

# Svg2tex Documentation

Lorenzo Tozzi

December 7, 2008

## Contents

<b>1</b>	<b>Licence</b>	<b>1</b>
<b>2</b>	<b>Intorduction</b>	<b>1</b>
<b>3</b>	<b>Usage</b>	<b>2</b>
3.1	Inkscape extension . . . . .	2
3.2	Command-line . . . . .	4
<b>4</b>	<b>Conclusion</b>	<b>5</b>

## 1 Licence

Svg2tex is released under the GNU/GPL version 2 license. Svg2tex is free software.

## 2 Intorduction

When I was writing my thesis with L<sup>A</sup>T<sub>E</sub>X, I wanted to include pictures in a vector format with the following features:

1. easily edit pictures without loosing too much time;
2. use L<sup>A</sup>T<sub>E</sub>X commands inside pictures, even custom ones defined in the main document;
3. have a consistent font and font size across the document, even inside pictures;

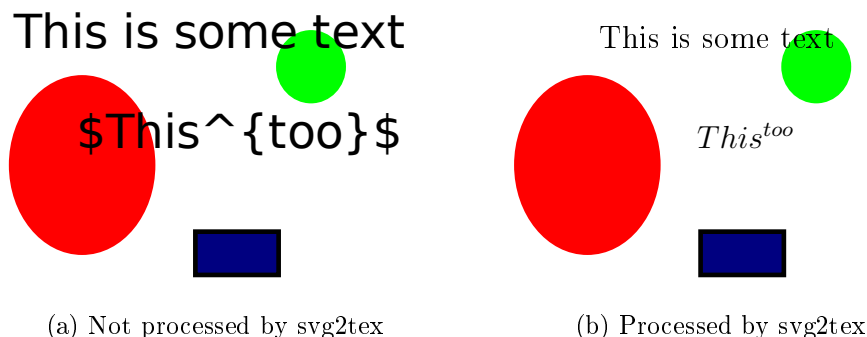


Figure 1: Pictures comparison

4. change font and font size without having to re-edit every picture; ideally just editing  $\text{\LaTeX}$  document's header.

I didn't find a suitable solution for all those issues, and that's why I've created `svg2tex`.

`Svg2tex` is a python script that extracts all text from a \*.svg file to a  $\text{\LaTeX}$  picture environment. This way the picture's text is processed directly by  $\text{\LaTeX}$  and can be included into the document. Figure 1 compares the final result with and without `svg2tex`.

## 3 Usage

`Svg2tex` can be used as an inkscape extension or as a standalone script. The final result is the same.

### 3.1 Inkscape extension

Installing `svg2tex` is very easy. Under Linux you only need to copy `svg2tex.py` and `tex_output.inx` under `"/home/<your username>/.inkscape/extensions"`. Under Windows just copy the same files under `"C:\<inkscape installation directory>\share\extensions\"`. At this point it's necessary to restart inkscape. Under the save menu there should be a new option: *"LaTeX (text only) picture environment (\*.tex)"*.

As an example we will create a simple picture as shown in picture 2. The first two lines have different sizes, while the third one has a  $\text{\LaTeX}$  command in it. At this point click on *"Save"* and choose *"LaTeX (text only) picture environment (\*.tex)"*. Set the file name as `example.tex` and click *"Save"*. We will be prompted with a pop-up window as in picture 3. For now leave



Figure 2: The example document



Figure 3: Inkscape pop-up

it empty.

The newly created file can now be imported into a  $\text{\LaTeX}$  document with the following code:

```
\begin{figure}
\centering
\input{example.tex}
\caption{This is an exaple file}
\label{fig:example_label}
\end{figure}
```

The main document must include `graphicx` and `rotating` packages in order to safely include pictures generated with `svg2tex`. The final result can be seen in picture 4. As you can see, the red shapes are missing. This happens because `svg2tex` only converts `*.svg` document's text. The rest of the image must be loaded as a "background".

Hello!

This is a big text

No, this is a big text

Figure 4: Svg2tex output

Now go back to inkscape and overwrite `example.tex`. This time, when we are prompted with the pop-up window that asks for a background image (picture 3) type in “`example_bg`”. At this stage `example.tex` is waiting for a background image called “`example_bg`” (file extension is added by L<sup>A</sup>T<sub>E</sub>X).

We must decide which file format to use for that background. L<sup>A</sup>T<sub>E</sub>X can import a wide range of vector or raster images such as `*.eps`, `*.png` and `*.jpeg`. It depends if we are using “`latex`” or “`pdflatex`” to compile our document. More informations here: [http://en.wikibooks.org/wiki/LaTeX/Importing\\_Graphics](http://en.wikibooks.org/wiki/LaTeX/Importing_Graphics). Choosing the appropriate format is outside the scope of this document. We are importing the picture as a vector image and we are using the `*.pdf` format.

Now we must delete all text from the picture (it’s already in our document, at this step we only need the red shapes) and click on “*Save*”. We have told to `example.tex` that the background image is called “`example_bg`” so the filename must be `example_bg.pdf` and must be placed in the same directory of `example.tex`. As the file format we choose “*PDF via Cairo (\*.pdf)*”. The final result is shown in picture 5.

As expected, the font and the font size is consistent with the rest of the document, so inkscape’s preferences are ignored and the first two lines have the same height. At the same time the L<sup>A</sup>T<sub>E</sub>X command `\huge` is correctly processed and the third line is bigger than the others.

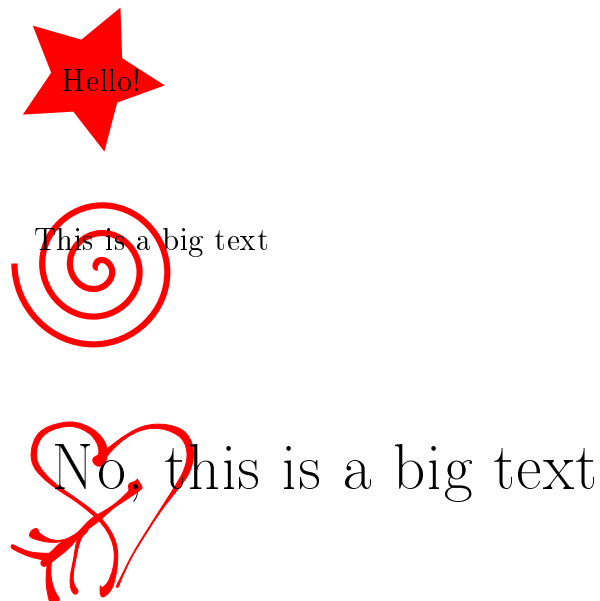


Figure 5: Svg2tex output with the proper background

## 3.2 Command-line

Svg2tex can also be called from command-line. Here it is the syntax:

```
python svg2tex.py <options> <svg-input-file> <tex-output-file>
```

The `<tex-output-file>` is optional and, if it's not given, the  $\text{\LaTeX}$  output is printed in standard output. The `<options>` can be any of the following:

- `-i <filename>` (or `--include <filename>`) – set the background image of the picture environment to `<filename>`. It's the same as entering `<filename>` in inkscape as seen in picture 3;
- `-t <filename>` (or `--textless <filename>`) – make a copy of the original `*.svg` file without text and save it as `<filename>`.

Even from command-line, `svg2tex` can be paired with `inkscape`. The next simple shell script, for example, converts all `*.svg` files in the directory into ready-to-use  $\text{\LaTeX}$  + `*.pdf` pictures:

```
#!/bin/sh

for file in *.svg
do
    echo "Processing ${file}..."
```

```

fn=${file%.svg}
python svg2tex.py -i "${fn}" -t "${fn}.tl.svg" "${file}" "${fn}.tex"
inkscape --export-pdf="${fn}.pdf" "${fn}.tl.svg"

rm "${fn}.tl.svg"
done

```

## 4 Conclusion

While I was working on `svg2tex` I focused around the command-line approach because it's a lot faster. It's possible to edit any of the `*.svg` files and run a simple shell script (like the one provided in this document) to update both, the `*.tex` and `*.pdf` counterpart. The `inkscape` extension, on the other hand, make it easy for windows users to use the script since `python` is embedded into `inkscape`. But at this time I feel it's too complicated and too slow. For each picture you have to:

1. save the  $\text{\LaTeX}$  file using `svg2tex`;
2. delete all text;
3. save the “background”.

The ideal solution should be a combined save  $\text{\LaTeX}$  + PDF or  $\text{\LaTeX}$  + EPS as it's done in `Xfig`. I'll contact `inkscape` developers ASAP to find out how to “dual save” a file. For now I feel that the `inkscape` extension is a clunky hack, but still usable.