Introduction

This documentation covers the brief *librsvg* details and its implementation and integration processes inside Vestel Television Software. The *librsvg* is Scalable Vector Graphics (SVG) render library which converts files to *Cairo surfaces* after rendering. The *Cairo* surfaces handle converted SVG data information by storing inside a memory of block. This memory of block or buffer allows reflecting the SVG image file to the end-user. During the documentation, each step will be explained and information about other SVG render libraries will be given.

Why use SVG render library inside Vestel Television (TV) Software?

New application version of TV has a bunch of extra features and one of them is integration with Amazon Alexa. Alexa allows end-user to interact with TV over voice command. This voice records converted to the text and if it is not special command then the search on the Internet is initialized. The result of the search is reflected on the TV screen after completed. The search results can contain a lot of different kind of data or file types which means result can be text or can contain image file too. Majority of these files and data types can be handled with the libraries used inside of the software. For image files, which are Portable Network Graphics (PNG) and Joint Photographic Experts Group (JPEG) typed, the software can decode and store it in the buffer and allow to show data from the buffer. But for SVG files, there is a small render engine which is mainly developed for a specific problem and it can handle only restricted and basic content which defined in World Wide Web Consortium (W3C) for SVG. Primarily, this engine is used to handle TV itself menu and other SVG files which are used in the user interface. Due to this drawback of the engine, any result from search engine covers SVG files are not handled properly. To extend this feature and boost SVG engine and cover all specifications committed by W3C, the usage of suitable SVG render library is decided.

Research and Requirements

Before research for the new suitable library, some requirements are listed. The TV platform is on Advanced RISC Machine (ARM) Architecture and some restrictions are absolutely exist which are the memory, usage of CPU and finally compilable of library on this platform. The new library can also support complicated SVG files and can handle store of rendered data in a block of memory.
The research shows that there are a lot of useful libraries exists and all has some pros and cons. The selection of *librsvg* for SVG render process is done.

Librsvg is an open source SVG render library developed as a part of GNOME project. It is lightweight and does not require to much operation. This feature is already cover first requirement which are memory and CPU usage. It can also be compiled on x86 platform which TV Simulator application run on and ARM platform which TV uses. The second main requirements also covered. The compilable of the library on multiple platforms, lightweight and fast render features are the primary factors which differentiate this library from others. Version 2.39 is selected for implementation. In the next versions of the library, implementation language is ported to Rust programming language. Currently, this language is an obstacle for the TV platform because Rust is not supported in the platform by the vendors. But this selected version is written on C programming language and allows us to use it in the platform.

The library uses two helper libraries on the rendering process too. The *libxml* library is used to parse the XML code orientation of SVG to make it quickly accessible by the library. The *Cairo*, which is the second library, is used to render the information obtained by the library to a memory block. Finally, this allows us to show SVG image on the TV screen from the buffer.

Implementation and Integration

The required libraries installation and configurations are covered in the documentation of the library. After covering the configuration procedure, the simple source code is implemented and tested on an SVG image. After getting success result the integration to the TV application is started and finished and tested on the TV simulator. But the test on the platform is still on pending. The required libraries integration is on progress by platform software architecture.

Input: tiger.svg **Output:** tiger.png

Other SVG Render Libraries and Details

Name	Description	Language	Why Not Selected ?
resvg	It has same functionalities like *librsvg* and it is the main competitor to the *librsvg*. The *resvg* does a lot of preprocessing before rendering. It converts shapes to paths, resolves attributes, removes groups and invisible elements, fixes a lot of issues in malformed SVG files. It has extra features which *librsvg* is not support.	Rust	Not supported by platform
AndroidSVG	*AndroidSVG* is a SVG parser and renderer for Android. It is suitable for the Android platform as seen from the library name	Java	Not supported by platform
wxSVG	*wxSVG* is the library to create, manipulate and render SVG files with the help of *wxWidgetToolkit*. *wxWidgetToolkit* is cross platform GUI library.	C++	Supported by platform but require *wxWidgetToolkit*. The TV application has its already developed GUI. Extra GUI library configuration causes useless memory usage
NanoSVG	*NanoSVG* is a simple small SVG parse. The output of the parser is a list of cubic bezier shapes. Support not all the features listed in W3C for SVG and has	C	Supported by platform but extra libraries and integration are required to get data from the parser and store it in the

	small community which supports.		buffer after some conversion operation. Due to not support all features in W3C this library is not selected
svgren	SVG renderer library is implemented using C++. *Svgren* uses *svgdom* to read the SVG files and *Cairo* to render graphics. *Svgren* is supposed to conform to SVG1.1 specification except for some features. The community plans not to support text tag for the future.	C++	Supported by platform. But the read process on *svgdom* can cause more CPU usage for complicated SVG files and slow down TV. Not selected due to performance
SVGTiny	*Libsvgtiny* is an implementation of *SVGTiny*, written in C. The overall idea of the library is to take some SVG as input, and return a list of paths and texts which can be rendered easily. The library does not do the actual rendering.	C	Not actually a rendering library
jfreeSVG	*jfreeSVG* is a fast, light-weight, vector graphics library for the Java platform that makes it easy to generate graphical output in SVG format directly.	Java	Not supported by platform
svgSalamander	*svgSalamander* is a light weight SVG renderer and animator for Java.	Java	Not supported by platform
Jaxb-svg11	*Jaxb* is the content model for SVG and is developed for Java Platform.	Java	Not supported by platform
Batik-svg-dom	It is java SVG toolkit.	Java	Not supported by platform
Lib-gwt-svg	It is a general purpose SVG library for Google Web Toolkit (GWT). The goal is to make it easy to do SVG graphics in a GWT application.	Java	Not supported by platform
svg2vector	It is toolbox with applications to convert SVG graphics to other vector formats and some bitmap formats.	Java	Not supported by platform
simplesvg	*SimpleSVG* project has been renamed to Iconify.design. This repository is archived and no longer updated.	Javascript	This library is not longer supported. Not suitable to use in TV software.
MonkSVG	An SVG parsing framework which is implemented in C++. But not any documentation and usage case exists for this library.	C++	Not any documentation provided for this library and it can cause to side-effects for usage in software
Inkscape	*Inkscape* is often used to convert SVG	C++	Big library to use and it

	to PNG. It is not a tiny library and very slow. But it has the best SVG support amongst others.		is SVG editor. Simple usage of conversion SVG to PNG or another format to use seems pointless in software
QtSVG	Without a doubt, *QtSVG* is heavily used in *Qt* applications. But *QtSVG* itself is very limited. In simple terms it correctly renders only primitive SVG images.	C++	Not suitable to use in application for only support primitive SVG images
Rasterific-SVG	It is an Haskell library used to render SVG files. Parse SVG file to a JuicyPixel image or Rasterific Drawing.	Haskell	Not supported by platform

Conclusion

In conclusion, the documentation covers the methodology of research and integration phases. Some libraries in the list can support rapid rendering or some extra feature like *resvg* does with removing malformed content. But those libraries are not suitable to use in TV application due to restrictions on the platform. Due to these characteristics, *Librsvg* is more suitable to use inside Vestel TV Application.