



TMS FNC SVG DEVELOPERS GUIDE

November 2019
Copyright © 2019 by tmssoftware.com bvba
Web: <http://www.tmssoftware.com>
Email: info@tmssoftware.com

Index

Supported SVG features, elements	3
Adding SVG support to your application	4
Grayscale	5
WEB Support	5
TVirtualImageList (Rio and newer)	5

Supported SVG features, elements

TMS FNC Core supports SVG files and can import and render SVG. SVG support is limited to

- Uncompressed SVG files in XML format

SVG supported elements:

- circle
- clipPath
- defs (limited to predefined style, no gradients, no paths, only single fill/stroke parameters)
- ellipse
- g
- image (limited to base64 encoded string value)
- line
- path
- pattern (no gradients, no paths, only single fill/stroke parameters)
- polygon
- polyline
- rect
- style

Within the style element supported parameters are:

- fill (no gradients)
- stroke (no gradients)
- stroke-width
- opacity

Within the primitives (line, polygon, polyline, rect, path) supported parameters are:

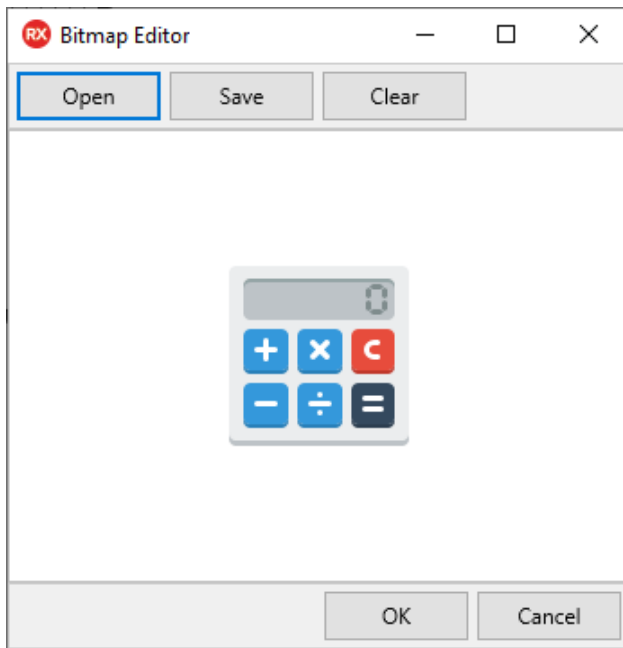
- fill (no gradients)
- stroke (no gradients)
- stroke-width
- transform (scale, rotate, translate, matrix)

With SVG support in TMS FNC Core we aim for smaller SVG icon sets that can be used to target normal and high DPI applications. When scaling, SVG will scale accordingly and still render sharp at different resolutions. SVG support has been added to avoid the need for numerous icons for each different scale / resolution.

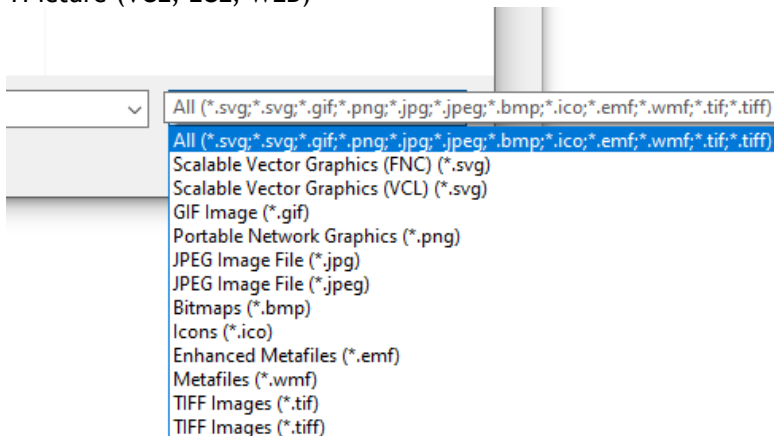
Adding SVG support to your application

SVG support can be added to your application by using the built-in TTMSFNCBitmap editor (FMX, VCL) or with TPicture default designtime open file dialog (VCL, LCL, WEB). When the open file dialog is shown at designtime, an option is available for picking SVG files.

TTMSFNCBitmap editor (FMX, VCL):



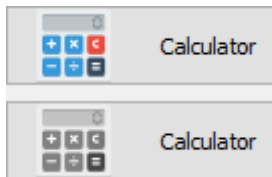
TPicture (VCL, LCL, WEB)



Runtime support includes LoadFromFile, LoadFromStream, SaveToFile, SaveToStream as this is the case with other file image types such as PNG, JPG, GIF, BMP, ICO,

Grayscale

After loading an SVG, there is a runtime option to apply grayscale to an SVG. This way you can enhance your application with disabled states even with only one icon loaded into the component. Below is a sample applied to the TTMSFNCButton. This is not supported for WEB (see chapter below for more information on SVG WEB support).



FMX

```
TMSFNCButton1.Image.Bitmap.SVG.GrayScale := True;
```

VCL, LCL

```
(TMSFNCButton1.Image.Bitmap.Graphic as TTMSFNCSVGBitmap).SVG.GrayScale := True;
```

After setting the property you'll need to call a repaint instruction:

```
TMSFNCButton1.Image.Repaint;
```

WEB Support

WEB supports SVG out of the box. There is no separate import / rendering engine necessary. Each component that can show images is capable of rendering SVG. There are no limitations except the ones that are tied to the SVG specification for HTML / JavaScript pages. The result of loading SVG can differ from desktop applications written in FMX, VCL and LCL especially the scaling part. Some SVG files have a ViewBox parameter that does not match the content size, and this is internally solved with aspect ratio parameters at the SVG element. This combination has been tested in our components and showed SVG files that ignore aspect ratio.

TVirtualImageList (Rio and newer)

Starting from Rio, TVirtualImageList is supported and can attach a separate image collection. This has been added to have a way to support high DPI images in controls.

With TMS FNC Core, we have added an extension to the base image collection to add support for SVG. The TTMSFNCSVGImageCollection component is registered separately and can be attached to the TVirtualImageList component. The TVirtualImageList component is responsible for generating images depending on the resolution. With SVG, you can attach a single list of SVG files via the TTMSFNCSVGImageCollection and your application will have auto-scaled images based on SVG for each resolution.

With SVG, there is no need to specify multiple icons with various sizes and you don't need to worry about the resolution.