

Export an Image to SVG

Medium size project proposal for Krita with Google Summer of Code

Introduction and Motivation

Krita as a drawing application is expected to be able to read and write to many different formats for interoperability with many different programs and use cases. SVG is a common file format that Krita is able to read but not fully save as. Currently Krita only supports exporting a single vector layer as SVG. SVG 1.1 and SVG 2 both offer the same functionality for embedded raster images. Considering that svg 2 is not recommended by the World Wide Web Consortium and Krita also only currently supports importing of svg 1.1 so that will be the version we support for exporting as well. This project proposes the ability to export as SVG 1.1 with support for both raster and vector layers.

Project Goals:

- Krita should be able to export the current file as group SVG
- Raster layers should be exportable to SVG and saved as embedded PNG's in the SVG file
- Vector layers should be saved as a group of strokes
- A kra-file with layers saved as SVG, should be later be opened by Krita as a document with layers

Implementation

1. Create a new export option for SVG

Add a `krita_svg_export.json` to the `svg` plugin in `krita/plugins/impex/svg/`. When `KoFileDialog` is called from the `save/save as/export` action it will pickup the new option. The contents of this file should be something like:

```
{  
  "Id": "Krita SVG Export Filter",
```

```

"NoDisplay": "true",
>Type": "Service",
>X-KDE-Export": "image/svg,application/x-krita-paintoppreset",

>X-KDE-Library": "kritasvgexport",
>X-KDE-ServiceTypes": [
    "Krita/FileFilter"
],
>X-KDE-Weight": 1,
>X-KDE-Extensions" : "svg"
}

```

When doing the save/export the file type specified during the KoFileDialog call is used to determine which KisImportExportFilter is used to export the document. A new svgExport filter class will be created that inherits KisImportExportFilter. This will be created in krita/plugins/impex/png/kis_svg_export.cc. The convert function of our new svgExport class will implement saving our document as SVG. Because a .kra file is the only file format that supports saving all of Krita's functionality the user will need to be warned about the limited capabilities of exporting in SVG. These will be defined in svgExport::initializeCapabilities() and include capabilities like supported color profiles, layer support, and support for different layer types. The full range of possible missing feature set is created in KisExportCheckRegistry::KisExportCheckRegistry() and each option will need to be specified whether it is supported or not.

2. Handle iterating over all vector layer and passing in shapes

The current functionality to export a layer as svg uses the SvgWriter class while passing it a QList of shapes from the current Vector Layer. This gets assigned to the internal shapes topLevelShapes list. When the Vector Layer gets saved as an SVG each stroke will get saved as a separate element in the XML of the SVG. While it seems like the SvgWriter does allow saving by group (with shapes inside that group) its current usage doesn't make use of that ability. The new convert function will use the svgWriter making use of the saveGroup function to group the various vector layers. The list of vector layers will be parsed from the KisDocument passed into convert, making sure to keep ordering of layers. As an alternative the order of layers may be retrieved from KoShape::zIndex().

Although `KoShape::zIndex()` will store the whole image memory so this should be avoided if possible. At this stage we will ignore any layer that is not a vector layer.

3. Convert raster layers to PNG and embed inside SVG

`SvgWriter` is currently able to accept an `ImageShape` in order to write to an SVG. A raster image can be converted to this format with `ImageShape::ImageShape()` using `layer-projection`.

Krita should already support reading embed raster layers from parsing in `SvgParser::parseSingleElement()` and checking for “image” tags in the SVG. However, loading color profiles doesn’t always succeed when running `SvgLoadingContext::parseProfile()`. `parseProfile()` doesn’t support every possible way to specify color profiles in SVG’s. Namely when the attribute is labeled as “name” instead of “local” the parser should still be able to load the correct color profile.

4. Preserve layers inside SVG

Any current raster, vector, or group layer when saved with a SVG should be retained when re-opening the document with Krita. The `SvgWriter::saveLayer` currently does save the layer but without any identifier that the saved group is a layer in any way. This function will be extended to add a “label” tag to specify the layer name and a “groupmode” tag to specify the layer type. When opening the SVG the `SvgParser::parseGroup()` function will need to check for these tags and create the specified layer.

Timeline

| Timeline | Task |
|-----------------------|--|
| May 20th - June 6th | Community bonding period Familiarization with the necessary portions of the code base Reading on the SVG specification |
| June 6th - June 13th | Add export config for SVG plugin |
| June 13th - June 27th | Implement new <code>svgExport</code> class |

| | |
|-----------------------------|--|
| June 27th - July 4th | Create unit tests for svgExport and SvgWriter |
| July 4th - July 18th | Extend svgExport to handle parsing the krita document and saving vector layers in SVG |
| July 18th - August 1st | Implement converting raster layers to PNG and enabling file format capabilities Submit first Google Summer of Code evaluation |
| August 1st - August 15th | Extend saveLayer and parseGroup to write and read layer information inside the SVG |
| August 15th - September 5th | Buffer time Submit final work and evaluation |
| September 5th | End of Google Summer of Code |

About me

Hello! I am Reinold Rojas, a Devops Engineer in New York City, USA. I have been a Devops/System Administrator for a few years doing infrastructure work at various companies. I've worked with teams across the globe so I should be fine doing that with Google Summer of Code as well. At a previous position I found I wasn't getting as much development work as I would like, so I decided to contribute to an open source project. I chose to contribute to Krita as a way to motivate me to learn to draw while satisfying my programming itch.

I am mostly a self taught programmer, although I did take some introductory computer science classes at Queens College (City University of New York). I learned most of my programming skills, and skills for my role as a Devops Engineer, by doing my own research, reading through documentation, and challenging myself to build things with what I learned. This honed my ability to figure out solutions and self direct my work.

Other Scheduling

I plan to maintain a full time job while doing Google Summer of Code. I specifically chose a medium project (around 175 hours of work) in order to make this manageable. I plan to work about 1 hour a day Monday to Friday and 10 hours on the weekend totaling 15 hours of work a week. Over the course of three months this should be more than the

175 hours of work for the project. I am also flexible in the start or end date so I add more time if necessary.

Communication

I plan to communicate in the IRC channel daily to notify on daily progress and work. Weekly during the Krita meeting I would give a synopsis of the past work, how the progress looks for the project, and if there are any blockers I am facing. I will also post a more formal blog update to PlanetKDE and the krita-artists forums. Both places are great for letting the community know about the progress of the work.

Prior Contributions

I've been contributing to Krita for about a year with about [two dozen](#) merge requests accepted. Here are some of examples of my work that I've enjoyed the most:

- [Add Transition for docker recorder preview](#)
- [Add configurable zoom scaling option](#)
- [Enable color sample preview for Color Sample Tool](#)
- [Bug 318347 grid brush keeps grid alignment when increasing brush size](#)
- [Fix artifacts in freehand selection tool in polygonal mode](#)

I am also active on the Krita Artist forums and [sometimes implement the requests](#) posted by artists.

Contact Info

Name: Reinold Rojas

IRC: reinold

KDE identity: [reinoldrojas](#)

Email: rojasreinold@gmail.com

Linkedin: <https://www.linkedin.com/in/reinoldrojas/>

Krita Artists: <https://krita-artists.org/u/reinold/summary>