixels if no units are specified. If the width is specified but not the height, Stata determines the appropriate height from the graph's aspect ratio.

`height(#px|#in)` specifies the height of the graph in pixels or inches. The default units are pixels if no units are specified. If the height is specified but not the width, Stata determines the appropriate width from the graph's aspect ratio.

`fontface(fontname)` specifies the name of the font to be used to render text for which no other font has been specified. The default is *Helvetica*, which may be restored by specifying *fontname* as `default`. If *fontname* contains spaces, it must be enclosed in double quotes.

`fontfacesans(fontname)` specifies the name of the font to be used to render text for which the `{stSans}` “font” has been specified. The default is *Helvetica*, which may be restored by specifying *fontname* as `default`. If *fontname* contains spaces, it must be enclosed in double quotes.

`fontfaceserif(fontname)` specifies the name of the font to be used to render text for which the `{stSerif}` “font” has been specified. The default is *Times*, which may be restored by specifying *fontname* as `default`. If *fontname* contains spaces, it must be enclosed in double quotes.

`fontfacemono(fontname)` specifies the name of the font to be used to render text for which the `{stMono}` “font” has been specified. The default is *Courier*, which may be restored by specifying *fontname* as `default`. If *fontname* contains spaces, it must be enclosed in double quotes.

`fontfacesymbol(fontname)` specifies the name of the font to be used to render text for which the `{stSymbol}` “font” has been specified. The default is *Symbol*, which may be restored by specifying *fontname* as `default`. If *fontname* contains spaces, it must be enclosed in double quotes.

Remarks and examples

[stata.com](https://www.stata.com)

Remarks are presented under the following headings:

[Using the `svg_options`](#)
[Setting defaults](#)

Using the `svg_options`

You have drawn a graph and wish to create a Scalable Vector Graphics file for including the file in an HTML document. You wish, however, to change text for which no other font has been specified from the default of *Helvetica* to *Arial*:

```
. graph ...                               (draw a graph)
. graph export myfile.svg, fontface(Arial)
```

Setting defaults

If you always wanted `graph export` (see [G-2] [graph export](#)) to use Arial when exporting to Scalable Vector Graphics files, you could type

```
. graph set svg fontface Arial
```

Later, you could type

```
. graph set svg fontface Helvetica
```

to change it back. You can list the current *svg_option* settings for Scalable Vector Graphics files by typing

```
. graph set svg
```

Also see

[G-2] [graph export](#) — Export current graph

[G-2] [graph set](#) — Set graphics options