# The packages svg and svg-extract

Philip Ilten (2012–2016) Falk Hanisch (2017–)

https://github.com/mrpiggi/svg hanisch.latex@outlook.com

v2.02j (2020/10/23)

The **svg** package is intended for the automated integration of SVG graphics into IATEX documents. The capabilities provided by *Inkscape*—or more precisely its command line interface—are used to export the text within a SVG graphic to a separate file, which is then rendered by IATEX. The two commands \includesvg and \includeinkscape are provided as central user-interface, which are very similar to the \includegraphics command of the **graphicx** package.

In addition, the package **svg-extract** allows the extraction of these graphics into independent files in different graphic formats, exactly as it is rendered within the LATEX document. For the creation of these graphics in the well-known formats PDF, EPS and PS, LATEX and possibly conversion tools shipped with the distribution are used. If the graphics are required in other file formats, either *ImageMagick* or *Ghostscript* can be invoked.

The command line interface (CLI) of *Inkscape* 1.0 has changed in comparison to previous versions. In order to provide a comfortable user-interface for invoking *Inkscape*, the used version is detected and if necessary switch to the outdated syntax of the CLI. If this approach fails for some reason, you can set the version of *Inkscape* manually with inkscapeversion=0 or inkscapeversion=1.

# Contents

I.	User documentation	2
1.	Introduction	2
2.	Usage of package svg	3
2.1.	General settings	4
2.2.	Options for the invocation of <i>Inkscape</i>	4
2.3.	Options for the graphic inclusion	6
2.4.	Including SVG files	6
2.5.	Including already exported SVG files	7
3.	Usage of package svg-extract	7
3.1.	General settings	8
3.2.	Extract independent graphic files	8
3.3.	Convert extracted graphic files	10
3.3.1.		11
3.3.2.		12
4.	Example	13
<b>5</b> .	Troubleshooting and reporting issues	14
6.	Include SVG files created with ROOT	15

II.	Implementation	
Α.	Initialization	
A.1.	Packages	
A.2.	Dealing with catcodes	
A.3.	General macros	
	Macros for process control	
	String manipulation	
	File handling	
	List handling	
В.	Including SVG files with package svg	
B.1.	Options	
B.1.1.	<u>▲</u>	
B.1.2.	Setting input folder and file	
B.1.3.	Setting output folder and file	
B.1.4.	Options for the inclusion of graphics	
B.2.	User commands	
B.2.1.		
B.2.2.	Definition of user commands	
B.3.	Auxiliary macros	
B.4.	Handling path and file names	
B.5.	Patches	
C.	Extracting independent graphic files with svg-extract	
C.1.	Options	
C.1.1.	Controlling the extract process	
C.1.2.	Invoking external application for graphic conversion	
C.1.3.	Setting output folder	
C.1.4.	Options for the extraction of graphics	
C.1.5.	Miscellaneous options	
C.2.	User commands	
C.3.	Auxiliary macros	
C.4.	Commands for the separate auxiliary LATEX-file	
D.	Processing Options	
Index		
Chang	ge History	

# Part I.

# **User documentation**

# 1. Introduction

The open source program *Inkscape* has provided an excellent resource for the simple and easy creation of images and diagrams using a graphical user-interface. The work by Johan B. C. Engelen has further enhanced the ability of *Inkscape* to split a SVG file into a text component that can be compiled with LATEX, and an image component that can be imported as a PDF file. For further information see the documentation of **svg-inkscape** on CTAN<sup>1</sup>. The procedure described therein is taken up and consistently expanded. Thus, it is now possible to include a SVG file into a LATEX document where the text within the SVG graphic will be rendered natively by LATEX.

<sup>1</sup>http://www.ctan.org/pkg/svg-inkscape

Both packages svg and svg-extract rely heavily upon executing commands on shell using the \ShellEscape command—or respectively the old known \write18—for executing the CLIs of the applications mentioned above. So passing flag --shell-escape to the IATEX engine is utterly essential when using package svg and/or svg-extract. The executed commands and the possibilities to adapt their invocation with the appropriate options are described later on in this documentation. All this is done automatically with the \includesvg command. If you don't want to use the --shell-escape flag, either for security reasons or because the export of the SVG files is done in another way, there's also the command \includeinkscape which includes files already exported by Inkscape.

An working installation of *Inkscape* is required for the automated integration of SVG graphics, whereby the installation path must be known to the operating system. This can be checked on shell by typing <code>inkscape -V</code>. Moreover, there are some required packages which are loaded by packages <code>svg</code> and <code>svg-extract</code> to provide the functionality. These are:

iftex for flow control depending on the used LATEX engine

scrbase for the definition and handling of options in key-value-syntax

pdftexcmds, shellesc to allocate the same primitives independent of the used LATEX engine ifplatform to control the file access depending on the operating system

trimspaces to remove unwanted spaces in file paths

graphicx for including the graphic files after the Inkscape export

xcolor, transparent are possibly needed by the separate LATEX files created by Inkscape xr is used by svg-extract in order to include labels within the independent graphic files

If you want to pass options to package graphicx, you must either load it before package svg

or use \PassOptionsToPackage.

The usage of packages **xcolor** and **transparent** can be switched off while loading package **svg**. See the two options **usexcolor** and **usetransparent** below.

# 2. Usage of package svg

The purpose of this package is to include standalone SVG graphics into a LATEX document. The command \includesvg is defined which does all necessary steps for this task. It first launches the export of a SVG file to a supported file format with Inkscape, if necessary, and includes the exported graphic file afterwards. The usage and the syntax is quite similiar to the command \includegraphics from the graphicx package. In fact, the inclusion of the exported graphic file is done with \includegraphics.

usexcolor (opt.)
usetransparent (opt.)
noxcolor (opt.)
notransparent (opt.)

The packages **xcolor** and **transparent** are loaded by default at the end of package **svg**. The listed options are intended to prevent these packages from loading. They are the only options which have to be given while loading the **svg** package. All supported boolean values (true/on/yes/false/off/no) can be assigned to usexcolor and usetransparent, while noxcolor and notransparent don't accept any value.

```
\label{local_local_local_local_local_local} $$ \arrowvert = \{options\} \
```

Due to the way the LATEX kernel parses package options, problems may occur if any option other than those just mentioned above – meaning the options explained hereafter – are passed through the optional argument of  $\scalebox{usepackage}[\langle options \rangle] \{svg\}$ . It is strongly recommended to use  $\scalebox{vsgsetup}\{\langle options \rangle\}$  for these after loading the package instead.

# 2.1. General settings

\svgsetup

All other options described in detail below can also be changed after loading the package either in the preamble or within the document. They don't have to be given as optional argument to  $\space{logorithms}\$  but can be set by using macro  $\space{logorithms}\$  where  $\space{logorithms}\$  is a comma separated list of options. These settings are done in the current scope which means either globally or within the current group.

```
\displaystyle \sum \{\langle options \rangle\}
```

Further, with the optional argument of commands  $\includesvg[\langle options \rangle] \{\langle svg\ filename \rangle\}$  or  $\includeinkscape[\langle options \rangle] \{\langle graphic\ filename \rangle\}$ , it's possible to reset any setting locally for a certain file.

\svgpath

Most likely you want to organize your SVG files in a separate folder either as a subfolder in the working directory or elsewhere in your local folder structure. For this purpose, a list of root paths to SVG files can be specified using the \svgpath command in the same way as \graphicspath is used. Every path has to be given in a group of braces {}—even if there is only one—and should terminate with a slash. For example:

```
\svgpath{{svg/}{/usr/local/svg/}}
```

would cause the system to look first in the subdirectory svg/ and afterwards in the absolute path /usr/local/svg/. Further, if no path was specified with \svgpath or the desired file wasn't found, all directories given with \graphicspath are searched too. Please keep in mind that the current working directory is browsed first in any case. It is recommended to avoid umlauts or any other Non-ASCII characters as well as any spaces and/or quotes respectively \dq both in paths and file names. Especially when DVI output is active using quotes will certainly cause an error.

# 2.2. Options for the invocation of *Inkscape*

inkscape (opt.) This option controls, when the export with Inkscape is invoked and is true by default.

false/off/no

Inkscape won't be invoked in any case, no export is done.

true/on/yes/newer/onlynewer

The export with *Inkscape* will only be done, if the exported graphic file either does not exist or the file modification date of the SVG file is newer than that of the exported graphic file. Thus the compilation time of the LaTeX document can be reduced to the necessary minimum.

forced/force/overwrite

The *Inkscape* export will definitely be done, any already existing exported file will overwritten regardlessly.

In addition to controlling the export behavior, the option inkscape can also be used to make additional settings, which then acts as a wrapper for the options described below.

inkscapepath (opt.)

The option inkscapepath specifies, where the resulting files of the *Inkscape* export should be located. The default setting is *basesubdir*, which uses the subfolder ./svg-inkscape/ within the current working directory.

#### svgdir/svgpath

The PDF/EPS/PS/PNG graphic files as well as the LATEX files generated by *Inkscape* will be located in the same directory as the corresponding SVG file.

#### svgsubdir/svgsubpath

Within the folder of the encountered SVG file, all exported files will be located in a subfolder named svg-inkscape/.

## basedir/basepath/jobdir/jobpath

All exported files will be located in the current working directory.

#### basesubdir/basesubpath/jobsubdir/jobsubpath

A subfolder named svg-inkscape/ within the current working directory will be used for files generated by *Inkscape*.

#### /path/to/somewhere/

It is also possible to give a custom path, either relative to the current working directory (./relative/path/) or as an absolute path.

#### inkscapeexe (opt.)

For including a SVG file, *Inkscape* is used to separate the text and image from the SVG file itself. In order to use the command line interface on shell, the path where the executable is located has to be known to the operating system and is assumed to be inkscape by default.

You can check if the default setting is valid for your system by typing <code>inkscape -V</code> into the terminal. If this fails and nothing is returned, you should add the binary directory of <code>Inkscape</code> to the environment variable PATH on your operating system. For the case, that this is not possible or you aren't willing to do so, you can alternatively pass option <code>inkscapeexe</code> to <code>\svgsetup</code> within the document preamble to set the absolute path where the executable of <code>Inkscape</code> is located.

Especially if the executable path to be defined *contains spaces*, it *must not* be passed as a package option but to \svgsetup{inkscapeexe=...} instead!

#### inkscapeversion (opt.)

The command line interface of *Inkscape* changed slightly from version 0.9x to 1.x and makes it necessary to distinguish between the two versions. By default, inkscapeversion=auto is set and the used version is automatically detected. This is done by calling *Inkscape*-CLI with parameter -V on shell—see option inkscapeexe described above. The returned result is evaluated by either piping stdout or eventually—if this fails—writing to a temporary file and read this back in (pipes with a potentially quoted path can not be used with MiKTEX).<sup>2</sup> It is also possible to switch off the automatic detection routine by setting the desired version manually with either inkscapeversion=0 to legacy mode or inkscapeversion=1 to the current CLI version.

#### inkscapename (opt.)

Inkscape export file names are derived from the SVG file name by default. However, the name of the exported file can be customized with  $inkscapename = \langle filename \rangle$ . It is possible to use counters for specifying the name of the exported file. Repeatedly specifying the same file name will overwrite previously created files.

# inkscapeformat (opt.)

With this option, the *Inkscape* export format can be controlled. Valid values are pdf, eps, ps and png, where a LATEX export is not possible for png and option inkscapelatex won't have any effect. By default, inkscapeformat=pdf is set unless DVI output was detected. In this case inkscapeformat=eps is the default setting.

#### inkscapelatex (opt.)

If option inkscapelatex=true is set, the output is split into a separate PDF/EPS/PS file (see option inkscapeformat) and a corresponding LATEX file. This is the default setting. Setting inkscapelatex=false will result in a single PDF/EPS/PS file, where any contained text won't be rendered by LATEX.

#### inkscapearea (opt.)

This option controls which area of the SVG file should be exported, *drawing* is set by default.

#### drawing/crop

The area exported corresponds to the bounding box of all objects in a drawing, including any that are not on the page.

#### page/nocrop

The area exported will correspond to the defined page area within the SVG file.

<sup>&</sup>lt;sup>2</sup>If this fails too, the *Inkscape* version is guessed when macro \svg@ink@run is used the very first time.

 ${\tt inkscapedpi}\ ({\rm opt.})$ 

The resolution used either for PNG export or for fallback rasterization of filtered objects when exporting to PDF/EPS/PS file. For PNG export it is set to 300 dpi by default, if no value was given. The given value should be a positive integer. The default behaviour can be reversed after a given value with inkscapedpi=\relax.

inkscapeopt (opt.)

You can use this option to pass additional switches to the Inkscape command line interface. For further information see the documentation of  $Inkscape^3$ .

 ${\tt svgextension}\;({\rm opt.})$ 

The package assumes SVG files with .svg extension as source for the *Inkscape* export. This option can be used to change this behaviour. For example, in order to process .dia files instead of .svg you could use

 $\includes vg[svgextension=dia, \langle additional options \rangle] \{\langle filename \rangle\}$ 

# 2.3. Options for the graphic inclusion

width (opt.)

height (opt.)

distort (opt.)

 ${\tt scale}\;({\rm opt.})$ 

The width of the included graphic file can be specified via the width option and the height by the height option. If both the width and height are specified, the figure will be scaled such that neither of the specified dimensions is exceeded, unless option distort=true is given. If width and/or height once have been set, this can be undone by setting them to Opt or \relax. If neither width nor height are set, the included graphic file can also be scaled by setting scale to a positive real number.

pretex (opt.)
apptex (opt.)

Commands prior and post to the inclusion of the graphic file may be desired, such as font or color commands. The options pretex and apptex are provided where the LATEX code given to pretex is included before the graphic file and apptex right afterwards. For example, to change the size of the included text one could use:

 $\includes vg[pretex=\tiny, \langle additional\ options \rangle] \{\langle svg\ filename \rangle\}$ 

 ${\tt draft}\ ({\rm opt.})$ 

This option can be used with boolean values and is equal to the identically named option of the **graphicx** package. If the **draft** option is given to **graphicx**, it's activated for **svg** as well.

 ${\tt lastpage}\ ({\rm opt.})$ 

A bug<sup>5</sup> concerning the L<sup>A</sup>T<sub>E</sub>X export has been reported for *Inkscape* 0.91. It may happen that within the exported L<sup>A</sup>T<sub>E</sub>X file, it's attempted to include more pages of the PDF graphics than actually exist. The **svg** package attempts to bypass the resulting error.

Consequently, the total number of pages is read and only existing PDF pages are included, if both options inkscapeformat=pdf and lastpage=true are set. This is the default setting (unless DVI output is active) and can be switched off with lastpage=false. It's also possible to set the number of the last page included of a PDF graphic manually as optional parameter for \includesvg or \includeinkscape. For details, see the description of the respective commands.

# 2.4. Including SVG files

\includesvg

The command \includesvg to include a SVG file is quite similar to the \includegraphics command provided by the **graphicx** package.

 $\includesvg[\langle parameters \rangle] \{\langle svg\ filename \rangle\}$ 

inkscape (param.)
inkscapeformat (param.)
inkscapelatex (param.)
inkscapearea (param.)
inkscapedpi (param.)
inkscapeopt (param.)
svgextension (param.)
width (param.)
height (param.)

apptex (param.)
draft (param.)

capeopt (param.)
tension (param.)
width (param.)
height (param.)
distort (param.)
scale (param.)
pretex (param.)

It is used right in the same way but where  $\langle svg\ filename \rangle$  is the file name of the SVG file, where any given file extension will be replaced with .svg ruthlessly. In order to change the source file format for the Inkscape export, you have to use parameter svgextension.

If the given file is not located in the current working directory but elsewhere on your file system, the command \svgpath could be used to specify this path. It is recommended to avoid umlauts or any other Non-ASCII characters as well as any spaces and/or quotes

<sup>3</sup>https://inkscape.org/de/doc/inkscape-man.html

<sup>&</sup>lt;sup>4</sup>to provide compatibility for package **graphicx**, it's possible to use **keepaspectratio=true** as alias for **distort=false** and the other way round

<sup>&</sup>lt;sup>5</sup>https://bugs.launchpad.net/ubuntu/+source/inkscape/+bug/1417470

respectively \dq both in paths and file names. Especially when DVI output is active using quotes will certainly cause an error.

The command \includesvg is intended to do an automated export with *Inkscape* at first, where the given SVG file is exported to a PDF/EPS/PS/PNG file (see inkscapeformat) and perhaps a correlating L<sup>A</sup>T<sub>E</sub>X file (see inkscapelatex). The export with *Inkscape* is only invoked, if the SVG file is newer than the exported graphic file or latter doesn't exist at all. Once the export has been done, the graphic file and maybe the L<sup>A</sup>T<sub>E</sub>X file are included.

All previously described options can also be used as optional parameters to \includesvg and do have the same effect as described before. However, the optional parameters specified have an effect only once when \includesvg is executed and remain unchanged afterwards.

 ${\tt lastpage}\ ({\rm param.})$ 

In addition to the use of boolean values, the parameter lastpage can also be assigned a specific (integer) page number, which defines the last used page of a PDF graphic. This, just like the identically named option, has an effect only when inkscapeformat=pdf is set.

angle (param.)
origin (param.)

Both parameters correlate to the identically named parameters of the \includegraphics command provided by the **graphicx** package. However, unlike to \includegraphics, they angle and origin are *always evaluated after* widht, height, distort and scale by \includesvg, regardless of the used order of the given parameters. This is mainly due to the inclusion of the LATEX files corresponding to the graphic files generated by *Inkscape*.

# 2.5. Including already exported SVG files

\includeinkscape

If you don't want to make use of the automated export with *Inkscape* but the user-interface provided by the **svg** package, you can use \includeinkscape instead of \includesvg.

 $\include inkscape [\langle parameters \rangle] \{\langle graphic\ filename \rangle\}$ 

inkscapeformat (param.)
inkscapelatex (param.)
width (param.)
height (param.)
distort (param.)
scale (param.)
pretex (param.)
apptex (param.)
draft (param.)
lastpage (param.)
angle (param.)

origin (param.)

You can use it similar to \includesvg but \( \text{graphic filename} \) has to be the filename of the already exported graphic file. If a valid file extension (.pdf/.eps/.ps/.png) is given, the current setting for inkscapeformat is overwritten. It's even possible to specify a file extension like .pdf\_tex to activate inkscapelatex. Furthermore, all optional parameters for \includeinkscape do have the same effect as described before for command \includesvg once when \includeinkscape is executed and remain unchanged afterwards.

# 3. Usage of package svg-extract

This package allows the extraction of independent graphic files out of SVG files which have been included and rendered with LATEX by the **svg** package. This is particularly useful when attempting to provide images to journals or collaborators, and one wishes the image to appear exactly as it does within the original LATEX document.

In order to extract to PDF, EPS, or PS files the programs pstoeps, pstopdf and pdftops are used which are usually provided by most of the LATEX distributions. In additon, the command line interfaces of *ImageMagick* and *Ghostscript* can be invoked for converting images in formats like PNG, JPG, TIF or something else. It's also possible to create PDF, EPS or PS files with one of the two programs. Therefor the desired program—magick and/or gswin32c/gswin64c on Windows respectively convert and/or gs on unix-like operating systems—must be installed. By typing \( \lambda program \rangle \) --version on shell, this can be checked.

If you want to extract independent graphic files from included SVG files, you only have to load **svg-extract**. All actions for the extraction process will be done by using \includesvg or \includeinkscape. Without any additional settings, the extraction will render the SVG file to the specified output formats(s) of choice using the same settings as specified within the two commands. Consequently, the scale between the image and text in the extracted files will remain identical to the scale within the document from which the SVG file was extracted.

In contrast to package **svg**, the console commands for graphic extraction are executed with each LaTeX run by package **svg-extract** when **--shell-escape** mode is activated. This behaviour can be switched off with option **extract=false**.

## Important changes

In version v1.0 of package **svg** the extracted files were named like the numbering of the current **subfig** environment by default. As package **subfig** sometime causes problems and because of the large amount of different IATEX packages which all provide the possibility to include subfigures with very different implementations, this feature can't be provided reliably by **svg-extract**. See option **extractname** for further information.

# 3.1. General settings

on (opt.)
off (opt.)

This options have to be given while loading the **svg-extract** package and are intended to toggle the functionality of this package. As both extracting and converting independent graphic files is invoked with every LATEX run when --shell-escape is activated, the option off can be given to save compilation time, once the creation of all desired images has been done and they no longer need to be re-generated. The option on can be used to reactivate functionality of this package. This can also be done by using extract=true/false.

\svgsetup \includesvg \includeinkscape With package **svg-extract** the applicable options for  $\svgsetup{\langle options \rangle}$  as well as parameters for the already described macros  $\includesvg[\langle parameters \rangle] {\langle svg\ filename \rangle}$  and  $\includeinkscape[\langle parameters \rangle] {\langle graphic\ filename \rangle}$  are extended. They can be used to control the process of graphic extraction and converting.

All options described below can be passed to  $\svgsetup{\langle options \rangle}$  and are then valid in the current scope. There also exist identically named parameters for the optional arguments of

These have an effect only once, when the specific command is executed.

# 3.2. Extract independent graphic files

extract (opt.)

This option can be used with boolean values. Using extract=true activates the functionality for both extracting and converting which is the default setting, whereas extract=false turns it off completely.

 ${\tt extractpath} \ ({\rm opt.})$ 

The path where the extracted and converted files are located can be specified with option extractpath, whereas basesubdir is set by default.

# svgdir/svgpath

The extracted and converted independent graphic files are located in the same directory as the corresponding SVG file.

## svgsubdir/svgsubpath

Within the folder of the encountered SVG file, all extracted and converted files will be located in a subfolder named svg-extract/.

# basedir/basepath/jobdir/jobpath

All extracted and converted files will be located in the current working directory. basesubatr/basesubpath/jobsubdir/jobsubpath

A subfolder named svg-extract/ within the current working directory will be used for all extracted and converted files.

## /path/to/somewhere/

It is also possible to give a custom path, either relative to the current working directory (./relative/path/) or as an absolute path.

 ${\tt extractname}~({\rm opt.})$ 

It's also possible to change the name for extracted and converted files. The default setting is extractname=filenamenumbered. The appended file extension is derived from option extractformat.

#### filename/name

The name of the exported *Inkscape* file is used and the suffix -extract is attached.

#### filenamenumbered/numberedfilename/numberedname

Same as above, but a prefix with the current enumerated count of SVG files is used instead of the suffix.

#### numbered/section/numberedsection/sectionnumbered

The file name is composed by the current enumerated count of SVG files and the present outline numbering.

 $\langle filename \rangle$ 

You can use any file name. It's possible to use counters for specifying the name of the extracted file. Repeatedly specifying the same file name will overwrite previously created files.

extractformat (opt.)

The included SVG file can be extracted from the document into an independent graphic file of type PDF, EPS or PS. The option can be used with either a single value (extractformat=pdf) or a comma separated list. For example,

```
\verb|\cludesvg[extractformat={pdf,eps,ps}]| \{ | svg| filename | \}|
```

will extract the SVG file to both PDF and EPS formats and generates two independent graphic files. By default, extractformat=pdf is set unless DVI output was detected. In this case extractformat=eps is the default setting.

extractwidth (opt.)
extractheight (opt.)
extractdistort (opt.)
extractscale (opt.)
extractpretex (opt.)

extractapptex (opt.)

These options can be used to overwrite the settings given for the appearance of a SVG file within the document. For example, a SVG file should cover the entire text width within the document but be extracted to a fixed width. This can be done with:

```
\verb|\color=| files | f
```

Assigning the value inherit to one of these options—which is set by default—leads to the usage of the corresponding option of package svg (width/height/scale/pretex/apptex), whereas extract...=\relax can be used to ignore a parent option utterly.

 $\label{eq:continuous} \mbox{extractpreamble} \ (\mbox{opt.}) \\ \mbox{extractpreamble} \mbox{end} \ (\mbox{opt.}) \\ \mbox{extractpreamble} \mbox{opt.}$ 

Within the included and extracted SVG files any LaTeX macro can be used either defined by the user—this should be done in the preamble of the LaTeX document in which the SVG file is to be included—or provided by a package which is loaded. As the extraction process of the SVG files needs an auxiliary LaTeX file all used packages and commands have to be known within this file. Consequently, the preamble of the current LaTeX document is used for the extraction of the SVG file by default.

However, it is possible to specify a different preamble file with the option extractpreamble where the file to use as the preamble is given as the argument—including maybe path, but file name and file extension in any case. The given preamble file is searched similar to SVG files meaning, every path given with \svgpath or \graphicspath is examined. The default definition of extractpreamble is \jobname.tex—more precisely the file extension given by option latexext is used—and should suffice for most cases. The preamble up to the line defined by the option extractpreambleend will be used, which is set to a default with \begin{document}.

\svghidepreamblestart \svghidepreambleend

In case, the preamble of the current LATEX document is used, there are maybe packages included or some parts within the preamble, which should not be used within the separate auxiliary LATEX file. These parts can be excluded if they are enclosed by \svghidepreamblestart and \svghidepreambleend.

For example, your current IATEX document uses package **showframe** which causes some problems with the extraction of independent graphic files. So you want to get rid of it within the auxiliary IATEX file. This can be done with:

```
\documentclass{\documentclassname\}
...
\usepackage{svg-extract}
...
\svghidepreamblestart
\usepackage{showframe}
\svghidepreambleend
...
```

extractruns (opt.) When extracting independent graphic files by compiling the generated auxiliary LATEX file, it's maybe necessary to do multiple LATEX iterations on this. The number of iterations is controlled with option extractruns. It is set to extractruns=2 by default.

Iatexexe (opt.)

For the extraction of an independent graphic file, the LATEX program is used which is set by the latexexe option. Depending on the LATEX engine used for the current LATEX document, it is set to either pdflatex, lualatex, xelatex or latex by default. It's also possible to specify additional flags or switches for the LATEX iterations, which are performed during the extraction process by the latexopt option. If you are used to utilize a different extension for LATEX files than .tex, option latexext can be used like latexext=ltx.

dvipsopt (opt.) Depending on the used LATEX engine, the file type of the extracted graphic differs. In order to pstopdfopt (opt.) pstopdfopt (opt.) by most of the LATEX distributions are maybe invoked. These are dvips, ps2eps, ps2pdf and/or pdftopsopt (opt.) pdftopsopt (opt.) every single tool with dvipsopt, pstoppdfopt, pdftoepsopt and pdftopsopt.

clean (opt.) During the extraction process many files are generated for each SVG file extraction. So it's oftentimes desirable to automatically remove these temporary files. Using the option clean=true will remove any generated files created other than the extracted output format(s) requested. Setting clean=false is useful for debugging and set by default. Additionally, it's possible to use option clean with a list of file extensions in order to specify auxiliary files generated by package svg-extract to be deleted, for example clean={log,aux}.

exclude (opt.) Sometimes it may be necessary to extract and/or convert a SVG file without including it. If the flag exclude is specified, the SVG file will not be rendered in the current LATEX document, but will be extracted and/or converted to the requested output format(s).

As previously mentioned, for extracting independent graphic files it is sufficient to load package **svg-extract** and afterwards everything necessary is done by just using \includesvg or \includeinkscape.

With this additional parameter the graphic is rotated during the extraction process. The value is not inherited from angle if it was given by default. This can be achieved by setting:

 $\includesvg[angle=\langle angle \rangle, extractangle=inherit] {\langle svg\ filename \rangle}$ 

# 3.3. Convert extracted graphic files

Based on the extraction of independent graphic files, the **svg-extract** packages also provides the possibility to convert these extracted graphics in another format than PDF, EPS or PS with either *ImageMagick*—which is set by default—or *Ghostscript*.

This option can be used to control the invocation of the conversion process. By default, convert=false is set. For Windows, there exist two different versions of *Ghostscript*, either 64 bit or 32 bit. If it is selected as converting tool the 64 bit executable is set by default. Please note, that option extract has to be activated.

false/off/no

No conversion is done.

true/on/yes

The conversion will be done with the current chosen converting tool.

magick/imagemagick/convert

The conversion is activated and *ImageMagick* is selected.

gs/ghostscript

The conversion is activated and **Ghostscript** is selected.

gs64/ghostscript64

This value activates *Ghostscript* as conversion tool and sets gsexe=gswin64c. On unix-like operating systems, the value for gsexe remains unchanged.

gs32/ghostscript32

The same as for the latter case applies, only option gsexe=gswin32c is set on Windows.

\includesvg \includeinkscape

convert (opt.)

extractangle (param.)

convertformat (opt.)

With this option, the desired output format(s) can be given. Multiple graphic formats can be specified in a list, for example something like convertformat={png,jpg,tif}. The value specified in extractformat is used as the source format for the conversion. If extractformat itself contains a file list, the first value within this list is considered. If extractformat is defined empty, the file generated anyway during the extraction is used.

#### Settings for specific converting formats

Maybe it's desired to apply varying settings for different output formats. Therefor some options described below can either be set for all converted files or for a specific output format. In particular, these are the options convertdpi as well as magicksetting, magickoperator, gsdevice and gsopt. All these mentioned options can be used like either  $\langle option \rangle = \langle value \rangle$  or  $\langle option \rangle = \{\langle outputformat \rangle = \langle value \rangle \}$  where the desired output format is trailed with + as inner key.

The first variant is applied to all output formats in general. If one of these mentioned options is evaluated and an output format specific value was given like in the second variant, the general setting is overwritten. If the general setting should be used and extended by an additional output format specific settings, then the third variant is to be used. In this case, no output format specific setting (second variant) must not have been used.

 ${\tt convertdpi}\ ({\tt opt.})$ 

This option controls the used density for all file formats or a specific one, whether *ImageMagick* or *Ghostscript* is used for the graphic conversion. The desired resolution of the converted file is given in dots per inch (DPI) either as a scalar value (e.g. convertdpi=600) or with different resolutions in x- and y-direction (e.g. convertdpi=600x400).

As described before, it's also possible to declare a specific resolution for each desired converting format. For example, you want to set different resolution for PNG and JPG formats and something for all other formats:

```
\svgsetup{%
  convertdpi={png=600},%
  convertdpi={jpg=150},%
  convertdpi=300%
}%
```

If a setting for a specific output format is given, any unspecific setting is overwritten, when the conversion to this format is done. With convertdpi= $\{\langle outputformat \rangle = \range a specific setting can be reversed.$ 

Please note that not every graphic format support different resolutions in x- and y-direction. So using a value like convertdpi=600x400 may not necessarily lead to the desired result. However, this is then due to the used conversion tool and not to the processing of the option.

# 3.3.1. Settings for the invocation of ImageMagick

magickexe (opt.)
magicksetting (opt.)
magickoperator (opt.)

The conversion with ImageMagick via the magick or convert command line interface can be controlled with these options. The option magickexe determines the used executable and is set to magick on Windows and otherwise to convert by default. Additionally, there are the two options magicksetting and magickoperator which can be used to define settings and operators for the conversion process. As described before, the two options magicksetting and magickoperator can be set for all output formats or a specific one either resetting or extending the general settings. For further information see the documentation of ImageMagick command line interface<sup>6</sup>.

 $<sup>^6 \</sup>verb|http://www.imagemagick.org/script/command-line-processing.php|$ 

#### 3.3.2. Settings for the invocation of Ghostscript

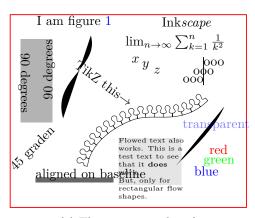
$$\begin{split} & \text{gsexe } (\mathrm{opt.}) \\ & \text{gsdevice } (\mathrm{opt.}) \\ & \text{gsopt } (\mathrm{opt.}) \end{split}$$

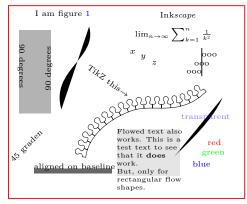
The conversion with *Ghostscript* is done with command line interface <code>gswin64c</code> or <code>gswin32c</code> on Windows and <code>gs</code> on unix-like operating systems. The executable can be changed with option <code>gsexe</code>. Because *Ghostscript* requires the specification of a device, there are some predefined for the most common output formats. These are:

```
\svgsetup{%
gsdevice={png=png16m},gsdevice={jpeg=jpeg},gsdevice={jpg=jpeg},%
gsdevice={tif=tiff48nc},gsdevice={tiff=tiff48nc},%
gsdevice={eps=eps2write},gsdevice={ps=ps2write}%
}%
```

Furthermore, with gsopt additional switches for Ghostscript can be set. As described before, both gsdevice and gsopt can be defined in general or for specific output formats. For further information see the documentation of  $Ghostscript^7$ .

<sup>&</sup>lt;sup>7</sup>https://ghostscript.com/doc/current/Use.htm





(a) This text is too large!

(b) This text fits better.

Figure 1.: An example figure with IATEX support

# 4. Example

As an minimal example<sup>8</sup> take the following lines of code:

```
\documentclass{article}
\usepackage[T1]{fontenc}
\usepackage{svg}
\usepackage[off]{svg-extract}
\svgsetup{clean=true}
\verb|\| \verb| hdf suppresswarning page group=1| \\
\usepackage{relsize}
\usepackage{subcaption}
\begin{document}
\begin{figure}
  \begin{minipage}{\dimexpr\linewidth/2\relax}
    \includesvg[width=\linewidth] {svg-example}%
   \subcaption{This text is too large!}%
  \end{minipage}%
  \begin{minipage}{\dimexpr\linewidth/2\relax}
   \includesvg[width=\linewidth,pretex=\relscale{0.6}]{svg-example}%
   \subcaption{This text fits better.}%
  \end{minipage}
\caption{An example figure with \LaTeX~support\label{fig:example}}%
\end{figure}
\begin{figure}\centering
  \includesvg[%
   width=.5\linewidth,inkscapelatex=false,extractformat={pdf,eps}%
 ]{svg-example}%
  \caption{The same example figure without \LaTeX~support}%
\end{figure}
\end{document}
```

The output is shown in Figure 1 and Figure 2. Within this example the file svg-example.svg was included three times using the \includesvg command.

If you are willing to compile the example, there are two aspects to consider. First, the included SVG file svg-example.svg has to be located in the current folder and is located in  $\langle texmf \rangle / doc/latex/svg/$ . Second, you have to run the desired LATEX engine with flag--shell-escape.

As you can see, Figure 1a is created with default settings, except the width specification. The *Inkscape* export with LaTeX support is done and the extraction of an independent graphic file in PDF format as the **svg-extract** package was loaded.

<sup>&</sup>lt;sup>8</sup>The image used here is a slightly modified version of the image used in the initial documentation on how to include a SVG file in LATEX by Johan B. C. Engelen available as package svg-inkscape on CTAN.

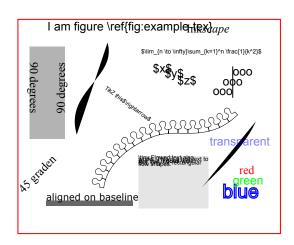


Figure 2.: The same example figure without LATEX support

However, the text is slightly overrunning the margins of the image, and so Figure 1b—which again uses the same *Inkscape* export results—decreases the font size of the text within the image relative using the pretex option together with the \relscale command provided by the **relsize** package.

In Figure 2 the same SVG file was used but without the export of a separate LATEX file containing all text elements.

Feel free to use this given example to try out all the options and possibilities described in section 2 for package svg. Especially if you want to use package svg-extract for the automated extraction of independent graphics (subsection 3.2) and their conversion to different graphic formats with ImageMagick and/or Ghostscript (subsection 3.3), this example can be easily used for the first steps.

# 5. Troubleshooting and reporting issues

When using the packages **svg** and **svg-extract**, the most likely occurring problems will be caused by calling the external programs. For this reason, a short package information is written into the log file right before each call of an external program on shell. If a file should have been created, both packages check after the external call, whether this file exists or not and raise an error or at least a warning, if this file is missing. If you got such a message, please check the log file for lines like:

Package svg Info: or Package svg-extract Info:

Right afterwards, there should appear runsystem(<command>)...excuted. which you should try to execute manually at the terminal in the right directory. In most cases, the problem will be an invalid command call. If something goes wrong during the extraction/converting process of package svg-extract, it would make sense to set option clean=false to not delete any auxiliary files that might be needed.

If you are sure that the problem is not caused by the configuration of your operating system, you can send an error report either via email or create a new issue on GitHub. Both addresses can be found on the title.

#### When using pdfLTFX there are a lot of warnings

It may happen that several warnings like

```
pdfTeX warning: pdflatex.exe(file \langle filename \rangle.pdf): PDF inclusion: multiple pdfs with page group included in a single page
```

occur when including the PDF graphics exported with *Inkscape*. This is related to the handling of transparency effects within PDF files. Since pdfT<sub>E</sub>X version 1.40.15 or later,

you can get rid of these messages by using \pdfsuppresswarningpagegroup=1. See also the discussion on LaTeX Stack Exchange<sup>9</sup> for more information.

# 6. Include SVG files created with ROOT

This section was originally written by Philip Ilten. In the hope that since then nothing has changed fundamentally in the described procedure. This passage remains in the documentation, even if it will almost certainly be relevant to experimental particle physicists only, who frequently use the analysis package ROOT.

ROOT has the ability to export directly to a SVG file, which means that it is possible to completely by-pass all of ROOT's internal text rendering machinery, and let IATEX handle the text natively. This means that all of the ugly fonts that are rendered by ROOT can now be completely avoided, with the additional bonus of being able to add references within plots. So how does one go about using this package with ROOT?

1. Create the plot with **ROOT** as normal, but turn off all IATEX interpretation of text strings. This is a bit tricky, but can be accomplished by setting the font in **ROOT** to a precision of zero as described in the documentation for TAttFill¹0. Remember that the font is set by using the function (TAttFill\*)→SetTextFont(i) with

```
i = (\text{font type}) \times 10 + (\text{font precision})
```

In the following lines of code, a TStyle is defined which sets the font to type "Courier New" with a precision of zero.

```
TStyle *style = new TStyle("style","style"); int FONT = 80;
style->SetTextFont(FONT);
style->SetLabelFont(FONT,"XYZ");
style->SetTitleFont(FONT,"XYZ");
style->SetTitleFont(FONT,"");
gROOT->SetStyle("style");
gROOT->ForceStyle();
```

Now, you can just use the well-known standard  $\LaTeX$  syntax for creating labels, etc. Note however, that backslashes have to be escaped due to interpretation of special characters by C++.

2. Print the plot as a SVG file.

```
gPad->Print("foo.svg");
```

3. Include the SVG file within the document using this package.

```
\usepackage{svg}
\usepackage{svg-extract}
\svgsetup{clean=true}
...
\includesvg[width=\linewidth]{foo}
```

Consider the following example image produced by ROOT in Figure 3. This figure was generated by the ROOT macro root.C, provided within  $\langle \textit{texmf} \rangle / \text{doc/latex/svg/}$ , which produces the file root.svg when run. The code used to produce this SVG file from within ROOT is

```
void root() {

// Set the style.
gStyle->SetTextFont(80); gStyle->SetLabelFont(80,"XYZ");
gStyle->SetTitleFont(80,""); gStyle->SetTitleFont(80,"XYZ");
gStyle->SetPalette(1); gStyle->SetOptStat(0);
```

 $<sup>^{9} \\ \</sup>text{http://tex.stackexchange.com/questions/76273/} \\ ^{10} \\ \text{http://root.cern.ch/root/html/TAttText.html}$ 

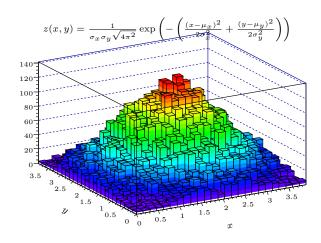


Figure 3.: Rendering of a **ROOT** plot—no more *Comic CERNs* 

```
// Draw the plot.
TH2D *h = new TH2D("", "", 25, 0, 3.9, 25, 0, 3.9); TRandom r;
for (int i = 0; i < 30000; i++) h->Fill(r.Gaus(2.,1), r.Gaus(2.,1));
h->GetXaxis()->CenterTitle(); h->GetXaxis()->SetTitleOffset(2.5);
h->GetYaxis()->CenterTitle(); h->GetYaxis()->SetTitleOffset(2.5);
h->GetXaxis()->SetTitle("\\larger[2]$x$");
h->GetYaxis()->SetTitle("\\larger[2]$y$");
h->Draw("LEGO2");
// Draw additional text.
TText *t = new TText(); t->SetTextAlign(31);
t\rightarrow DrawText(0.7, 0.9, "\larger[2]$z(x,y) = \frac{1}{\sigma_x\sigma_y}"
           "\\right)$");
// Print the plot.
gPad->Print("root.svg");
```

where the text produced within the  ${\it ROOT}$  plot is set to a precision of zero.

The plot was then included within this document using the following LATEX code

```
\begin{figure}
  \centering%
  \includesvg[%
    inkscapearea=page,height=6cm,pretex=\tiny,convertformat=png%
  ]{root}%
  \caption{%
    Rendering of a \app{ROOT} plot---no more \emph{Comic CERNs}%
    \label{fig:root}%
}%
\end{figure}
```

which includes the graphic as well as the LATEX file exported by Inkscape, produces the extracted PDF image (root.pdf) and converts this to a PNG image (root.png) by using ImageMagick. Enjoy plots from ROOT with natively rendered LATEX!

# Part II. Implementation

# A. Initialization

# A.1. Packages

The package **svg** mainly requires **scrbase** for options processing and **graphicx** for the inclusion of the exported graphic files.

The packages **iftex** and **pdftexcmds** are needed to detect the used IATEX engine on the one hand and enabling pdfTEX primitives independent of the used IATEX engine on the other hand. Additionally, **trimspaces** is responsible for string manipulation. Both packages **shellesc** and **ifplatform** are used for engine independent access to systems commands and files. The package **svg-extract** only needs package **svg** itself, which is loaded during initialization.

```
1 \( \*main \)
2 \RequirePackage{iftex} [2020/03/06]
3 \RequirePackage{scrbase} [2020/09/21]
4 \RequirePackage{pdftexcmds} [2019/11/24]
5 \RequirePackage{trimspaces} [2009/09/17]
6 \RequirePackage{graphicx} [2019/11/30]
7 \RequirePackage{shellesc} [2019/11/08]
```

In order to do not raise a warning, package **ifplatform** is only used if **--shell-escape** flag is enabled.

```
8 \ifnum\pdf@shellescape=\@ne\relax
9 \RequirePackage{ifplatform}[2017/10/13]
10 \fi
11 \langle/main\rangle
```

# A.2. Dealing with catcodes

The catcode for double quotes are temporarily changed and restored at the very end of both packages.

```
12 \edef\svg@catcodecodes@restore{%
13 \catcode'\noexpand\"\the\catcode'\"\relax%
14 }
15 \@makeother\"%
16 \AtEndOfPackage{\svg@catcodecodes@restore}
```

# A.3. General macros

\svg@tempa \svg@tempb \if@svg@tempswa Internal temporary macros.

```
17 \newcommand*\svg@tempa{}
18 \newcommand*\svg@tempb{}
```

19 \newif\if@svg@tempswa

#### A.3.1. Macros for process control

\svg@ifwindowsdetected Do some Windows specific stuff if it was detected.

```
20 \newcommand*\svg@ifwindowsdetected{\@secondoftwo}
21 \AfterPackage*{ifplatform}{%
    \renewcommand*\svg@ifwindowsdetected{%
23
      \ifwindows%
        \expandafter\@firstoftwo%
24
25
      \else%
        \expandafter\@secondoftwo%
26
27
      \fi%
    }%
28
29 }
```

\svg@ifvalueisrelax

For some keys the usage of \relax as a value should lead to a special reaction, such as restoring to default behavior or resetting the key. Therefore, \svg@ifvalueisrelax checks, whether \relax was used as value or not.

```
30 \newcommand*\svg@ifvalueisrelax[1]{%
    \begingroup%
31
32
       \def\svg@tempa{#1}%
       \def\svg@tempb{\relax}%
33
      \ifx\svg@tempa\svg@tempb%
34
        \aftergroup\@firstoftwo%
35
36
37
        \aftergroup\@secondoftwo%
38
       \fi%
    \endgroup%
39
40 }
```

\svgx@ifkeyandval

It is checked whether a key was given as  $\langle key \rangle = \langle value \rangle$  or like  $\langle key \rangle = \{\langle format \rangle = \langle value \rangle\}$ .

```
41 \newcommand*\svgx@@ifkeyandval{}
42 \newcommand*\svgx@ifkeyandval[3]{%
43 \def\svgx@@ifkeyandval##1=##2=##3\@nil{\IfArgIsEmpty{##3}{#3}{#2}}%
44 \svgx@@ifkeyandval#1==\@nil%
45 }
```

# A.3.2. String manipulation

Both packages **svg** and **svg-extract** should be able to handle user-defined input and output paths. As there is the possibility for users to provide paths with or without quotes to LATEX, this is taken into account.

\svg@deactivate@dq

In order to avoid errors concerning file names with package **babel** and its active double quotes, this command is defined.

```
46 \newcommand*\svg@deactivate@dq{}
47 \AfterAtEndOfPackage*{babel}-{%
48 \renewcommand*\svg@deactivate@dq{\bbl@deactivate{"}}%
49 \providecommand*\bbl@deactivate[1]-{}%
50 }
```

\svg@sanitize@dq

Save expansion of the second argument in the macro from the first argument with deactivated double quotes.

```
51 \newcommand*\svg@sanitize@dq[2]{%
52 \begingroup%
53 \svg@deactivate@dq%
54 \edef\svg@tempa{\endgroup\def\noexpand#1{#2}}%
55 \svg@tempa%
56 }
```

\svg@quotes@check \svg@quotes@@check \if@svg@quotes@found

During the treatment of paths, it may be necessary to temporarily remove quotes and, if required, add them again later. For this purpose, the switch \if@svg@quotes@found as well as the commands \svg@quotes@check and \svg@quotes@check, which controls the switch, are defined. As before, the string is passed in a macro to \svg@quotes@check.

```
57 \newif\if@svg@quotes@found
58 \newcommand*\svg@quotes@check[1]{%
59 \expandafter\svg@quotes@check#1"\@nil%
60 }
61 \newcommand*\svg@quotes@check{}
62 \def\svg@quotes@check#1"#2\@nil{%
63 \IfArgIsEmpty{#2}{\@svg@quotes@foundfalse}{\@svg@quotes@foundtrue}%
64 }
```

\svg@quotes@remove \svg@quotes@@remove These two commands are used to remove all occurring quotes within a string. The only argument passed to \svg@quotes@remove is not the string itself but a macro in which a string is stored.

```
65 \newcommand*\svg@quotes@remove[2][]{%
    \begingroup%
66
      \IfArgIsEmpty{#1}{\def\svg@tempb{#2}}{\def\svg@tempb{#1}}%
67
      \svg@sanitize@dq\svg@tempa{\svg@tempb}%
68
69
      \expandafter\svg@quotes@check\expandafter{\svg@tempa}%
70
      \expandafter\svg@quotes@@remove\svg@tempa""\@nil%
71
      \edef\svg@tempb{%
72
         \endgroup%
        \def\noexpand#2{\svg@tempa}%
73
        \if@svg@quotes@found%
74
           \noexpand\@svg@quotes@foundtrue%
75
76
          \noexpand\@svg@quotes@foundfalse%
77
78
        \fi%
      }%
79
80
    \svg@tempb%
81 }
82 \newcommand*\svg@quotes@@remove{}
83 \def\svg@quotes@@remove#1"#2"#3\@nil{%
84
    \IfArgIsEmpty{#2}{%
85
      \edef\svg@tempa{#1}%
    }{%
86
      \svg@quotes@@remove#1#2#3""\@nil%
87
    }%
88
89 }
```

\svg@remove@leadingchar

This command removes the single character in given with the first argument from the expanded macro in the second argument.

```
90 \newcommand*\svg@remove@leadingchar[2]{%
91
     \begingroup%
       \svg@sanitize@dq\svg@tempa{#2}%
92
93
       \def\svg@tempb{%
         \def\svg@tempa###1\@nil{\def\svg@tempa{###1}}%
94
         \kernel@ifnextchar#1%
95
96
           {\expandafter\svg@tempa\@gobble}%
97
           {\svg@tempa}%
98
       \expandafter\svg@tempb\svg@tempa\@nil%
99
       \edef\svg@tempb{%
100
         \endgroup%
101
         \def\noexpand#2{\svg@tempa}%
102
103
104
     \svg@tempb%
105 }
```

#### A.3.3. File handling

\svg@filename@parse

As the internal LATEX command \filename@parse is not able to split a given file name containing quotes, \svg@filename@parse is defined to resolve this problem. The optional argument can be used to give a specific file extension, which should be searched within \filename@ext. If found at the very end, the previous part is appended to \filename@base.

```
106 \newcommand*\svg@filename@parse[2][]{% 107 \begingroup%
```

The given path and file is parsed with \filename@parse.

```
\svg@sanitize@dq\svg@tempa{#2}%
       \expandafter\filename@parse\expandafter{\svg@tempa}%
109
110 % If there are quotes in the file path, the closing one will be found as first
111 % character in \cs{filename@base} as \cs{filename@area} is splitted at the last
112 % slash. This leading quote is removed from \cs{filename@base} with
113 % \cs{svg@remove@leadingchar}.
114 %
        \begin{macrocode}
       \svg@quotes@remove{\filename@area}%
115
       \if@svg@quotes@found%
116
         \edef\filename@area{"\filename@area"}%
117
         \verb|\svg@remove@leadingchar"\filename@base%| \\
118
119
       \fi%
```

The found extension is parsed against the optional argument. If a double quote was found within the extension, it actually belongs to \filename@base.

```
120 \ifx\filename@ext\relax\else%
121 \svg@quotes@remove{\filename@ext}%
122 \svg@extension@parse{#1}%
123 \if@svg@quotes@found%
124 \edef\filename@base{\filename@base"}%
125 \fi%
126 \fi%
```

Quotes within \filename@base are normalized.

```
127 \svg@quotes@remove{\filename@base}%

128 \if@svg@quotes@found%

129 \edef\filename@base{"\filename@base"}%

130 \fi%
```

With \svg@tempa the group is closed and the results are saved in the macros \filename@....

```
\edef\svg@tempa{%
131
         \endgroup%
132
133
         \def\noexpand\filename@area{\filename@area}%
134
         \def\noexpand\filename@base{\filename@base}%
          \ifx\filename@ext\relax%
135
            \let\noexpand\filename@ext\noexpand\relax%
136
137
          \else%
138
            \def\noexpand\filename@ext{\filename@ext}%
139
         \fi%
       }%
140
141
     \svg@tempa%
142 }
```

\svg@extension@parse \svg@extension@@parse These macros are used to permit multiple dots in file names. The content of \filename@ext is split at each occurrence of . and the trailing part is compared against the content of the argument of \svg@extension@parse, which is probably \svg@file@ext. If they are equal, the previous part is appended to \filename@base and \filename@ext is set to the content of the first argument.

```
145 \@expandtwoargs\Ifstr%
146 {\detokenize\expandafter{\filename@ext}}{\detokenize\expandafter{#1}}{}%
147 \begingroup%
```

Macro \svg@tempa is used to temporarily store anything before the searched extension at the end of \filename@ext and \svg@tempb is set to the actual searched extension if found.

```
\edef\svg@tempa{%
148
              \def\noexpand\svg@tempa{}%
149
150
              \let\noexpand\svg@tempb\relax%
151
              \noexpand\svg@extension@@parse%
152
                \filename@ext.\noexpand\@nil#1\noexpand\@nil%
           }%
153
            \svg@tempa%
154
            \edef\svg@tempa{%
155
              \endgroup%
156
```

If the trailing extension was found, \filename@base and \filename@ext are adopted.

```
\def\noexpand\filename@base{\filename@base\svg@tempa}%
157
              \ifx\svg@tempb\relax%
158
                \let\noexpand\filename@ext\relax%
159
              \else%
160
161
                \def\noexpand\filename@ext{\svg@tempb}%
162
              \fi%
           }%
163
164
          \svg@tempa%
165
       }%
     }%
166
167 }
```

Macro \svg@extension@@parse is recursively called as long as there are any dots or the searched extension is found.

```
168 \newcommand*\svg@extension@@parse{}
169 \def\svg@extension@@parse#1.#2\@nil#3\@nil{%
170 \edef\svg@tempa{\svg@tempa.#1}%
171 \IfArgIsEmpty{#2}{}{%
172 \Ifstr{\detokenize{#2}}{\detokenize{#3.}}{%
```

If the trailing extension is found, \svg@tempb is defined.

```
173 \edef\svg@tempb{#3}%
174 \}{%
175 \svg@extension@@parse#2\@nil#3\@nil%
176 \}%
177 \}%
178 \
```

\svg@iffilenewer

The macro \svg@iffilenewer is used to decide, whether the export with *Inkscape* is necessary due to an updated SVG file. This can only be done, if \pdf@filemoddate or \filemoddate is defined.

```
179 \newcommand*\svg@iffilenewer[2]{\@gobbletwo}
180 \ifx\pdf@filemoddate\@undefined
     \ifx\filemoddate\@undefined\else
181
       \ifx\strcmp\@undefined\else
182
         \renewcommand*\svg@iffilenewer[2]{%
183
           \begingroup%
184
              \edef\svg@tempa{\filemoddate{#1}}%
185
             \edef\svg@tempb{\filemoddate{#2}}%
186
             \ifnum\strcmp{\svg@tempa}{\svg@tempb}>\z@\relax%
187
188
                \aftergroup\@firstoftwo%
              \else%
189
                \aftergroup\@secondoftwo%
190
191
             \fi%
```

```
192
            \endgroup%
193
         }%
194
       \fi
     \fi
195
196 \else
197
     \ifx\pdf@strcmp\@undefined\else
        \renewcommand*\svg@iffilenewer[2]{%
198
199
         \begingroup%
            \edef\svg@tempa{\pdf@filemoddate{#1}}%
200
            \edef\svg@tempb{\pdf@filemoddate{#2}}%
201
            \ifnum\pdf@strcmp{\svg@tempa}{\svg@tempb}>\z@\relax%
202
              \aftergroup\@firstoftwo%
203
204
            \else%
205
              \aftergroup\@secondoftwo%
206
            \fi%
207
          \endgroup%
208
       }%
     \fi
209
210 \fi
```

\svg@shell@mkdir \svg@shell@mkdir \svg@shell@mv

\svg@shell@@mv

\svg@shell@rm \svg@shell@@rm Finally, platform dependent macros for creating directories as well as moving and deleting files are provided.

```
211 \newcommand*\svg@shell@mkdir[1]{%
212 \begingroup%
```

A directory should only be created, if it isn't the current working directory.

```
\svg@quotes@remove[{#1}]{\svg@tempa}%
213
       \@svg@tempswatrue%
214
215
       \Ifstr{\svg@tempa}{}{\@svg@tempswafalse}{%
216
       \Ifstr{\svg@tempa}{./}{\@svg@tempswafalse}{%
217
       }}%
218
       \if@svg@tempswa%
         \ShellEscape{\svg@shell@@mkdir{\svg@tempa}}%
219
220
       \fi%
221
     \endgroup%
222 }
223 \newcommand*\svg@shell@mv[2]{%
     \ShellEscape{\svg@shell@@mv\space"#1"\space"#2"}%
225 }
226 \newcommand*\svg@shell@rm[1]{%
227
     \ShellEscape{\svg@shell@@rm\space"#1"}%
228 }
```

The platform dependent commands for file access.

```
229 \svg@ifwindowsdetected{%
230 \newcommand*\svg@shell@@mkdir[1]{if not exist "#1" mkdir "#1"}%
231 \newcommand*\svg@shell@@mv{move}%
232 \newcommand*\svg@shell@@rm{del}%
233 }{%
234 \newcommand*\svg@shell@@mkdir[1]{mkdir -p "#1"}%
235 \newcommand*\svg@shell@@mv{mv}%
236 \newcommand*\svg@shell@@rm{rm}%
237 }
```

\svg@normalize@path \svg@normalize@@path If any path is given, a trailing slash is needed. These two macros ensure that this condition is fulfilled in any case, even if this is not considered by the user. As before, a macro containing the path string is passed to \svg@normalize@path.

```
238 \newcommand*\svg@normalize@path[1]{%
239 \begingroup%
240 \svg@quotes@remove[{#1}]{\svg@tempa}%
241 \ifx\svg@tempa\@empty\relax%
```

```
242
        \def\svg@tempa{./}%
243
244
       \expandafter\svg@normalize@@path\svg@tempa//\@nil%
245
       \edef\svg@tempb{%
        \endgroup%
246
247
        \if@svg@quotes@found%
248
          \def\noexpand#1{"\svg@tempa"}%
249
          \def\noexpand#1{\svg@tempa}%
250
        \fi%
251
      ጉ%
252
253
     \svg@tempb%
254 }
255 \newcommand*\svg@normalize@@path{}
256 \def\svg@normalize@@path#1/#2/\@nil{%
    \IfArgIsEmpty{#2}{%
258
       259
       \svg@normalize@@path#2/\@nil%
260
       \edef\svg@tempa{#1/\svg@tempa}%
261
    }%
262
263 }
```

#### A.3.4. List handling

\svgx@ifinlist

Check, if the first argument is included in a comma-separated list in the second argument. Keep in mind that the first argument is not expanded at all, the second one exactly once.

```
264 \newcommand*\svgx@ifinlist[2]{%
265
     \begingroup%
        \def\svg@tempa##1,#1,##2\@nil{%
266
          \IfArgIsEmpty{##2}{%
267
            \aftergroup\@secondoftwo%
268
269
          }{%
            \aftergroup\@firstoftwo%
270
          }%
271
272
273
        \expandafter\svg@tempa\expandafter,#2,#1,\@nil%
274
     \endgroup%
275 }
```

# B. Including SVG files with package svg

# **B.1.** Options

All options, which are recommended to be set with  $\operatorname{svgsetup}\{\langle options\rangle\}\$  but are also available as package options, as well as the optional parameters for both user commands  $\operatorname{locludesvg}[\langle parameters\rangle]\{\langle svg\ file\rangle\}\$  and  $\operatorname{locludeinkscape}[\langle parameters\rangle]\{\langle file\rangle\}\$  are defined with the interface provided by package **scrbase**.

```
276 \DefineFamily{SVG}
277 \DefineFamilyMember{SVG}
```

\svg@deprecated@key

With version v2.00 the whole user-interface was renewed. For reasons of compatibility, outdated options and parameters from version v1.0 are also provided. If an old key was given, a warning is issued and the valid key is used.

```
278 \newcommand*\svg@deprecated@key[3][svg]{%
279 \PackageWarning{#1}{%
280 The option key '#2' is deprecated. \MessageBreak%
281 It's recommended to use '#3'\MessageBreak%
282 instead%
```

Within the exported LATEX files of *Inkscape*, some commands are used out of additional packages. But maybe the user doesn't want to load this packages anyhow.

usexcolor (opt.)
noxcolor (opt.)
\if@svg@use@xcolor
usetransparent (opt.)
notransparent (opt.)
\if@svg@use@transparent

Options for preventing packages **xcolor** and **transparent** to be loaded.

```
286 \newif\if@svg@use@xcolor
287 \FamilyBoolKey{SVG}{usexcolor}{@svg@use@xcolor}
288 \DeclareOption{noxcolor}{\FamilyOptions{SVG}{usexcolor=false}}
289 \newif\if@svg@use@transparent
290 \FamilyBoolKey{SVG}{usetransparent}{@svg@use@transparent}
291 \DeclareOption{notransparent}{\FamilyOptions{SVG}{usetransparent=false}}
```

They are only available during the loading process of package svg.

```
292 \AtEndOfPackage{%
    \RelaxFamilyKey{SVG}{usexcolor}%
293
     \RelaxFamilyKey{SVG}{usetransparent}%
294
     \if@svg@use@xcolor%
295
296
      \RequirePackage{xcolor}[2016/05/11]%
297
     \else%
      \AfterPackage*{xcolor}{%
298
        \PackageWarning{svg}{Package 'xcolor' was loaded anyway}%
299
300
    \fi%
301
302
    \if@svg@use@transparent%
      \RequirePackage{transparent}[2019/11/29]%
303
304
      \AfterPackage*{transparent}{%
305
        306
      }%
307
308
    \fi%
```

There is an issue with package **transparent**, which currently implements an *invalid* check relying on internal commands of package **pgfsys**, whereas these have changed in the latest version.  $^{11}$ 

```
309 \AfterPackage*{transparent}{%

310 \ifcsname Gin@driver\endcsname%

311 \RequirePackage{pgfsys}%

312 \fi%

313 }%

314 }
```

#### B.1.1. The invocation of Inkscape

The Application *Inkscape* is used to create includable graphic files in a desired format (PDF/EPS/PS/PNG) out of files in SVG format, whereas the support of LATEX can optionally be used.

inkscape (opt.)
\svg@ink@mode

The intension of option inkscape is to control the running behaviour of *Inkscape*. It can be switched off at all (inkscape=false) or invoked only if necessary (inkscape=true) and even be forced with every IATEX run (inkscape=forced). Additionally, option inkscape can be used as wrapper for options inkscapeformat, inkscapelatex, inkscapearea and inkscapedpi, which are declared later.

```
315 \newcommand*\svg@ink@mode{}
316 \DefineFamilyKey{SVG}{inkscape}[true]{%
317 \svg@sanitize@dq\svg@tempb{#1}%
```

<sup>11</sup>https://github.com/ho-tex/transparent/issues/3

```
318
     \FamilySetNumerical{SVG}{inkscape}{svg@tempa}{%
319
       {false}{0},{off}{0},{no}{0},%
       {true}{1}, {on}{1}, {yes}{1}, {auto}{1}, {onlynewer}{1}, {newer}{1}, %
320
       {forced}{2},{force}{2},{overwrite}{2},%
321
       {pdf}{3},{PDF}{3},{eps}{4},{EPS}{4},{ps}{5},{PS}{5},{png}{6},{PNG}{6},%
322
323
       {drawing}{7},{crop}{7},%
324
       {page}{8}, {nocrop}{8}, %
       {tex}{9},{latex}{9},{exportlatex}{9},{latexexport}{9},%
325
       {notex}{10}, {nolatex}{10}, {noexportlatex}{10}, {nolatexexport}{10}, %
326
       {latexnoexport}{10},{raw}{10},{plain}{10},{simple}{10}%
327
     }{\svg@tempb}%
328
     \ifx\FamilyKeyState\FamilyKeyStateProcessed%
329
```

Setting the mode for invoking *Inkscape*...

```
330 \ifnum\svg@tempa<\thr@@\relax%
331 \let\svg@ink@mode\svg@tempa%
332 \else%
```

...and the part as wrapper for different options.

```
\ifcase\svg@tempa\relax\or\or\or\ pdf
334
            \FamilyOptions{SVG}{inkscapeformat=pdf}%
335
            \FamilyOptions{SVG}{inkscapeformat=eps}%
336
337
          \or% ps
338
            \FamilyOptions{SVG}{inkscapeformat=ps}%
339
          \or% png
            \FamilyOptions{SVG}{inkscapeformat=png}%
340
341
          \or% drawing
            \FamilyOptions{SVG}{inkscapearea=drawing}%
342
343
          \or% page
344
            \FamilyOptions{SVG}{inkscapearea=page}%
345
          \or% tex
346
            \FamilyOptions{SVG}{inkscapelatex=true}%
347
          \or% notex
348
            \FamilyOptions{SVG}{inkscapelatex=false}%
         \fi%
349
350
       \fi%
```

It's also possible to set the option inkscapedpi by passing a number followed by dpi like inkscape=300dpi.

```
351 \else% dpi
352 \def\svg@tempa##1dpi##2\@nil{%
353 \Ifstr{##2}{dpi}{\FamilyOptions{SVG}{inkscapedpi=##1}}{}%
354 }%
355 \lowercase{\expandafter\svg@tempa\svg@tempb dpi\@nil}%
```

In version v1.0 the option inkscape was used to set both the executable and options for *Inkscape*. This is taken into account here.

```
356 \ifx\FamilyKeyState\FamilyKeyStateProcessed\else% legacy option
```

Splitting executable from options with delimited macros. After calling \svg@tempa with the given value, the part for the executable is stored in \svg@tempa and the option part—which is recognized by the first - character— in \svg@tempb.

```
357
        \svg@quotes@remove[{#1}]{\svg@tempb}%
358
        \IfArgIsEmpty{##2}{\let\svg@tempb\@empty}{%
359
            \def\svg@tempa###1-\@nil{\def\svg@tempb{-###1}}%
360
            \svg@tempa##2\@nil%
361
          }%
362
          \edef\svg@tempa{\trim@spaces{##1}}%
363
364
365
        \edef\svg@tempb{%
```

```
\noexpand\svg@tempa\svg@tempb-\noexpand\@nil%
                       367
                                 }%
                                 \svg@tempb%
                       368
                                 \if@svg@quotes@found%
                       369
                        370
                                   \edef\svg@tempa{"\svg@tempa"}%
                       371
                                 \fi%
                                 \PackageWarning{svg}{%
                       372
                                   Setting the executable%
                       373
                                   \ifx\svg@tempb\@empty\else%
                       374
                                      \space and associated options%
                       375
                                   \fi%
                       376
                                   \MessageBreak%
                        377
                       378
                                   for Inkscape should be done with options\MessageBreak%
                                   'inkscapeexe=\svg@tempa'%
                        380
                                   \ifx\svg@tempb\@empty\else%
                        381
                                      \MessageBreak and 'inkscapeopt=\svg@tempb'%
                        382
                                   \fi.\MessageBreak%
                                   Nevertheless, this was done by now anyway%
                       383
                                 }%
                       384
                                 \edef\svg@tempa{%
                       385
                                   \noexpand\FamilyOptions{SVG}{inkscapeexe=\svg@tempa}%
                       386
                                   \ifx\svg@tempb\@empty\else%
                       387
                                      \noexpand\FamilyOptions{SVG}{inkscapeopt=\svg@tempb}%
                       388
                       389
                                   \fi%
                                 }%
                        390
                        391
                                 \svg@tempa%
                        392
                               \fi%
                       393
                             \fi%
                       394 }
                      Package options which can be used to switch functionality on or off during the loading of
             on (opt.)
                      package svg.
            off (opt.)
                       395 \DeclareOption{on}{\FamilyOptions{SVG}{inkscape=true}}
                       396 \ensuremath{\mbox{\scape=false}}\} \\
                       With these options, the executed command for invoking Inkscape as well as additional
inkscapeversion (opt.)
                       options can be defined.
        \svg@ink@ver
    {\tt inkscapeexe}\;({\rm opt.})
                       397 \newcommand*\svg@ink@ver{\m@ne}
        \svg@ink@exe
                       398 \DefineFamilyKey{SVG}{inkscapeversion}[true]{%
    inkscapeopt (opt.)
                             \FamilySetNumerical{SVG}{inkscape}{svg@tempa}{%
        \svg@ink@opt
                        400
                               {true}{0},{on}{0},{yes}{0},{auto}{0},{detect}{0},{determine}{0},{fetch}{0},%
                        401
                               {enquire}{0}, {identify}{0}, {request}{0}, {retrieve}{0}, {obtain}{0}%
                        402
                             \ifx\FamilyKeyState\FamilyKeyStateProcessed%
                        403
                               \renewcommand*\svg@ink@ver{\m@ne}%
                        404
                        405
                             \else%
                               \def\svg@tempa##1.##2\@nil{%
                        406
                                 \Ifnumber{##1}{%
                        407
                                   \renewcommand*\svg@ink@ver{##1}%
                        408
                                   \FamilyKeyStateProcessed%
                        409
                       410
                                 }{}%
                       411
                               \svg@tempa#1.\@nil%
                       412
                       413
                             \fi%
                       414 }
                       415 \newcommand*\svg@ink@exe{inkscape}
                       416 \DefineFamilyKey{SVG}{inkscapeexe}{%
                             \svg@sanitize@dq\svg@ink@exe{#1}%
                       417
                             \FamilyKeyStateProcessed%
                       418
                       419 }
                       420 \newcommand*\svg@ink@opt{}
                       421 \DefineFamilyKey{SVG}{inkscapeopt}{%
                             \renewcommand*\svg@ink@opt{#1}%
```

366

```
423 \FamilyKeyStateProcessed% 424 }
```

The two options inkscapeversion and inkscapeexe can only be used within the preamble.

```
\def\svg@tempa#1{%
426
     \AtBeginDocument{%
       \DefineFamilyKey[]{SVG}{#1}[]{%
427
         \PackageError{svg}{Option '#1' too late}{%
428
            Option '#1' can only be set within\MessageBreak%
429
            the preamble but you have tried to set it up later.%
430
         }%
431
          \FamilyKeyStateProcessed%
432
433
     }%
434
435 }
436 \svg@tempa{inkscapeexe}
437 \svg@tempa{inkscapeversion}
```

# $\label{eq:continuous} \textbf{inkscapeformat} \ (\text{opt.}) \\ \textbf{\scapeformat} \ \\$

With option inkscapeformat the output format of the *Inkscape* export function, which is called via \ShellEscape, can be configured. It is set to pdf or, if dvi output could be detected, to eps during initialization.

```
438 \newcommand*\svg@ink@format{pdf}
439 \ifxetex\else\ifpdf\else
     \renewcommand*\svg@ink@format{eps}
441 \fi\fi
442 \DefineFamilyKey{SVG}{inkscapeformat}{%
     \FamilySetNumerical{SVG}{inkscapeformat}{svg@tempa}{%
443
       {pdf}{0},{PDF}{0},{eps}{1},{EPS}{1},{ps}{2},{PS}{2},{png}{3},{PNG}{3}%
444
     }{#1}%
445
     \ifx\FamilyKeyState\FamilyKeyStateProcessed%
446
       \ifcase\svg@tempa\relax% latex
447
         \renewcommand*\svg@ink@format{pdf}%
448
449
       \or% eps
450
         \renewcommand*\svg@ink@format{eps}%
451
       \or% ps
         \renewcommand*\svg@ink@format{ps}%
452
453
       \or% png
         \renewcommand*\svg@ink@format{png}%
454
455
       \fi%
456
     \fi%
457 }
```

inkscapelatex (opt.)

latex (opt.)

tex (opt.)
\svg@ink@latex

This option controls whether the Inkscape export will be invoked with or without the generation of a separate  $IAT_EX$  file.

```
458 \newif\if@svg@ink@latex
459 \FamilyBoolKey{SVG}{inkscapelatex}{@svg@ink@latex}
460 \FamilyBoolKey{SVG}{latex}{@svg@ink@latex}
461 \FamilyBoolKey{SVG}{tex}{@svg@ink@latex}
```

# inkscapearea (opt.) \svg@ink@area

The exported area for an *Inkscape* graphic can be set with this option.

```
462 \newcommand*\svg@ink@area{}
463 \DefineFamilyKey{SVG}{inkscapearea}{%
     \FamilySetNumerical{SVG}{inkscapearea}{svg@tempa}{%
464
465
       {drawing}{0},{crop}{0},%
466
       {page}{1}, {nocrop}{1}%
467
     }{#1}%
468
     \ifx\FamilyKeyState\FamilyKeyStateProcessed%
469
       \ifcase\svg@tempa\relax% drawing
         \renewcommand*\svg@ink@area{-D}%
470
       \else% page
471
         \renewcommand*\svg@ink@area{-C}%
472
```

```
473 \fi%
474 \fi%
475 }
```

inkscapedpi (opt.)
inkscapedensity (opt.)
\svg@ink@dpi

A density can be chosen, which is used during export with *Inkscape* for bitmaps and rasterization of filters.

```
476 \newcommand*\svg@ink@dpi{}
477 \let\svg@ink@dpi\relax
478 \DefineFamilyKey{SVG}{inkscapedpi}{%
479
     \FamilyKeyStateUnknownValue%
     \svg@ifvalueisrelax{#1}{%
480
       \let\svg@ink@dpi\relax%
481
       \FamilyKeyStateProcessed%
482
483
       \def\svg@tempa##1dpi##2\@nil{\def\svg@tempa{##1}}%
484
485
       \lowercase{\svg@tempa#1dpi\@nil}%
       \Ifnumber{\svg@tempa}{%
486
         \edef\svg@ink@dpi{\svg@tempa}%
487
         \FamilyKeyStateProcessed%
488
489
       }{}%
490
     }%
491 }
492 \DefineFamilyKey{SVG}{inkscapedensity}{\FamilyOptions{SVG}{inkscapedpi=#1}}
```

\svg@ink@cmd The actual usage of the Inkscape command line interface.

```
493 \newcommand*\svg@ink@cmd[2]{%
     \svg@ink@exe\space"#1.\svg@file@ext"\space\svg@ink@area\space%
494
     \ifx\svg@ink@dpi\relax\else--export-dpi=\svg@ink@dpi\space\fi%
495
     \if@svg@ink@latex--export-latex\space\fi%
497
     \ifx\svg@ink@opt\@empty\else\svg@ink@opt\space\fi%
     \ifcase\svg@ink@ver\relax% 0.x detected
498
499
       --without-gui\space%
       --export-\svg@ink@format="#2.\svg@ink@format"%
500
     \else% 1.x or nothing detected
501
       --export-filename="#2.\svg@ink@format"%
502
503
     \fi%
504 }
```

# B.1.2. Setting input folder and file

svgpath (opt.) In version v1.0 setting the path to SVG files was done via option. So this method is provided as well.

```
505 \DefineFamilyKey{SVG}{svgpath}{%
506
     \PackageWarning{svg}{%
       The key 'svgpath' is deprecated. It's recommended\MessageBreak%
507
       to use '\string\svgpath' instead%
508
     }%
509
     \ifx\svgpath\@undefined%
510
       \AtEndOfPackage{\svgpath{{#1}}}%
511
     \else%
512
       \svgpath{{#1}}%
513
514
     \fi%
515
     \FamilyKeyStateProcessed%
516 }
```

```
svgextension (opt.)
  extension (opt.)
        ext (opt.)
  \svg@file@ext
```

This option modifies the expected extension for the input file which is exported with *Inkscape*. It is set to svg by default.

```
517 \newcommand*\svg@file@ext{svg}
518 \DefineFamilyKey{SVG}{svgextension}{%
```

The extension should be in lower case letters.

```
519 \lowercase{\svg@quotes@remove[{#1}]{\svg@file@ext}}%
```

Remove leading dots from the extension.

#### B.1.3. Setting output folder and file

inkscapepath (opt.)
\svg@out@path

The option inkscapepath controls, in which folder the results of the *Inkscape* export will be located.

```
524 \newcommand*\svg@out@path{}
525 \DefineFamilyKey{SVG}{inkscapepath}{%
     \svg@sanitize@dq\svg@tempb{#1}%
527
     \FamilySetNumerical{SVG}{inkscapepath}{svg@tempa}{%
       {svgpath}{0},{svgdir}{0},%
528
       {svgsubpath}{1},{svgsubdir}{1},%
529
       {basepath}{2}, {basedir}{2}, {jobpath}{2}, {jobdir}{2}, %
530
       {basesubpath}{3}, {basesubdir}{3}, {jobsubpath}{3}, {jobsubdir}{3}%
531
     }{\svg@tempb}%
532
533
     \ifx\FamilyKeyState\FamilyKeyStateProcessed%
534
       \ifcase\svg@tempa\relax% svgpath
         \renewcommand*\svg@out@path{\svg@file@path}%
535
536
       \or% svgsubpath
537
         \renewcommand*\svg@out@path{\svg@file@path svg-inkscape/}%
       \or% basepath
538
         \renewcommand*\svg@out@path{./}%
539
540
       \or% basesubpath
         \renewcommand*\svg@out@path{./svg-inkscape/}%
541
542
       \fi%
     \else%
543
       \edef\svg@out@path{\svg@tempb}%
544
545
       \svg@normalize@path{\svg@out@path}%
546
       \FamilyKeyStateProcessed%
547
     \fi%
548 }
```

inkscapename (opt.)
 \svg@out@name
 \svg@out@base

With option inkscapename the name of the exported file can be changed.

```
549 \newcommand*\svg@out@name{\svg@file@name\svg@file@suffix}
550 \newcommand*\svg@out@base{\svg@out@path\svg@out@name.\svg@ink@format}
551 \DefineFamilyKey{SVG}{inkscapename}{%
552 \renewcommand*\svg@out@name{#1\svg@file@suffix}%
553 \FamilyKeyStateProcessed%
554 }
```

#### B.1.4. Options for the inclusion of graphics

After the graphic export with *Inkscape*, the inclusion of those graphics can be controlled with the following options.

```
width (opt.)
\svg@param@width
height (opt.)
\svg@param@width
distort (opt.)
keepaspectratio (opt.)
\if@svg@param@distort
scale (opt.)
\svg@param@scale
```

These options determine the size of the included graphics. The usage of \relax as value resets the respective option to the default behavior.

```
555 \newcommand*\svg@param@width{\z@}
556 \DefineFamilyKey{SVG}{width}{%
557 \FamilyKeyStateUnknownValue%
558 \svg@ifvalueisrelax{#1}{%
```

```
560
                             \FamilyKeyStateProcessed%
                     561
                          }{%
                             \FamilySetLengthMacro{SVG}{width}{\svg@param@width}{#1}%
                     562
                             \ifx\FamilyKeyState\FamilyKeyStateProcessed%
                     563
                     564
                              \ifdim\svg@param@width<\z@\relax%
                     565
                                 \FamilyKeyStateUnknownValue%
                     566
                               \fi%
                             \fi%
                     567
                          }%
                     568
                     569 }
                     570 \newcommand*\svg@param@height{\z@}
                     571 \DefineFamilyKey{SVG}{height}{%
                          \FamilyKeyStateUnknownValue%
                     573
                          \svg@ifvalueisrelax{#1}{%
                     574
                             \renewcommand*\svg@param@height{\z@}%
                     575
                             \FamilyKeyStateProcessed%
                          }{%
                     576
                             \FamilySetLengthMacro{SVG}{height}{\svg@param@height}{#1}%
                     577
                             \ifx\FamilyKeyState\FamilyKeyStateProcessed%
                     578
                               \ifdim\svg@param@height<\z@\relax%
                     579
                                 \FamilyKeyStateUnknownValue%
                     580
                              \fi%
                     581
                     582
                            \fi%
                          }%
                     583
                     584 }
                     585 \newif\if@svg@param@distort
                     586 \FamilyBoolKey{SVG}{distort}{@svg@param@distort}
                     587 \DefineFamilyKey{SVG}{keepaspectratio}[true]{%
                          \FamilySetBool{SVG}{keepaspectratio}{@svg@tempswa}{#1}%
                     588
                          \ifx\FamilyKeyState\FamilyKeyStateProcessed%
                     589
                             \if@svg@tempswa%
                     590
                               \FamilyExecuteOptions[.svg.sty]{SVG}{distort=false}%
                     591
                     592
                             \else%
                               \FamilyExecuteOptions[.svg.sty]{SVG}{distort=true}%
                     594
                     595
                          \fi%
                     596 }
                     597 \newcommand*\svg@param@scale{1}
                        \DefineFamilyKey{SVG}{scale}{%
                     598
                          \FamilyKeyStateUnknownValue%
                     599
                     600
                          \svg@ifvalueisrelax{#1}{%
                             \renewcommand*\svg@param@scale{1}%
                     601
                     602
                             \FamilyKeyStateProcessed%
                     603
                             \Ifisdimension{#1\p@}{%}
                     604
                     605
                              \ifdim\dimexpr#1\p@\relax>\z@\relax%
                     606
                                 \renewcommand*\svg@param@scale{#1}%
                     607
                                 \FamilyKeyStateProcessed%
                     608
                               \fi%
                            }{}%
                     609
                          }%
                     610
                     611 }
                    For executing code right before or after the graphic inclusion, two hooks are defined.
      pretex (opt.)
\svg@param@pretex
                     612 \newcommand*\svg@param@pretex{}
      apptex (opt.)
                     613 \let\svg@param@pretex\relax
\svg@param@apptex
                     614 \DefineFamilyKey{SVG}{pretex}{%
      postex (opt.)
                     615
                          \svg@ifvalueisrelax{#1}{%
                             \let\svg@param@pretex\relax%
                     616
                          }{%
                     617
                             \def\svg@param@pretex{#1}%
                     618
                          ጉ%
                     619
                     620
                          \FamilyKeyStateProcessed%
```

\renewcommand\*\svg@param@width{\z@}%

559

```
621 }
622 \newcommand*\svg@param@apptex{}
623 \let\svg@param@apptex\relax
624 \DefineFamilyKey{SVG}{apptex}{%
     \svg@ifvalueisrelax{#1}{%
       \let\svg@param@apptex\relax%
626
627
628
       \def\svg@param@apptex{#1}%
     }%
629
     \FamilyKeyStateProcessed%
630
631 }
632 \DefineFamilyKey{SVG}{postex}{%
     \svg@deprecated@key{postex=#1}{apptex=#1}%
633
634 }
```

lastpage (opt.) svg@param@lastpage (counter)

For Inkscape 0.91 a bug concerning the IATEX export has been reported (https://bugs.launchpad.net/ubuntu/+source/inkscape/+bug/1417470). Sometimes the IATEX file created by Inkscape tries to include more pages than actually are present in the PDF file. To work around this problem, a patch is provided. For this purpose, the total page number is read from the PDF file.

```
635 \newcounter{svg@param@lastpage}
636 \DefineFamilyKey{SVG}{lastpage}[true]{%
      \FamilySetNumerical{SVG}{lastpage}{svg@tempa}{%
637
        {false}{0},{off}{0},{no}{0},{ignore}{0},%
638
        {true}{1}, {on}{1}, {yes}{1}, {auto}{1}%
639
640
641
     \ifx\FamilyKeyState\FamilyKeyStateProcessed%
642
        \ifcase\svg@tempa\relax% false
          \label{lastpage} $$\operatorname{SVG}_{1astpage}_{svg@param@lastpage}_{m@ne}\%$ $$
643
644
          \FamilySetCounter{SVG}{lastpage}{svg@param@lastpage}{\z@}%
645
646
        \fi%
647
     \fi%
648 }
```

$$\label{eq:draft} \begin{split} & \texttt{draft} \ (\mathrm{opt.}) \\ & \texttt{\label{eq:draft}} \end{split}$$

The option draft has the same effect as the eponymous option of package graphicx.

```
649 \newif\if@svg@draft
650 \FamilyBoolKey{SVG}{draft}{@svg@draft}
651 \AfterPackage*{graphicx}{\ifGin@draft\@svg@drafttrue\fi}
```

# **B.2.** User commands

### B.2.1. Optional parameters for user commands

The family member is defined for both svg and svg-extract.

```
652 \langle *package \& body \rangle 653 \DefineFamilyMember[.param] {SVG} 654 \langle /package \& body \rangle
```

\svg@local@param@def \svg@local@param@use \svg@local@param@set Most of the package options can also be used as optional parameters for \includesvg or \includeinkscape. Some of them are overloaded for the usage as optional argument and there are some keys, which *only* can be used as optional parameters. This is realized in such a way that \svg@local@param@use is extended with \svg@local@param@def by the definition of local keys during the loading of package svg.

```
655 \newcommand*\svg@local@param@use{}
656 \newcommand*\svg@local@param@def[1]{%
657 \edef\svg@local@param@use{%
658 \unexpanded\expandafter{\svg@local@param@use}\unexpanded{#1}%
659 }%
```

```
660 }
661 \newcommand*\svg@local@param@set[1]{%
662 \svg@local@param@use%
663 \FamilyOptions{SVG}{#1}%
```

As \svg@local@param@set is always used in a local group, it is possible to set inkscapelatex to false, if the output format was set to png with option inkscapeformat.

Using distort=true is only reasonable, if height and width are given.

```
665 \@svg@tempswatrue%
666 \ifdim\svg@param@width>\z@\relax\ifdim\svg@param@height>\z@\relax%
667 \@svg@tempswafalse%
668 \fi\fi%
669 \if@svg@tempswa%
670 \FamilyExecuteOptions[.svg.sty]{SVG}{distort=false}%
671 \fi%
672 }
```

\svg@deprecated@param

This macro checks, if \svgwidth or \svgscale are defined. In this case, the given values are passed to the correlating parameters and a warning is raised.

```
673 \newcommand*\svg@deprecated@param{%
674
     \@svg@tempswafalse%
     \ifx\svgwidth\@undefined\else%
675
676
       \edef\svg@tempa{\noexpand\FamilyOptions{SVG}{width=\svgwidth}}%
677
       \svg@tempa%
       \@svg@tempswatrue%
678
679
     \fi%
     \ifx\svgscale\@undefined\else%
680
       \edef\svg@tempa{\noexpand\FamilyOptions{SVG}{scale=\svgscale}}%
681
682
       \svg@tempa%
       \@svg@tempswatrue%
683
684
     \fi%
     \if@svg@tempswa%
685
       \PackageWarning{svg}{%
686
         You should specify the image size with parameters\MessageBreak%
687
          'width' and 'height' or 'scale' instead of using\MessageBreak%
688
689
          '\string\svgscale' or '\string\svgwidth'%
690
       \let\svgwidth\@undefined%
691
       \let\svgscale\@undefined%
692
693
     \fi%
694 }
```

#### B.2.2. Definition of user commands

\svgsetup \setsvg The macro  $\svgsetup{\langle options \rangle}$  can be used to change options after loading packages svg or svg-extract both in preamble and the document body. For compatibility reasons,  $\svgsetsvg$  is also defined.

```
695 \newcommand*\svgsetup{\FamilyOptions{SVG}} 696 \newcommand*\setsvg{\FamilyOptions{SVG}}
```

\svgpath \svg@input@path With \svgpath the user can give several root paths to SVG files in the same way as \graphicspath is used. The only difference is that a missing slash is added at the end of the path, if needed.

```
697 \newcommand*\svg@input@path{}
698 \let\svg@input@path\input@path
699 \newcommand*\svgpath[1]{%
700 \def\svg@tempa##1\@nil{%
```

```
701 \ifx\svg@tempb\bgroup%
702 \def\svg@input@path{#1}%
703 \else%
704 \def\svg@input@path{{#1}}%
705 \fi%
706 }%
707 \futurelet\svg@tempb\svg@tempa#1\@nil%
708 }
```

\includesvg For the inclusion of SVG files the command \includesvg is defined.

```
709 \newcommand*\includesvg[2][]{%
710 \begingroup%
```

Checking for deprecated commands \svgwidth and \svgscale.

711 \svg@deprecated@param%

```
Most of the optional parameters have the same effect as the identically named options.
      {\tt inkscape}\ ({\rm param.})
                         Only parameter lastpage is extended (see below). Moreover, there are some additional
inkscapeformat (param.)
 inkscapelatex (param.)
                         parameters, which can only be used as optional argument for \includesvg (angle and
                         origin) but not as an option. Now all parameters are set in local context (within a group).
  inkscapearea (param.)
   inkscapedpi (param.)
                         712
                                 \svg@local@param@set{#1}%
   inkscapeopt (param.)
  svgextension (param.)
                        The file suffix used by both packages svg and svg-extract.
         width (param.)
        height (param.)
                         713
                                 \if@svg@ink@latex%
       distort (param.)
                          714
                                    \edef\svg@file@suffix{_\svg@file@ext-tex}%
         scale (param.)
                         715
                                 \else%
        pretex (param.)
                         716
                                   \edef\svg@file@suffix{_\svg@file@ext-raw}%
        apptex (param.)
                         717
                                 \fi%
         draft (param.)
                                 \@onelevel@sanitize\svg@file@suffix%
```

Searching all given paths for the relevant SVG file.

```
719 \svg@get@path{#2}{}%
720 \if@svg@file@found%
```

Running the export with  ${\it Inkscape}$  (if necessary) and checking the required files for graphic inclusion.

```
721
         \svg@ink@run%
722
         \IfFileExists{\svg@out@base}{}{%
723
           \@svg@file@foundfalse%
            \svg@file@missing{\svg@out@base}{\svg@file@base.\svg@file@ext}%
724
         }%
725
         \if@svg@ink@latex%
726
           \IfFileExists{\svg@out@base_tex}{}{%
727
              \@svg@file@foundfalse%
728
              \svg@file@missing{\svg@out@base_tex}{\svg@file@base.\svg@file@ext}%
729
           }%
730
731
         \fi%
```

Include the resulting graphic file and maybe extract independent files.

```
732 \if@svg@file@found%
733 \svg@input{\svg@out@base}%
734 \svg@extract{\svg@out@base}%
735 \fi%
736 \else%
```

Raise an error, if the requested SVG file wasn't found.

```
737 \svg@file@missing[\svg@file@ext]{\svg@file@base}{}%

738 \fi%

739 \endgroup%

740}
```

 ${\tt lastpage}\ ({\rm param.})$ 

In addition to the automatic finding of the last page, which is included, it can also be given directly as parameter.

angle (param.)
origin (param.)

The parameters angle and origin are defined as pendants to the keys provided by \includegraphics.

```
744 \newcommand*\svg@param@angle{0}
745 \svg@local@param@def{%
     \DefineFamilyKey[.param]{SVG}{angle}{%
746
       \FamilyKeyStateUnknownValue%
747
       \Ifisdimension{#1\p@}{%}
748
749
         \renewcommand*\svg@param@angle{#1}%
750
         \FamilyKeyStateProcessed%
751
       }{}%
     }%
752
753 }
754 \newcommand*\svg@param@origin{c}
755 \svg@local@param@def{%
     \DefineFamilyKey[.param]{SVG}{origin}[c]{%
       \renewcommand*\svg@param@origin{#1}%
757
       \FamilyKeyStateProcessed%
758
     }%
759
760 }
```

\includeinkscape

The command \includeinkscape can be used for including the export results of *Inkscape*, if this part of the job was done in another way.

```
761 \newcommand*\includeinkscape[2][]{% 762 \begingroup%
```

Checking for deprecated commands \svgwidth and \svgscale.

```
763 \svg@deprecated@param%
```

The given file extension is examined, where a known extension overwrites the current setting for inkscapeformat. If there's a suffix \_tex, the option inkscapelatex is set to true by default.

```
764
       \svg@filename@parse{#2}%
765
       \ifx\filename@ext\relax\else%
766
         \svg@quotes@remove{\filename@ext}%
767
         \expandafter\lowercase\expandafter{%
           \expandafter\def\expandafter\filename@ext\expandafter{\filename@ext}%
768
         }%
769
         \def\svg@tempb##1_tex##2\@nil{%
770
           \IfArgIsEmpty{##1}{}{\def\filename@ext{##1}}%
771
           \Ifstr{##2}{_tex}{\@svg@tempswatrue}{\@svg@tempswafalse}%
772
773
         \@svg@tempswafalse%
774
         775
776
           \begingroup%
             \expandafter\svg@tempb\filename@ext_tex\@nil%
777
             \svg@extension@parse{\svg@tempa}%
778
             \ifx\filename@ext\relax%
779
```

```
780
                \def\svg@tempb{\endgroup}%
              \else%
781
                \edef\svg@tempb{%
782
                  \endgroup%
783
                  \noexpand\FamilyOptions{SVG}{inkscapeformat=\svg@tempa}%
784
                  \if@svg@tempswa%
785
                    \noexpand\FamilyOptions{SVG}{inkscapelatex=true}%
786
787
                  \fi%
                  \def\noexpand\filename@base{\filename@base}%
788
                  \def\noexpand\filename@ext{\filename@ext}%
789
                  \noexpand\@svg@tempswatrue%
790
                }%
791
              \fi%
792
793
            \svg@tempb%
```

Break for loop, if valid extension was found.

```
794 \if@svg@tempswa%
795 \@break@tfor%
796 \fi%
797 }%
```

If no valid extension was found, it is set to the specified format and the actual found one is appended to cssvg.dtx@base.

```
798 \if@svg@tempswa\else%
799 \svg@extension@parse{\svg@ink@format}%
800 \fi%
801 \fi%
```

```
inkscapeformat (param.)
inkscapelatex (param.)
width (param.)
```

Parameters, which are supported by \includesvg, can also be used with \includeinkscape even if some of them—more precisely those that control the export with *Inkscape*—don't have an effect at all. Nevertheless, they are set right now in local context (within a group).

```
height (param.)
distort (param.) 802 \svg@local@param@set{#1}%
```

Searching all given paths for the relevant PDF/EPS file.

```
pretex (param.) Sea apptex (param.) 80 araft (param.) 80 angle (param.) Ch
```

origin (param.)

scale (param.)

803 \svg@get@path[\svg@ink@format]{\filename@area\filename@base}{\svg@out@path}% 804 \if@svg@file@found%

Checking the required files for graphic inclusion.

```
805
         \edef\svg@out@name{\svg@file@name}%
806
         \edef\svg@out@base{\svg@file@path\svg@file@name.\svg@ink@format}%
         \if@svg@ink@latex%
807
           \IfFileExists{\svg@out@base_tex}{}{%
808
809
              \@svg@file@foundfalse%
810
              \svg@file@missing{\svg@out@base_tex}{\svg@out@base}%
           }%
811
         \fi%
812
```

Include the resulting graphic file and maybe extract independent files.

```
813 \if@svg@file@found%

814 \svg@input{\svg@out@base}%

815 \svg@extract{\svg@out@base}%

816 \fi%

817 \else%
```

Raise an error, if the requested PDF/EPS file wasn't found.

```
818 \svg@file@missing[\svg@ink@format]{\svg@file@base}{\svg@out@path}%

819 \fi%

820 \endgroup%

821 }
```

# **B.3.** Auxiliary macros

\svg@ink@run \if@svg@ink@run The command, which performs the call of *Inkscape* via \ShellEscape.

```
822 \newif\if@svg@ink@run

823 \newcommand*\svg@ink@run{%

824 \ifnum\svg@ink@mode>\z@\relax%

825 \begingroup%
```

If the mode for inkscape was set to forced, *Inkscape* will be called in any case. Otherwise, some checks are performed to detect, if a run of *Inkscape* is actually necessary.

```
826 \@svg@ink@runtrue%
827 \ifnum\svg@ink@mode=\tw@\relax\else%
```

This is the case when the SVG file is newer than the corresponding exported file, or if the latter isn't present at all.

```
828 \svg@iffilenewer{\svg@file@base.\svg@file@ext}{\svg@out@base}{}{%
829 \@svg@ink@runfalse%
830 }%
```

The same is true, when the associated LATEX file is missing. But when this file already exists, maybe the user did some changes to this file. In this case, overwriting this file is maybe not intended.

```
831
           \if@svg@ink@latex%
832
              \IfFileExists{\svg@out@base_tex}{%
                \ifnum\pdf@shellescape=\@ne\relax\if@svg@ink@run%
833
834
                  \svg@iffilenewer{\svg@out@base_tex}{\svg@out@base}{%
                    \@svg@ink@runfalse%
835
                    \svg@quotes@remove[\svg@out@base]{\svg@tempa}%
836
                    \PackageWarning{svg}{%
837
                      Since the encountered filedate of file\MessageBreak%
838
                      '\svg@tempa_tex' is newer than \MessageBreak%
839
840
                      '\svg@tempa' it's supposed that\MessageBreak%
841
                      you customized this file. To avoid an accidental\MessageBreak%
842
                      overwriting of this file, the Inkscape export\MessageBreak%
843
                      won't be done. If you want to overwrite the\MessageBreak%
844
                      existing file please choose the parameter\MessageBreak%
                      'inkscape=force'%
845
                    }%
846
                  }{}%
847
                \fi\fi%
848
             }{\@svg@ink@runtrue}%
849
           \fi%
850
851
         \fi%
```

If all checks were positive, the export with *Inkscape* can be done in case flag --shell-escape is used.

```
852 \if@svg@ink@run%
853 \ifnum\pdf@shellescape=\@ne\relax%
```

For exporting PNG files, the used density is set to 300dpi, if no value was given.

```
\ifx\svg@ink@dpi\relax%
854
                \Ifstr{\svg@ink@format}{png}{%
855
                  \FamilyOptions{SVG}{inkscapedpi=300}%
856
                }{}%
857
              \fi%
858
              \PackageInfo{svg}{%
859
                Calling Inkscape%
860
                \ifx\svg@ink@opt\@empty\else%
861
                  \space with added options '\svg@ink@opt'%
862
                \fi%
863
              }%
864
```

Executing *Inkscape* on shell. Afterwards, the export results are moved into the given output path.

```
\svg@quotes@remove[\svg@file@base]{\svg@tempa}%
svg@quotes@remove[\svg@out@name]{\svg@tempb}%
```

The last try to detect the version automatically, if this wasn't successful until now.

```
867 \ifnum\svg@ink@ver=\m@ne\relax%
868 \svg@ink@ver@explore{\svg@tempa}{\svg@tempb}{\svg@out@name}%
869 \fi%
```

Now it's time to actually create the desired graphic.

```
\ShellEscape{\svg@ink@cmd{\svg@tempa}{\svg@tempb}}%
870
871
              \IfFileExists{\svg@out@name.\svg@ink@format}{%
872
                \edef\svg@tempb{\svg@tempb.\svg@ink@format}%
873
                \svg@quotes@remove{\svg@out@base}%
874
                \svg@shell@mkdir{\svg@out@path}%
                \svg@shell@mv{\svg@tempb}{\svg@out@base}%
875
                \if@svg@ink@latex%
876
                  \svg@shell@mv{\svg@tempb_tex}{\svg@out@base_tex}%
877
878
                \fi%
              ጉ{%
879
                \gdef\svg@ink@ver{\m@ne}%
880
                \PackageWarning{svg}{%
881
                  The export with Inkscape failed for file\MessageBreak%
882
                  '\svg@tempa.\svg@file@ext'\MessageBreak%
883
                  Troubleshooting: Please check in the log file how\MessageBreak%
884
                  the invocation of Inkscape took place and try to\MessageBreak%
885
886
                  execute it yourself in the terminal%
887
               }%
             }%
888
```

If --shell-escape wasn't enabled, a warning is issued.

```
\else%
889
              \svg@quotes@remove[\svg@file@base]{\svg@tempa}%
890
              \PackageWarning{svg}{%
891
                You didn't enable 'shell escape' (or 'write18')\MessageBreak%
892
                so it wasn't possible to launch the Inkscape export\MessageBreak%
893
                for '\svg@tempa.\svg@file@ext'%
894
              }%
895
896
            \fi%
         \fi%
897
        \endgroup%
898
     \fi%
899
900 }
```

\svg@input \svg@input \svg@box

With \svg@@input the export results of Inkscape are included. The macro \svg@input is defined in order to realize the option exclude for package svg-extract. The macro \svg@set@input@path is called to support commands like \input{ $\langle tex\ filename \rangle$ } within SVG files.

```
901 \newsavebox\svg@box
902 \newcommand*\svg@input{\svg@@input}
903 \newcommand*\svg@@input[2][]{%
904 \IfArgIsEmpty{#1}{}\svg@local@param@set{#1}}%
905 \svg@set@input@path%
```

If the export with Inkscape was done with IATEX support enabled, the corresponding file will be used together with \input. The necessary patches to environment picture as well as command \includegraphics are made beforehand with \svg@patches.

```
906 \@svg@tempswatrue%
907 \if@svg@draft%
```

```
908
        \@svg@tempswafalse%
909
     \fi%
     \if@svg@ink@latex\else%
910
        \@svg@tempswafalse%
911
912
     \ensuremath{\tt def\svg@tempa{\#2}\%}
913
914
     \if@svg@tempswa%
        \svg@patches{\svg@tempa}%
915
        \ifnum\value{svg@param@lastpage}=\z@\relax%
916
          \expandafter\svg@get@lastpage\expandafter{\svg@tempa}%
917
        \fi%
918
        \edef\svg@tempa{%
919
920
          \ifx\svg@param@pretex\relax\else%
921
            \noexpand\svg@param@pretex%
922
923
          \noexpand\input{\svg@tempa_tex}%
924
          \ifx\svg@param@apptex\relax\else%
925
            \noexpand\svg@param@apptex%
          \fi%
926
       }%
927
```

If distort=true is desired, the input is resized with \resizebox\*.

```
928 \if@svg@param@distort%
929 \def\svg@tempb{\resizebox*{\svg@param@width}{\svg@param@height}}%
930 \else%
931 \let\svg@tempb\@firstofone%
932 \fi%
933 \sbox\svg@box{\svg@tempb{\svg@tempa}}%
```

If a rotation angle was given, the input is done within \rotatebox.

```
934
       \ifdim\dimexpr\svg@param@angle\p@\relax=\z@\relax%
935
         \let\svg@tempb\@firstofone%
936
       \else%
937
         \edef\svg@tempb{%
            \noexpand\rotatebox[origin=\svg@param@origin]{\svg@param@angle}%
938
939
       \fi%
940
       \svg@tempb{\usebox\svg@box}%
941
     \else%
942
```

If the export with *Inkscape* was done without LaTeX support, the resulting graphic file will be included with \includegraphics.

```
\svg@wrn@scale%
943
       \edef\svg@tempb{%
944
         draft\if@svg@draft\else=false\fi,%
945
946
         scale=\svg@param@scale,%
         keepaspectratio\if@svg@param@distort=false\fi%
947
948
       \ifdim\svg@param@height>\z@\relax%
949
         \edef\svg@tempb{\svg@tempb,height=\svg@param@height}%
950
       \fi%
951
       \ifdim\svg@param@width>\z@\relax%
952
953
         \edef\svg@tempb{\svg@tempb,width=\svg@param@width}%
954
       \ifdim\dimexpr\svg@param@angle\p@\relax=\z@\relax\else%
955
         \edef\svg@tempb{%
956
957
            \svg@tempb,origin=\svg@param@origin,angle=\svg@param@angle%
958
         ጉ%
       \fi%
959
       \expandafter\includegraphics\expandafter[\svg@tempb]{\svg@tempa}%
960
961
     \fi%
962 }
```

\svg@wrn@scale The option scale respectively the parameter scale is only considered if the size was not specified.

```
963 \newcommand*\svg@wrn@scale{%
964
     \ifdim\dimexpr\svg@param@scale\p@\relax=\p@\relax\else%
       \@svg@tempswafalse%
965
       \ifdim\svg@param@width>\z@\relax%
966
967
         \@svg@tempswatrue%
968
       \fi%
969
       \ifdim\svg@param@height>\z@\relax%
970
         \@svg@tempswatrue%
971
972
       \if@svg@tempswa%
973
         \PackageWarning{svg}{%
           The parameter 'scale' is only considered if neither\MessageBreak%
974
            'width' nor 'height' are specified%
975
976
         ጉ%
977
       \fi%
978
     \fi%
979 }
```

\svg@get@lastpage

This macro is used to circumvent the multiple pages bug for PDF files of *Inkscape* 0.91, when the LaTeX export was enabled. For this purpose, the total page number is read from the PDF file.

```
980 \newcommand*\svg@get@lastpage[1]{%
      \Ifstr{\svg@ink@format}{pdf}{%
981
982
        \begingroup%
983
          \@tempcnta=\m@ne\relax%
 984
          \ifx\XeTeXpdfpagecount\@undefined%
 985
             \ifpdf%
 986
               \ifx\pdfximage\@undefined%
 987
                 \ifx\saveimageresource\@undefined\else%
988
                   \saveimageresource{#1}%
                   \@tempcnta=\lastsavedimageresourcepages\relax%
989
                 \fi%
990
               \else%
991
                 \pdfximage{#1}%
992
993
                 \@tempcnta=\pdflastximagepages\relax%
               \fi%
994
            \fi%
 995
          \else%
 996
997
            \@tempcnta=\XeTeXpdfpagecount#1\relax%
998
          \fi%
          \ifnum\@tempcnta=\m@ne\relax%
999
1000
            \PackageWarning{svg}{%
               It wasn't possible to detect the last page\MessageBreak%
1001
               of '#1'%
1002
1003
            }%
1004
             \PackageInfo{svg}{Last page of '#1' is \the\@tempcnta}%
1005
          \fi%
1006
          \edef\svg@tempa{%
1007
1008
             \endgroup%
             \noexpand\FamilyOptions{SVG}{lastpage=\the\@tempcnta}%
1009
          }%
1010
1011
        \svg@tempa%
      }{}%
1012
1013 }
```

\svg@file@missing The error message, which is raised, if a file is missing either after the export with *Inkscape* or in general.

```
1014 \newcommand*\svg@file@missing[3][]{% 1015 \begingroup%
```

```
1016
       \svg@quotes@remove[{#2}]{\svg@tempa}%
1017
       \svg@filename@parse[{#1}]{\svg@tempa}%
1018
       \IfArgIsEmpty{#1}{%
         \svg@quotes@remove[{#3}]{\svg@tempb}%
1019
         \def\svg@tempa{%
1020
1021
          Did you run the export with Inkscape? There's no file\MessageBreak%
1022
           '\filename@area\filename@base.\filename@ext'\MessageBreak%
1023
           although '\svg@tempb' was found.%
         }%
1024
       ጉ{%
1025
         \edef\filename@ext{#1}%
1026
         1027
```

Collecting all considered path for the error message.

```
\edef\svg@tempb{#3}%
1028
          \Ifstr{\svg@tempb}{./}{\let\svg@tempb\@empty}{}%
1029
1030
          \ifx\svg@tempb\@empty%
1031
            \svg@set@input@path%
          \else%
1032
            \svg@set@input@path[\svg@tempb]%
1033
1034
          \fi%
          \ifx\input@path\@undefined%
1035
1036
            \def\svg@tempb{No additional path was given.}%
1037
            \def\svg@tempb{Following folders have additionally been searched:}%
1038
1039
            \expandafter\@tfor\expandafter\svg@tempa\expandafter:\expandafter=%
1040
                 \input@path\do{%
              \edef\svg@tempb{\svg@tempb\noexpand\MessageBreak\svg@tempa}%
1041
            ጉ%
1042
          \fi%
1043
```

The error message itself.

```
1044
          \def\svg@tempa{%
1045
            There's no file '\filename@base.\filename@ext'\MessageBreak%
1046
            \ifx\filename@area\@empty%
              neither in the current directory nor any other searched\MessageBreak%
1047
              path given by \string\svgpath\space or \string\graphicspath.%
1048
              \MessageBreak\svg@tempb%
1049
            \else%
1050
              in folder '\filename@area'.%
1051
1052
            \fi%
          }%
1053
1054
        \PackageError{svg}{%
1055
1056
          File '\filename@base.\filename@ext' is missing%
1057
        }{\svg@tempa}%
1058
      \endgroup%
1059 }
```

\svg@ink@ver@settings \svg@ink@ver@detect \svg@@ink@ver@detect \if@svg@ink@ver@detect As the command line interface of *Inkscape* has changed between versions 0.x and 1.x, option inkscapeversion=detect allows to detect the used version of *Inkscape* in order to define the calling macro \svg@ink@cmd. The obtained version is stored in \svg@ink@ver, whereas the following meanings are applied:

- -1 version check has not be done or *Inkscape* could not be found/executed
- 0 Inkscape version 0.x was found
- 1 Inkscape version 1.x or later was found

All necessary information are stored within \svg@ink@ver@settings as three arguments, whereas the first one is the manually set version, the second is the used inkscapeexe for automatic detection and the third one is the detected version itself.

1060 \newcommand\*\svg@ink@ver@settings{{\svg@ink@ver}{\svg@ink@exe}{\m@ne}}

In order to run the check for *Inkscape* version at the beginning of the document only if needed, changes of both inkscapeversion—at this point stored in \svg@ink@ver— as well as inkscapeexe are detected and are triggering the version check. After evaluating the triggers, the current values set are stored as two tokens in \svg@ink@ver@settings. If a check has been triggered, the detected version will be evaluated further on and is stored in the third token of \svg@ink@ver@settings.

```
1062 \newcommand*\svg@ink@ver@detect[3]{%
1063 \@svg@ink@ver@detectfalse%
1064 \ifnum\pdf@shellescape=\@ne\relax%
1065 \ifnum\svg@ink@ver=\m@ne\relax%
```

If inkscapeexe was not changed...

```
1066 \svg@sanitize@dq\svg@tempa{#2}%
1067 \ifx\svg@tempa\svg@ink@exe%
```

...then enforce the check after a change of mode to detect...

```
1068 \ifnum#1>\m@ne\relax%
1069 \@svg@ink@ver@detecttrue%
```

... or if detection was never invoked, do so.

```
1070 \else%
1071 \ifnum#3=\m@ne\relax%
1072 \@svg@ink@ver@detecttrue%
1073 \fi%
1074 \fi%
```

Enforce the check after a change of inkscapeexe.

```
1075 \else%
1076 \@svg@ink@ver@detecttrue%
1077 \fi%
1078 \fi%
1079 \fi%
```

After evaluating the last settings and maybe setting the trigger for version detection, the current settings are stored in the main aux file. The detected version will be expanded during the write to the aux file.

```
1080 \edef\svg@ink@ver@settings{%
1081 {\svg@ink@ver}{\svg@ink@exe}{\noexpand\svg@ink@ver}%
1082 }%
```

Run detection if necessary and store the result in \svg@ink@ver...

```
1083 \if@svg@ink@ver@detect%
1084 \svg@@ink@ver@detect%
1085 \else%
```

...or otherwise set previous detected version in automatic mode.

```
1086 \ifnum\svg@ink@ver=\m@ne\relax%

1087 \def\svg@ink@ver{#3}%

1088 \fi%

1089 \fi%

1090 }
```

If the switch \if@svg@ink@ver@detect was set by \svg@ink@ver@detect during the evaluation of \svg@ink@settings, which holds the settings of the last run. The call of *Inkscape* stored in \svg@ink@exe is done with \@@input|"'...'" -V in order to read from stdout.

```
1091 \newcommand*\svg@@ink@ver@detect{%
1092 \begingroup%
1093 \@makeother\\%
1094 \@makeother\&%
1095 \everyeof{\noexpand}%
1096 \svg@quotes@remove{\svg@ink@exe}%
```

Here stderr is redirected to stdout in order to fetch all information from *Inkscape* (some are passed via stderr [sic]) as well as pass any found error message to the user.

```
1097 \edef\svg@tempa{\\\
1098 \edef\noexpand\svg@tempa{\noexpand\@@input|"'\svg@ink@exe' -V 2>&1" }\\\
1099 }\\\
1100 \svg@tempa\\\\\
1101 \trim@spaces@in\svg@tempa\\\\\
```

The invocation of commands through a pipe is buggy for MiKTEX especially for XeIATEX (https://github.com/MiKTeX/miktex/issues/648). Either stdout or stderr if former not present gets swallowed and stderr or \par is returned...

```
1102 \def\svg@tempb{\par}%
1103 \ifx\svg@tempa\svg@tempb
1104 \let\svg@tempa\relax%
1105 \fi%
```

The found version is stored in \svg@tempa and parsed afterwards. Any other output than the expected format is considered as error message, which is picked up with ##1.

With *Inkscape* 1.0 the additional version information for *Pango* is passed via stderr but stdout is gobbled, so this can be used to indetify the version.

```
1109 %^^A
                 \PackageWarning{svg}{%
1110 %^^A
                   '\svg@ink@exe\space-V' failed with:\MessageBreak%
1111 %^^A
                   \detokenize{##1}%
1112 %^^A
                }%
            \def\svg@tempc ####1Pango version:####2\@nil{%
1113
              \@svg@tempswafalse%
1114
              \IfArgIsEmpty{####2}{}{\IfArgIsEmpty{####1}{\@svg@tempswatrue}{}}%
1115
1116
1117
            \svg@tempc ##1 Pango version:\@nil%
            \if@svg@tempswa%
1118
               \gdef\svg@ink@ver{1}%
1119
1120
            \else%
```

No messages can be passed for MiKTeX/XeLaTeX.

```
\ifx\svg@tempa\relax%
1121
1122
                 \def\svg@tempa{MiKTeX/XeLaTeX did not return an error message}%
1123
               \else%
1124
                 \def\svg@tempa{##1}%
               \fi%
1125
               \PackageWarning{svg}{%
1126
                 '\svg@ink@exe\space-V' failed with:\MessageBreak%
1127
1128
                 \detokenize\expandafter{\svg@tempa}%
              }%
1129
            \fi%
1130
          }%
1131
        }%
1132
        \expandafter\svg@tempb\svg@tempa Inkscape \m@ne.\@nil%
1133
```

```
1134 \endgroup%
1135 }
```

Comparing the stored settings from last the last run with current settings.

1136 \AtBeginDocument{\expandafter\svg@ink@ver@detect\svg@ink@ver@settings}

Writing \svg@ink@exe and \svg@ink@ver to the main aux-file.

```
1137 \BeforeClosingMainAux{%

1138 \if@filesw%

1139 \immediate\write\@mainaux{%

1140 \string\gdef\string\svg@ink@ver@settings{\svg@ink@ver@settings}%

1141 }%

1142 \fi%

1143 }
```

\svg@ink@ver@explore

If detecting the used version automatically with <code>inkscape</code> -V was not successful, it is tried to explore this by calling <code>Inkscape</code> through its command line interface with all known variations. If the desired file was created the used version is stored in \svg@ink@ver.

```
1144 \newcommand*\svg@ink@ver@explore[3]{%
1145 \begingroup%
1146 \@svg@tempswafalse%
1147 \@tfor\svg@ink@ver:={1}{0}\do{%
1148 \ShellEscape{\svg@ink@cmd{#1}{#2}}%
1149 \IffileExists{#3.\svg@ink@format}{\@svg@tempswatrue}{}%
```

An output file was found, break loop.

```
1150 \if@svg@tempswa%
1151 \@break@tfor%
1152 \fi%
1153 }%
```

If even this attempt fails, an error message is shown.

```
\if@svg@tempswa%
1154
          \xdef\svg@ink@ver{\svg@ink@ver}%
1155
1156
        \else%
          \PackageError{svg}{Inkscape version not detected}{%
1157
            It was tried to invoke '\svg@ink@exe'\MessageBreak%
1158
            for file "#1.\svg@file@ext"\MessageBreak%
1159
1160
            but no result was produced. Check the log file\MessageBreak%
1161
            and set 'inkscapeversion='version' manually.%
1162
          ጉ%
1163
        \fi%
1164
      \endgroup%
1165 }
```

# B.4. Handling path and file names

\svg@set@input@path \svg@append@input@path In order to import SVG files from different folders, \svg@set@input@path evaluates several macros, which are supposed to be used for holding different search folders. Any given path will be handled by \svg@normalize@path. The optional argument can be used to append an additional search path.

```
1166 \newcommand*\svg@set@input@path[1][]{%
1167 \begingroup%
1168 \svg@deactivate@dq%
```

If a path was already found and stored within \svg@file@path, it is searched first and wrapped in curly braces. This is necessary for using commands like  $\input{\langle tex\ filename \rangle}$ within SVG files.

```
1169
        \ifx\svg@file@path\@empty\else%
1170
          \svg@normalize@path{\svg@file@path}%
1171
          \edef\svg@file@path{{\svg@file@path}}%
1172
        \fi%
```

Afterwards, several search paths are appended. If \svgpath was used, it is searched next. If nothing was found, \graphicspath is considered if defined followed by a path given in the third argument. If nothing was found yet, the standard \input@path is searched last.

```
\svg@append@input@path{\svg@file@path}{\svg@input@path}%
1173
1174
       \svg@append@input@path{\svg@file@path}{\Ginput@path}%
1175
       \IfArgIsEmpty{#1}{}{\svg@append@input@path{\svg@file@path}{{#1}}}%
       \svg@append@input@path{\svg@file@path}{\input@path}%
1176
Finally, \input@path is set.
       \edef\svg@tempa{%
1177
1178
         \endgroup%
         \ifx\svg@file@path\@empty\else%
1179
           1180
1181
         \fi%
1182
       }%
     \svg@tempa%
```

Only, if a certain search path is defined, it is added. The paths given in the first argument are compared to each path in the second argument and only new ones are added.

```
1185 \newcommand*\svg@append@input@path[2]{%
1186
      \ifx#2\@undefined\else%
        \edef\svg@tempb{#2}%
1187
        \expandafter\@tfor\expandafter\svg@tempa\expandafter:\expandafter=%
1188
            \svg@tempb\do{%
1189
```

Passing each new path to \svg@normalize@path. If a path already exists, switch \if@svg@tempswa is set to false.

```
\ifx\svg@tempa\@empty\else%
1190
             \@svg@tempswatrue%
1191
1192
             \svg@normalize@path{\svg@tempa}%
1193
             \expandafter\@tfor\expandafter\svg@tempb\expandafter:\expandafter=%
1194
                 #1\do{%
               \ifx\svg@tempa\svg@tempb%
1195
                 \@svg@tempswafalse%
1196
                 \@break@tfor%
1197
               \fi%
1198
             }%
1199
1200
             \if@svg@tempswa%
               \edef#1{#1{\svg@tempa}}%
1201
             \fi%
1202
          \fi%
1203
        }%
1204
1205
      \fi%
1206 }
```

\svg@get@path \if@svg@file@found \svg@file@path \svg@file@name \svg@file@base \svg@file@suffix 1183 1184 }

The command \svg@get@path tries to find a given SVG file. If the searched file wasn't found in the current path, all paths given with \svgpath are evaluated. If there was no appropriate file again, all paths given by \graphicspath are examined. In the last step, a given path within the second mandatory argument is browsed. The results for file path and name are stored in \svg@file@path and \svg@file@name as well as the compound of both is saved in \svg@file@base.

```
1207 \newif\if@svg@file@found
1208 \newcommand*\svg@file@path{}
1209 \newcommand*\svg@file@name{}
1210 \newcommand*\svg@file@base{}
 1211 \newcommand*\svg@file@suffix{}
1212 \verb| newcommand*| svg@get@path[3][|svg@file@ext]{%} % The command is a simple of the command is a simple of the command o
                                                                \begingroup%
                                                                                        \svg@filename@parse[{#1}]{#2}%
1214
                                                                                      \IfArgIsEmpty{#1}{%
1215
                                                                                                           \verb|\edge f| svg@tempa{\filename@area\filename@base.\filename@ext}|| % filename@ext}|| % filename@ext}
1216
 1217
 1218
                                                                                                             \edef\svg@tempa{\filename@area\filename@base.#1}%
 1219
                                                                                     }%
```

After calling \svg@set@input@path, all search paths are stored in \input@path, a single path given in the third argument will also be considered.

```
1220 \svg@set@input@path[{#3}]%
```

The specified file is searched with \IfFileExists. If the file search was successful, the macro \svg@filename@parse is called with the result.

```
1221
        \@svg@tempswafalse%
        \expandafter\IfFileExists\expandafter{\svg@tempa}{%
1222
          \expandafter\svg@quotes@check\expandafter{\svg@tempa}%
1223
          \if@svg@quotes@found\else%
1224
            \svg@quotes@remove{\@filef@und}%
1225
1226
          \fi%
1227
          \@svg@tempswatrue%
1228
          \edef\@filef@und{\expandafter\trim@spaces\expandafter{\@filef@und}}%
          \svg@filename@parse[{#1}]{\@filef@und}%
1229
1230
        \edef\svg@tempa{%
1231
1232
          \endgroup%
1233
          \if@svg@tempswa%
            \noexpand\@svg@file@foundtrue%
1234
            \def\noexpand\svg@file@path{\filename@area}%
1235
            \def\noexpand\svg@file@name{\filename@base}%
1236
            \def\noexpand\svg@file@base{\filename@area\filename@base}%
1237
          \else%
1238
1239
            \noexpand\@svg@file@foundfalse%
1240
            \def\noexpand\svg@file@path{}%
1241
            \def\noexpand\svg@file@name{#2}%
1242
            \def\noexpand\svg@file@base{#2}%
1243
          \fi%
        }%
1244
      \svg@tempa%
1245
1246 }
```

## **B.5.** Patches

\svg@patches \svg@picture@saved \svg@includegraphics@saved For including the export results from *Inkscape* with LATEX support enabled, there are some patches necessary for environment picture and \includegraphics. These patches are done with \svg@patches.

```
1247 \newcommand*\svg@picture@saved{}

1248 \let\svg@picture@saved\picture

1249 \newcommand*\svg@includegraphics@saved{}

1250 \let\svg@includegraphics@saved\includegraphics

1251 \newcommand*\svg@patches[1]{%

1252 \let\picture\svg@picture@patched%

1253 \let\includegraphics\svg@includegraphics@patched%

1254 \edef\svg@includegraphics@file{#1}%

1255 }
```

\svg@pictur@patched

\svg@picture@patched In order to provide the possibility specify the desired width of a graphic, the appropriate \unitlength is calculated at the beginning of the picture environment.

```
1256 \newcommand*\svg@picture@patched{}
1257 \newcommand*\svg@pictur@patched{}
1258 \verb|\long\def\svg@picture@patched#1{\svg@pictur@patched@#1}|
1259 \def\svg@pictur@patched@(#1,#2){%
1260
      \svg@wrn@scale%
```

If a desired height is present, the resulting \unitlength is calculated with the ratio of the coordinates of the picture environment given as arguments for x- and y-direction by using \Gscale@div. With this factor, \unitlength—which is connected to the x-coordinate—can be scaled in a suitable manner.

```
\ifdim\svg@param@height>\z@\relax%
1261
        Gscale@div\svg@tempa{#1\p@}{#2\p@}%
1262
1263
        \setlength\unitlength{\svg@param@height}%
        \setlength\unitlength{\svg@tempa\unitlength}%
1264
1265
        \ifdim\svg@param@width>\z@\relax%
1266
          \ifdim\unitlength>\svg@param@width\relax%
1267
            \setlength\unitlength{\svg@param@width}%
1268
          \fi%
        \fi%
1269
1270
      \else%
```

If no height is given, \unitlength can be set easily.

```
\ifdim\svg@param@width>\z@\relax%
1271
1272
          \setlength\unitlength{\svg@param@width}%
1273
1274
          \setlength\unitlength{\svg@param@scale\unitlength}%
        \fi%
1275
1276
      \fi%
```

After setting \unitlength, the picture environment can be called with its original definition.

```
\svg@picture@saved(#1,#2)%
1277
1278 }
```

\svg@includegraphics@patched \svg@includegraphics@file

The patch to \includegraphics is meant to dissolve the Inkscape bug concerning the inclusion of more PDF pages than actually are existing.

The given optional parameters to \includegraphics are processed and the counter svg@param@currpage is set to the value of a given page. The value of parameter width is ignored.

```
1279 \DefineFamily{SVGpatch}
1280 \DefineFamilyMember{SVGpatch}
1281 \newcounter{svg@param@currpage}
1282 \setcounter{svg@param@currpage}{\m@ne}
1283 \FamilyCounterKey{SVGpatch}{page}{svg@param@currpage}
1284 \DefineFamilyKey{SVGpatch}{width}{\FamilyKeyStateProcessed}
1285 \newcommand*\svg@includegraphics@file{}
1286 \newcommand*\svg@includegraphics@patched[2][]{%
      \FamilyOptions{SVGpatch}{#1}%
```

If option lastpage was set to false, each page is included—even if it doesn't exist, which may cause errors.

```
\ifnum\value{svg@param@lastpage}<\z@\relax%
1288
1289
        \FamilySetCounter{SVGpatch}{page}{svg@param@currpage}{%
1290
          \the\value{svg@param@lastpage}%
        }%
1291
      \fi%
1292
```

Only if counter svg@param@lastpage is smaller than svg@param@currpage, pages are included, where svg@param@lastpage was either given as a number with parameter lastpage or was automatically calculated with \svg@get@lastpage.

```
1293 \ifnum\value{svg@param@currpage}>\value{svg@param@lastpage}\relax\else%
```

A page is included with the original definition of \includegraphics. All optional parameters are passed.

```
1294 \svg@includegraphics@saved[{#1}]{\svg@includegraphics@file}%
1295 \fi%
1296 }
```

# C. Extracting independent graphic files with svg-extract

# C.1. Options

For package **svg-extract** the user-interface of package **svg** is extended. The following options can either be set with \svgsetup or be used as optional parameters for \includesvg and \includeinkscape.

\svg@dummy@key

If package **svg-extract** wasn't loaded, the following options are defined for package **svg** in order to raise a warning message. Primarily this is done for compatibility reasons.

```
1297 (*main)
1298 \DefineFamilyMember[.dummy]{SVG}
1299 \newcommand*\svg@dummy@key[2][]{%
      \@ifpackageloaded{svg-extract}{}{%
1301
        \IfArgIsEmpty{#1}{%
          1302
1303
            \PackageWarning{svg}{%
              The option key '#2' can only\MessageBreak%
1304
1305
              be used with package 'svg-extract', but\MessageBreak%
1306
              you didn't load it%
            }%
1307
            \FamilyKeyStateProcessed%
1308
1309
          }%
1310
          \label{lem:lykey[.dummy]} $$ \operatorname{SVG}_{\#2}[{\#1}]_{\%} $$
1311
1312
            \PackageWarning{svg}{%
              The option key '#2' can only\MessageBreak%
1313
              be used with package 'svg-extract', but\MessageBreak%
1314
              you didn't load it%
1315
            }%
1316
1317
            \FamilyKeyStateProcessed%
1318
          }%
        }%
```

Before package svg-extract the given key #2 of family member .dummy is relaxed.

## C.1.1. Controlling the extract process

should be done. \if@svgx@run 1324 (\*main) 1325 \svg@dummy@key[true]{extract} 1326 (/main) 1327 (\*extract) 1328 \newif\if@svgx@run 1329 \DefineFamilyKey{SVG}{extract}[true]{% \lowercase{\def\svg@tempa{#1}}% \FamilySetNumerical{SVG}{extract}{svg@tempa}{% 1331 {false}{0},{off}{0},{no}{0},% 1332 1333 {true}{1},{on}{1},{yes}{1},{onlynewer}{1},{newer}{1},% {overwrite}{1},{force}{1},{forced}{1},% 1334 {pdf}{2},{eps}{3},{ps}{4}% 1335 1336 }{\svg@tempa}% 1337 \ifx\FamilyKeyState\FamilyKeyStateProcessed% 1338 \ifcase\svg@tempa\relax% false 1339 \@svgx@runfalse% 1340 \or% true 1341\@svgx@runtrue% 1342 \or% pdf \FamilyOptions{SVG}{extractformat=pdf}% 1343 \or% eps 1344 \FamilyOptions{SVG}{extractformat=eps}% 1345 \or% ps 1346 1347 \FamilyOptions{SVG}{extractformat=ps}% 1348 \fi% \fi% 1349 1350 } 1351 (/extract) Package options which can be used to switch functionality on or off during the loading of on (opt.) package svg-extract. off (opt.) 1352 (\*extract) 1353 \DeclareOption{on}{\FamilyOptions{SVG}{extract=true}} 1354 \DeclareOption{off}{\FamilyOptions{SVG}{extract=false}} 1355 (/extract) extractformat (opt.) Option extractformat controls the output format (pdf/eps/ps). It is set to pdf or, if dvi \svgx@format output could be detected, to eps during initialization. pdf (opt.) 1356 (\*main) eps (opt.) 1357 \svg@dummy@key{extractformat} 1358 \svg@dummy@key[true]{pdf} 1359 \svg@dummy@key[true]{eps} 1360 (/main) 1361 (\*extract) 1362 \newcommand\*\svgx@format{pdf} 1363 \ifxetex\else\ifpdf\else 1364 \renewcommand\*\svgx@format{eps} 1365 \fi\fi 1366 \DefineFamilyKey{SVG}{extractformat}{%  $\verb|\lowercase{\edef\svgx@format{#1}}||$ 1367 \FamilyKeyStateProcessed% 1368 1369 } 1370 \DefineFamilyKey{SVG}{pdf}[true]{% \FamilySetBool{SVG}{pdf}{@svg@tempswa}{#1}% 1372 \ifx\FamilyKeyState\FamilyKeyStateProcessed% 1373 \if@svg@tempswa% \svgx@ifinlist{pdf}{\svgx@format}{}{% 1374

With option extract it can be controlled, if the extraction of independent graphic files

\edef\svgx@format{\svgx@format,pdf}%

1375

```
1376
                                     }%
                           1377
                                     \svg@deprecated@key{pdf}{extractformat={\svgx@format}}%
                           1378
                                   \else%
                                     \FamilyKeyStateUnknownValue%
                           1379
                                   \fi%
                           1380
                           1381
                                 \fi%
                           1382 }
                           1383 \DefineFamilyKey{SVG}{eps}[true]{%
                                 \label{lem:condition} $$ \widetilde{SVG}_{eps}_{osvg0tempswa}_{\#1}\% $$
                           1384
                                 \ifx\FamilyKeyState\FamilyKeyStateProcessed%
                           1385
                                   \if@svg@tempswa%
                           1386
                                     \svgx@ifinlist{eps}{\svgx@format}{}{%
                           1387
                           1388
                                        \edef\svgx@format{\svgx@format,eps}%
                           1389
                           1390
                                      \svg@deprecated@key{eps}{extractformat={\svgx@format}}%
                           1391
                           1392
                                     \FamilyKeyStateUnknownValue%
                           1393
                                   \fi%
                           1394
                                 \fi%
                           1395 }
                           1396 (/extract)
                           For the extraction process, a preamble is necessary for a separate auxiliary LATEX file.
   extractpreamble (opt.)
                           By default, the preamble of the main document is used, which end is detected at
          preamble (opt.)
                          \begin{document}.
         \svgx@preamble
extractpreambleend (opt.)
                           1397 (*main)
                end (opt.)
                           1398 \svg@dummy@key{extractpreamble}
      \svgx@endpreamble
                           1399 \svg@dummy@key{preamble}
                           1400 \svg@dummy@key{extractpreambleend}
                           1401 \svg@dummy@key{end}
                           1402 (/main)
                           1403 (*extract)
                           1404 \newcommand*\svgx@preamble{\jobname.\svgx@latex@ext}%
                           1405 \DefineFamilyKey{SVG}{extractpreamble}{%
                           1406
                                 \renewcommand*\svgx@preamble{#1}%
                                 \FamilyKeyStateProcessed%
                           1407
                           1408 }
                           1409 \DefineFamilyKey{SVG}{preamble}{%
                                 \svg@deprecated@key[svg-extract]{preamble=#1}{extractpreamble=#1}%
                           1410
                           1411 }
                           1412 \newcommand*\svgx@endpreamble{}
                           1413 \expandafter\def\expandafter\svgx@endpreamble\expandafter{%
                                 \csname begin\endcsname{document}%
                           1414
                           1415 }
                           1416 \DefineFamilyKey{SVG}{extractpreambleend}{%
                                 \renewcommand*\svgx@endpreamble{#1}%
                           1418
                                 \FamilyKeyStateProcessed%
                           1419 }
                           1420 \DefineFamilyKey{SVG}{end}{%
                                 \svg@deprecated@key[svg-extract]{end=#1}{extractpreambleend=#1}%
                           1421
                           1422 }
                           1423 (/extract)
                          With this option, the number of LATEX runs for the separate auxiliary file can be set.
       extractruns (opt.)
      svgx@runs (counter)
                           1424 (*main)
                           1425 \svg@dummy@key{extractruns}
                           1426 \langle /main \rangle
                           1427 (*extract)
                           1428 \newcounter{svgx@runs}
                           1429 \verb|\DefineFamilyKey{SVG}{extractruns}{|} \%
                                 \FamilySetCounter{SVG}{extractruns}{svgx@runs}{#1}%
                           1430
                                 \ifx\FamilyKeyState\FamilyKeyStateProcessed%
                           1431
                                   \ifnum\value{svgx@runs}<\@ne\relax%
                           1432
```

```
1433
                                \PackageWarning{svg-extract}{%
                     1434
                                  The count for runs has to be at least one%
                     1435
                                \FamilySetCounter{SVG}{extractruns}{svgx@runs}{\@ne}%
                     1436
                     1437
                              \fi%
                     1438
                            \fi%
                     1439 }
                     1440 (/extract)
                     The command and facultative options for the LATEX call of the separate auxiliary file. The
     latexexe (opt.)
                     default is set according to the currently used engine.
     pdflatex (opt.)
   \svgx@latex@exe
                     1441 (*main)
     latexext (opt.)
                     1442 \svg@dummy@key{latexexe}
   \svgx@latex@ext
                     1443 \svg@dummy@key{pdflatex}
     {\tt latexopt}\ ({\rm opt.})
                     1444 \svg@dummy@key{latexext}
   \svgx@latex@opt
                     1445 \svg@dummy@key{latexopt}
                     1446 (/main)
                     1447 (*extract)
                     1448 \ifxetex
                           \newcommand*\svgx@latex@exe{xelatex}
                     1450 \else\ifluatex
                     1451
                           \ifpdf
                              \newcommand*\svgx@latex@exe{lualatex}
                     1452
                           \else
                     1453
                              \newcommand*\svgx@latex@exe{lualatex --output-format=dvi}
                     1454
                           \fi
                     1455
                     1456 \leq ifpdf
                           \newcommand*\svgx@latex@exe{pdflatex}
                     1457
                     1458 \else
                     1459
                           \newcommand*\svgx@latex@exe{latex}
                     1460 \fi\fi\fi
                     1461 \DefineFamilyKey{SVG}{latexexe}{%
                     1462
                            \renewcommand*\svgx@latex@exe{#1}%
                            \FamilyKeyStateProcessed%
                     1463
                     1464 }
                     1465 \DefineFamilyKey{SVG}{pdflatex}{%
                            \svg@deprecated@key[svg-extract]{pdflatex=#1}{latexexe=#1}%
                     1466
                     1467 }
                     1468 \newcommand*\svgx@latex@ext{tex}
                     1469 \DefineFamilyKey{SVG}{latexext}{%
                            \renewcommand*\svgx@latex@ext{#1}%
                     1471
                            \FamilyKeyStateProcessed%
                     1472 }
                     1473 \newcommand*\svgx@latex@opt{}
                     1474 \DefineFamilyKey{SVG}{latexopt}{%
                            \renewcommand*\svgx@latex@opt{#1}%
                            \FamilyKeyStateProcessed%
                     1476
                     1477 }
                     1478 (/extract)
                     Options and macros for calling convert commands, which are supplied by most LATEX distri-
     dvipsopt (opt.)
   \svgx@dvips@exe
                     butions. These are used to generate all files, which are supported by option extractformat,
                     as they don't need an additional application.
   \svgx@dvips@opt
   pstoepsopt (opt.)
                     1479 (*main)
\svgx@pstoeps@exe
                     1480 \svg@dummy@key{dvipsopt}
\svgx@pstoeps@opt
                     1481 \svg@dummy@key{pstoepsopt}
   pstopdfopt (opt.)
                     1482 \svg@dummy@key{pstopdfopt}
 \svgx@pstopdf@exe
                     1483 \svg@dummy@key{pdftoepsopt}
\svgx@pstopdf@opt
                     1484 \svg@dummy@key{pdftopsopt}
  pdftoepsopt (opt.)
                     1485 \svg@dummy@key{pdftops}
\svgx@pdftoeps@exe
                     1486 \langle /main \rangle
\svgx@pdftoeps@opt
                     1487 (*extract)
                     1488 \newcommand*\svgx@dvips@exe{dvips}
   {\tt pdftopsopt}\ ({\rm opt.})
\svgx@pdftops@exe
\svgx@pdftops@opt
      pdftops (opt.)
```

```
1489 \newcommand*\svgx@dvips@opt{}
1490 \DefineFamilyKey{SVG}{dvipsopt}{%
      \renewcommand*\svgx@dvips@opt{#1}%
      \FamilyKeyStateProcessed%
1493 }
1494 \verb|\newcommand*\svgx@pstoeps@exe{ps2eps}|
1495 \newcommand*\svgx@pstoeps@opt{-B -C}
1496 \DefineFamilyKey{SVG}{pstoepsopt}{%
1497
      \renewcommand*\svgx@pstoeps@opt{#1}%
1498
      \FamilyKeyStateProcessed%
1499 }
1500 \newcommand*\svgx@pstopdf@exe{ps2pdf}
1501 \newcommand*\svgx@pstopdf@opt{}
1502 \DefineFamilyKey{SVG}{pstopdfopt}{%
      \renewcommand*\svgx@pstopdf@opt{#1}%
1504
      \FamilyKeyStateProcessed%
1505 }
1506 \newcommand*\svgx@pdftoeps@exe{pdftops -eps}
1507 \newcommand*\svgx@pdftoeps@opt{}
1508 \verb|\DefineFamilyKey{SVG}{pdftoepsopt}{%}|
      \renewcommand*\svgx@pdftoeps@opt{#1}%
      \FamilyKeyStateProcessed%
1510
1511 }
1512 \newcommand*\svgx@pdftops@exe{pdftops}
1513 \newcommand*\svgx@pdftops@opt{}
1514 \DefineFamilyKey{SVG}{pdftopsopt}{%
      \renewcommand*\svgx@pdftops@opt{#1}%
1516
      \FamilyKeyStateProcessed%
1517 }
1518 \verb|\DefineFamilyKey{SVG}{pdftops}{%}
      \PackageWarning{svg-extract}{%
1519
        The option key 'pdftops' is deprecated. \MessageBreak%
1520
        You should use either 'pdftoepsopt' or\MessageBreak%
1521
        'pdftopsopt' instead. See the manual for\MessageBreak%
1522
        more. Nothing was done%
1523
1524
      \FamilyKeyStateProcessed%
1525
1526 }
1527 (/extract)
```

## C.1.2. Invoking external application for graphic conversion

Besides the use of a conversion tool supplied by the LATEX distribution, the applications ImageMagick and Ghostscript can be used for converting graphics.

convert (opt.)
\if@svgx@cnv@run
\svgx@cnv@cmd

The option convert can be used to define, which of both applications should be use. ImageMagick is set by default.

```
1528 (*main)
1529 \svg@dummy@key[true]{convert}
1530 (/main)
1531 \langle *extract \rangle
1532 \newif\if@svgx@cnv@run
1533 \newcommand*\svgx@cnv@cmd{}
1534 \DefineFamilyKey{SVG}{convert}[true]{%
      \FamilySetNumerical{SVG}{convert}{svg@tempa}{%
1535
1536
         \{false\}\{0\}, \{off\}\{0\}, \{no\}\{0\}, \%
         {true}{1}, {on}{1}, {yes}{1}, {onlynewer}{1}, {newer}{1}, %
1537
         {overwrite}{1},{force}{1},{forced}{1},%
1538
         {magick}{2},{imagemagick}{2},{convert}{2},%
1539
         {gs}{3},{ghostscript}{3},%
1540
         {gs64}{4},{ghostscript64}{4},%
1541
         {gs32}{5},{ghostscript32}{5}%
1542
      }{#1}%
1543
```

```
1544
      \ifx\FamilyKeyState\FamilyKeyStateProcessed%
1545
        \ifcase\svg@tempa\relax% false
          \@svgx@cnv@runfalse%
1546
        \or% true
1547
          \@svgx@cnv@runtrue%
1548
1549
        \or% magick
1550
          \@svgx@cnv@runtrue%
          \renewcommand*\svgx@cnv@cmd{\svgx@magick@cmd}%
1551
1552
        \or% gs
          \@svgx@cnv@runtrue%
1553
          \renewcommand*\svgx@cnv@cmd{\svgx@gs@cmd}%
1554
        \or% gs64
1555
          \@svgx@cnv@runtrue%
1556
1557
          \renewcommand*\svgx@cnv@cmd{\svgx@gs@cmd}%
1558
          \svg@ifwindowsdetected{%
1559
            \renewcommand*\svgx@gs@exe{gswin64c}%
1560
          }{}%
        \or% gs32
1561
          \@svgx@cnv@runtrue%
1562
          \renewcommand*\svgx@cnv@cmd{\svgx@gs@cmd}%
1563
          \svg@ifwindowsdetected{%
1564
            \renewcommand*\svgx@gs@exe{gswin32c}%
1565
1566
          }{}%
1567
        \fi%
```

In version v1.0 the option convert was used to set both the executable and options for the conversion application, meant for the usage of *ImageMagick*. This is taken into account here.

```
1568 \else% legacy option
```

Same doing like with legacy part of option inkscape.

```
\def\svg@tempa##1-##2\@nil{%
1569
          \IfArgIsEmpty{##2}{\def\svg@tempb{}}{%
1570
            1571
            \svg@tempa#1\@nil%
1572
          }%
1573
          \def\svg@tempa{##1}%
1574
       }%
1575
1576
        \svg@tempa#1-\@nil%
1577
        \PackageWarning{svg-extract}{%
1578
          Setting the executable%
1579
          \ifx\svg@tempb\@empty\else%
1580
            \space and associated options%
          \fi%
1581
          \MessageBreak%
1582
          for ImageMagick should be done with options\MessageBreak%
1583
          'magickexe=\svg@tempa'%
1584
          \ifx\svg@tempb\@empty\else%
1585
            \MessageBreak and 'magicksetting' and/or 'magickoperator'%
1586
1587
          \fi.\MessageBreak%
          Nevertheless, this was done by now%
1588
          \ifx\svg@tempb\@empty\else%
1589
1590
             whereby \MessageBreak 'magicksetting=\svg@tempb' was used%
1591
          \fi%
1592
        \FamilyOptions{SVG}{convert=magick}%
1593
        \edef\svg@tempa{%
1594
          \noexpand\FamilyOptions{SVG}{magickexe=\svg@tempa}%
1595
          \ifx\svg@tempb\@empty\else%
1596
            \noexpand\FamilyOptions{SVG}{magicksetting=\svg@tempb}%
1597
          \fi%
1598
        }%
1599
1600
        \svg@tempa%
1601
      \fi%
```

```
1602 }
1603 ⟨/extract⟩
```

 $\label{eq:convertformat} \textbf{convertformat} \ (\text{opt.}) \\ \textbf{\scalebel{eq:convertformat}}$ 

 $png \ ({\rm opt.})$ 

Option convertformat controls the output format for converted files. It is set to png by default.

```
1604 (*main)
1605 \svg@dummy@key{convertformat}
1606 \svg@dummy@key[true]{png}
1607 (/main)
1608 (*extract)
1609 \newcommand*\svgx@cnv@format{png}
1610 \DefineFamilyKey{SVG}{convertformat}{%
      \lowercase{\edef\svgx@cnv@format{#1}}%
1611
      \ifx\svgx@cnv@format\@empty\else%
1612
1613
        \@svgx@cnv@runtrue%
1614
      \fi%
1615
      \FamilyKeyStateProcessed%
1616 }
1617 \DefineFamilyKey{SVG}{png}[true]{%
      \FamilySetBool{SVG}{png}{@svg@tempswa}{#1}%
1619
      \ifx\FamilyKeyState\FamilyKeyStateProcessed%
1620
        \if@svg@tempswa%
          \svgx@ifinlist{png}{\svgx@cnv@format}{}{%
1621
            \edef\svgx@cnv@format{\svgx@cnv@format,png}%
1622
          }%
1623
          \svg@deprecated@key{png}{convertformat={\svgx@cnv@format}}%
1624
1625
        \else%
1626
          \FamilyKeyStateUnknownValue%
1627
        \fi%
1628
      \fi%
1629 }
1630 (/extract)
```

convertdpi (opt.)
convertdensity (opt.)
\svgx@cnv@dpi
\svgx@cnv@get@dpi

The option convertdpi is meant to define the used density during the conversion process. It can be set either for all designated output formats or targeted for a specific format. It's also possible to use something like 500x300. Given values are resolved by \svgx@cnv@get@dpi. It's used like convertdpi=300 or convertdpi={png=600} If the option is used for a specific or for all output formats is recognized by \svgx@ifkeyandval.

```
1631 (*main)
1632 \svg@dummy@key{convertdpi}
1633 \svg@dummy@key{convertdensity}
1634 (/main)
1635 (*extract)
1636 \newcommand*\svgx@cnv@dpi{}
1637 \let\svgx@cnv@dpi\relax
1638 \DefineFamilyKey{SVG}{convertdpi}{%
1639
      \FamilyKeyStateUnknownValue%
      \svgx@ifkeyandval{#1}{%
1640
        \svgx@cnv@get@dpi{##2}%
1641
        \ifx\svg@tempa\relax\else%
1642
1643
          \expandafter\edef\csname svgx@cnv@dpi@##1\endcsname{\svg@tempa}%
1644
          \FamilyKeyStateProcessed%
1645
      }{%
1646
        \svgx@cnv@get@dpi{##1}%
1647
1648
        \ifx\svg@tempa\relax\else%
1649
          \edef\svgx@cnv@dpi{\svg@tempa}%
          \FamilyKeyStateProcessed%
1650
1651
        \fi%
      }%
1652
1653 }
1654 \DefineFamilyKey{SVG}{convertdensity}{\FamilyOptions{SVG}{convertdpi=#1}}
```

This macro is used to resolve a given value to set the density for the conversion. The delimited macros \svg@tempa and \svg@tempb are defined to first crop any given suffix dpi and second to split two numbers at x, if present. Pay attention how both macros are invoked. In the end, a passed value in any of the forms 300, 300dpi, 300x400 or 300x400dpi and even 300dpix400dpi is possible. The result is stored in \svg@tempa.

```
1655 \newcommand*\svgx@cnv@get@dpi[1]{%
1656 \begingroup%
1657 \def\svg@tempa##1dpi##2x##3dpi##4\@nil{%
1658 \edef\svg@tempa{##1}%
```

Switch \if@svg@tempswa as \iftrue means, a valid value was found.

```
1659 \@svg@tempswafalse%
```

If only the first argument is a number and third is empty, a single number was given and there's nothing more to do. If the argument is something like 300dpix400dpi, the third argument is the second number.

```
1660
          \Ifnumber{##1}{%
            \IfArgIsEmpty{##3}{\@svg@tempswatrue}{%
1661
               \Ifnumber{##3}{\edef\svg@tempa{##1x##3}}{}%
1662
            }%
1663
          }{}%
1664
1665
          \if@svg@tempswa\else%
             \expandafter\svg@tempb\svg@tempa xx\@nil%
1666
1667
          \fi%
1668
```

Macro  $\svg@tempb$  splits at x and checks, if something valid like 300x400 was given. If true, the value is stored in  $\svg@tempa$ .

```
1669
        \def\svg@tempb##1x##2x##3\@ni1{%}
1670
          \left\{ x\right\} 
1671
             \@svg@tempswatrue%
             \IfArgIsEmpty{##1}{\@svg@tempswafalse}{%
1672
               \Ifnumber{##1}{}{\@svg@tempswafalse}%
1673
             }%
1674
             \IfArgIsEmpty{##2}{\@svg@tempswafalse}{%
1675
               \Ifnumber{##2}{}{\@svg@tempswafalse}%
1676
1677
             }%
1678
             \if@svg@tempswa%
1679
               \edef\svg@tempa{##1x##2}%
1680
             \fi%
1681
          }{}%
        }%
1682
        \IfArgIsEmpty{#1}{%
1683
          \let\svg@tempa\@empty%
1684
        }{%
1685
           \lowercase{\svg@tempa#1dpi#1xdpi\@nil}%
1686
1687
           \if@svg@tempswa\else%
             \let\svg@tempa\relax%
1688
          \fi%
1689
        }%
1690
1691
        \edef\svg@tempb{%
1692
          \endgroup%
1693
           \ifx\svg@tempa\relax%
1694
             \let\noexpand\svg@tempa\noexpand\relax%
1695
             \def\noexpand\svg@tempa{\svg@tempa}%
1696
1697
          \fi%
        }%
1698
      \svg@tempb%
1699
1700 }
1701 (/extract)
```

\svgx@setformatkey \svgx@useformatkey

With \svgx@setformatkey the—maybe output format depend—keys for the conversion tools are set. First argument contains the value given to a key, second the command sequence name of the macro, to whom the value shall be allocated.

1702 \newcommand\*\svgx@setformatkey[2]{%

A key of the form  $\langle key \rangle = \{\langle format \rangle = \langle value \rangle\}$  is given. The desired output format can be accessed with ##1, the value with ##2 within the arguments of \svgx@ifkeyandval.

```
1703 \svgx@ifkeyandval{#1}{%

1704 \svg@ifvalueisrelax{##2}{%

1705 \expandafter\let\csname #2@##1\endcsname\relax%

1706 }{%

1707 \@namedef{#2@##1}{##2}%

1708 }%
```

A key of the form  $\langle key \rangle = \{\langle format \rangle = \langle value \rangle \}$  is given. The value can be used with ##1.

The command \svgx@useformatkey checks, if a format specific key was defined with \svgx@setformatkey, whereas the format is given in the second argument. If this is not the case, the setting for all output formats is used. After that, a specific key appended with a + can be used to do some additional stuff.

```
1717 \newcommand*\svgx@useformatkey[3]{%
      \scr@ifundefinedorrelax{#1@#2}{%
1718
        \scr@ifundefinedorrelax{#1}{}{%
1719
          \expandafter\ifx\csname #1\endcsname\@empty\else%
1720
            #3\@nameuse{#1}\space%
1721
1722
          \fi%
1723
1724
        \scr@ifundefinedorrelax{#1@#2+}{}{%
1725
          \expandafter\ifx\csname #10#2+\endcsname\@empty\else%
1726
            #3\@nameuse{#1@#2+}\space%
1727
          \fi%
        }%
1728
      }{%
1729
```

If a format specific key was definded, it is used.

```
1730 \expandafter\ifx\csname #1@#2\endcsname\@empty\else%

1731 #3\@nameuse{#1@#2}\space%

1732 \fi%

1733 }%

1734 }
```

magickexe (opt.)
\svgx@magick@exe
magicksetting (opt.)
\svgx@magick@set
magickoperator (opt.)
\svgx@magick@opr
\svgx@magick@cmd

Setting the command including maybe the path to *ImageMagick*. The keys magicksetting and magickoperator should be used to add optional arguments before (*Settings*) or after (*Operators*) the input file. They can either be set for all or a specific output format as like option convertdpi. For this \svgx@setformatkey is used.

```
1735 \ *main \\
1736 \ svg@dummy@key{magickexe}
1737 \ svg@dummy@key{magicksetting}
1738 \ svg@dummy@key{magickoperator}
1739 \ /main \\
1740 \ *extract \\
1741 \ svg@ifwindowsdetected{%
```

```
1742
                       \newcommand*\svgx@magick@exe{magick}%
                 1743 }{%
                 1744
                       \newcommand*\svgx@magick@exe{convert}%
                 1745 }
                 1746 \DefineFamilyKey{SVG}{magickexe}{%
                 1747
                       \renewcommand*\svgx@magick@exe{#1}%
                       \FamilyKeyStateProcessed%
                 1748
                 1749 }
                 1750 \newcommand*\svgx@magick@set{}
                 1751 \DefineFamilyKey{SVG}{magicksetting}{%
                       \svgx@setformatkey{#1}{svgx@magick@set}%
                       \FamilyKeyStateProcessed%
                 1753
                 1754 }
                 1755 \newcommand*\svgx@magick@opr{}
                 1756 \DefineFamilyKey{SVG}{magickoperator}{%
                       \svgx@setformatkey{#1}{svgx@magick@opr}%
                 1758
                       \FamilyKeyStateProcessed%
                 1759 }
                 1760 \newcommand*\svgx@magick@cmd[3]{%
                       \svgx@magick@exe\space%
                 1761
                       \svgx@useformatkey{svgx@cnv@dpi}{#3}{-density }%
                 1762
                       1763
                       "#1.#2"\space%
                 1764
                 1765
                       \svgx@useformatkey{svgx@magick@opr}{#3}{}%
                       "#1.#3"%
                 1766
                 1767 }
                 1768 (/extract)
                 Options to set the command including maybe the path to Ghostscript. As Ghostscript
     gsexe (opt.)
                 needs a specific device defined for different output formats, the option gsdevice can be used.
  \svgx@gs@exe
                 It can either be set for all or a specific output format just like gsopt in the same manner
     gsopt (opt.)
                 like option convertdpi.
  \svgx@gs@opt
  gsdevice (opt.)
                 1769 (*main)
\svgx@gs@device
                 1770 \svg@dummy@key{gsexe}
  \svgx@gs@cmd
                 1771 \svg@dummy@key{gsopt}
                 1772 \svg@dummy@key{gsdevice}
                 1773 (/main)
                 1774 (*extract)
                 1775 \svg@ifwindowsdetected{%
                       \newcommand*\svgx@gs@exe{gswin64c}%
                 1777 }{%
                       \newcommand*\svgx@gs@exe{gs}%
                 1778
                 1779 }
                 1780 \DefineFamilyKey{SVG}{gsexe}{%
                       \renewcommand*\svgx@gs@exe{#1}%
                 1781
                       \FamilyKeyStateProcessed%
                 1782
                 1784 \newcommand*\svgx@gs@opt{}
                 1785 \DefineFamilyKey{SVG}{gsopt}{%
                       \svgx@setformatkey{#1}{svgx@gs@opt}%
                 1787
                       \FamilyKeyStateProcessed%
                 1788 }
                 1789 \newcommand*\svgx@gs@device{}
                 1790 \DefineFamilyKey{SVG}{gsdevice}{%
                       \svgx@setformatkey{#1}{svgx@gs@device}%
                 1791
                       \FamilyKeyStateProcessed%
                 1792
                 1793 }
                 1794 \newcommand*\svgx@gs@cmd[3]{%
                       \svgx@gs@exe\space-dSAFER -dBATCH -dNOPAUSE\space%
                 1795
                 1796
                       \svgx@useformatkey{svgx@gs@device}{#3}{-sDEVICE=}%
                 1797
                       \svgx@useformatkey{svgx@cnv@dpi}{#3}{-r}%
                       \svgx@useformatkey{svgx@gs@opt}{#3}{}%
                 1798
                       -sOutputFile="#1.#3"\space"#1.#2"%
                 1799
                 1800 }
```

## C.1.3. Setting output folder

```
extractpath (opt.)
                         The option extractpath controls, in which folder the results both of the extraction as well
                         as the conversion of ImageMagick or Ghostscript will be located.
             path (opt.)
        \svgx@out@path
                         1802 (*main)
                         1803 \svg@dummy@key{extractpath}
                         1804 \svg@dummy@key{path}
                         1805 (/main)
                         1806 (*extract)
                         1807 \newcommand*\svgx@out@path{}
                         1808 \DefineFamilyKey{SVG}{extractpath}{%
                         1809
                                \svg@sanitize@dq\svg@tempb{#1}%
                         1810
                                \FamilySetNumerical{SVG}{extractpath}{svg@tempa}{%
                         1811
                                  {svgpath}{0},{svgdir}{0},%
                         1812
                                  {svgsubpath}{1},{svgsubdir}{1},%
                                  \label{local-equation} $$\{basepath\}_{2}, \{basedir}_{2}, \{jobpath\}_{2}, \{jobdir}_{2}, \%$
                         1813
                                  {basesubpath}{3},{basesubdir}{3},{jobsubpath}{3},{jobsubdir}{3}%
                         1814
                         1815
                                }{\svg@tempb}%
                                \ifx\FamilyKeyState\FamilyKeyStateProcessed%
                         1816
                                  \ifcase\svg@tempa\relax% svgpath
                         1817
                         1818
                                    \renewcommand*\svgx@out@path{\svg@file@path}%
                         1819
                                  \or% svgsubpath
                                    \renewcommand*\svgx@out@path{\svg@file@path svg-extract/}%
                         1820
                                  \or% basepath
                         1821
                         1822
                                    \renewcommand*\svgx@out@path{./}%
                         1823
                                  \or% basesubpath
                         1824
                                    \renewcommand*\svgx@out@path{./svg-extract/}%
                         1825
                                  \fi%
                                \else%
                         1826
                                  \edef\svgx@out@path{\svg@tempb}%
                         1827
                                  \svg@normalize@path{\svgx@out@path}%
                         1828
                         1829
                                  \FamilyKeyStateProcessed%
                         1830
                                \fi%
                         1831 }
                         1832 \DefineFamilyKey{SVG}{path}{%
                         1833
                                \svg@deprecated@key[svg-extract]{path=#1}{extractpath=#1}%
                         1834 }
                         1835 \langle /extract \rangle
                         With option extractname the name of the extracted and maybe converted file itself can be
     extractname (opt.)
                         changed.
             name (opt.)
        \svgx@out@name
                         1836 (*main)
     \if@svgx@out@sec
                         1837 \svg@dummy@key{extractname}
svgx@out@count (counter)
                         1838 \svg@dummy@key{name}
         \svgx@out@sec
                         1839 (/main)
                         1840 (*extract)
                         1841 \newcommand*\svgx@out@name{}
                         1842 \newif\if@svgx@out@sec
                         1843 \newcounter{svgx@out@count}
                         1844 \newcommand*\svgx@out@sec{unknown}
                         1845 \DefineFamilyKey{SVG}{extractname}{%
                                \svg@quotes@remove[{#1}]{\svg@tempb}%
                         1846
                                \FamilySetNumerical{SVG}{extractname}{svg@tempa}{%
                         1847
                                  {filename}{0},{name}{0},%
                         1848
                                  {filenamenumbered}{1},{namenumbered}{1},%
                         1849
                                  {numberedfilename}{1}, {numberedname}{1}, %
                         1850
                         1851
                                  {numbered}{2}, {section}{2}, {numberedsection}{2}, {sectionnumbered}{2}%
                                }{\svg@tempb}%
                         1852
```

\@svgx@out@secfalse%

1853

1854

```
1855
        \ifcase\svg@tempa\relax% filename
1856
          \renewcommand*\svgx@out@name{\svg@out@name-extract}%
1857
        \or% filenamenumbered
           \renewcommand*\svgx@out@name{\the\value{svgx@out@count}-\svg@out@name}%
1858
1859
1860
          \renewcommand*\svgx@out@name{\the\value{svgx@out@count}-\svgx@out@sec}%
1861
          \@svgx@out@sectrue%
1862
        \fi%
1863
      \else%
        \if@svg@quotes@found%
1864
          \edef\svgx@out@name{"\svg@tempb"}%
1865
        \else%
1866
          \edef\svgx@out@name{\svg@tempb}%
1867
1868
1869
        \FamilyKeyStateProcessed%
1870
      fi%
1871 }
1872 \DefineFamilyKey{SVG}{name}{%}
      \svg@deprecated@key[svg-extract]{name=#1}{extractname=#1}%
1874 }
1875 \langle /extract \rangle
```

### C.1.4. Options for the extraction of graphics

extractwidth (opt.)
\svgx@param@width
extractheight (opt.)
\svgx@param@width
extractdistort (opt.)
extractkeepaspectratio (opt.)
\svgx@param@distort
extractscale (opt.)
\svgx@param@scale

For graphic extraction, the given settings regarding the size for inclusion can be overwritten with these options. Using \relax as value leads to resetting an option as unset, regardless of what was previously given. The value inherit means, that the actual option for including is used for extraction as well. This is the default setting.

```
1876 (*main)
1877 \svg@dummy@key{extractwidth}
1878 \svg@dummy@key{extractheight}
1879 \svg@dummy@key{extractdistort}
1880 \svg@dummy@key{extractkeepaspectratio}
1881 \svg@dummy@key{extractscale}
1882 (/main)
1883 (*extract)
1884 \newcommand*\svgx@param@width{\svg@param@width}
1885 \DefineFamilyKey{SVG}{extractwidth}{%
      \FamilyKeyStateUnknownValue%
1886
      \svg@ifvalueisrelax{#1}{%
1887
         \renewcommand*\svgx@param@width{\z@}%
1888
         \FamilyKeyStateProcessed%
1889
1890
         \Ifstr{#1}{inherit}{%
1891
           \renewcommand*\svgx@param@width{\svg@param@width}%
1892
1893
           \FamilyKeyStateProcessed%
         }{%
1894
           \FamilySetLengthMacro{SVG}{extractwidth}{\svgx@param@width}{#1}%
1895
           \ifx\FamilyKeyState\FamilyKeyStateProcessed%
1896
             \label{lem:condition} $$ \left( \frac{y}{2} \right) = \frac{1}{2} . $$ \left( \frac{y}{2} \right) = \frac{y}{2} . $$
1897
                \FamilyKeyStateUnknownValue%
1898
1899
             \fi%
1900
           \fi%
         }%
1901
1902
      }%
1903 }
1904 \newcommand*\svgx@param@height{\svg@param@height}
1905 \DefineFamilyKey{SVG}{extractheight}{%
      \FamilyKeyStateUnknownValue%
1906
      \svg@ifvalueisrelax{#1}{%
1907
1908
         \renewcommand*\svgx@param@height{\z@}%
1909
         \FamilyKeyStateProcessed%
1910
      }{%
```

```
1912
                                \renewcommand*\svgx@param@height{\svg@param@height}%
                     1913
                                \FamilyKeyStateProcessed%
                     1914
                                \FamilySetLengthMacro{SVG}{extractheight}{\svgx@param@height}{#1}%
                     1915
                     1916
                                \ifx\FamilyKeyState\FamilyKeyStateProcessed%
                     1917
                                  \ifdim\svgx@param@height<\z@\relax%
                                    \FamilyKeyStateUnknownValue%
                     1918
                                  \fi%
                     1919
                                \fi%
                     1920
                              ጉ%
                     1921
                     1922
                           }%
                     1923 }
                     1924 \newif\if@svgx@param@distort
                     1925 \DefineFamilyKey{SVG}{extractdistort}[true]{%
                            \FamilyKeyStateUnknownValue%
                     1927
                            \svg@ifvalueisrelax{#1}{%
                     1928
                              \@svgx@param@distortfalse%
                     1929
                              \FamilyKeyStateProcessed%
                           }{%
                     1930
                              \Ifstr{#1}{inherit}{%
                     1931
                                \renewcommand*\if@svgx@param@distort{\if@svg@param@distort}%
                     1932
                                \FamilyKeyStateProcessed%
                     1933
                     1934
                                \FamilySetBool{SVG}{extractdistort}{@svgx@param@distort}{#1}%
                     1935
                              }%
                     1936
                     1937
                           }%
                     1938 }
                     1939 \DefineFamilyKey{SVG}{extractkeepaspectratio}[true]{%
                     1940
                            \FamilySetBool{SVG}{extractkeepaspectratio}{@svg@tempswa}{#1}%
                            \ifx\FamilyKeyState\FamilyKeyStateProcessed%
                     1941
                              \if@svg@tempswa%
                     1942
                                \FamilyOptions{SVG}{extractdistort=false}%
                     1943
                              \else%
                     1944
                                \FamilyOptions{SVG}{extractdistort=true}%
                     1945
                     1946
                     1947
                            \else%
                     1948
                              \FamilyOptions{SVG}{extractdistort=#1}%
                     1949
                            \fi%
                     1950 }
                     1951 \newcommand*\svgx@param@scale{\svg@param@scale}
                     1952 \DefineFamilyKey{SVG}{extractscale}{%
                            \FamilyKeyStateUnknownValue%
                     1953
                            \svg@ifvalueisrelax{#1}{%
                     1954
                     1955
                              \renewcommand*\svgx@param@scale{1}%
                              \FamilyKeyStateProcessed%
                     1956
                     1957
                     1958
                              \Ifstr{#1}{inherit}{%
                     1959
                                \renewcommand*\svgx@param@scale{\svg@param@scale}%
                     1960
                                \FamilyKeyStateProcessed%
                             }{%
                     1961
                                \Ifisdimension{#1\p@}{%}
                     1962
                                  \ifdim\dimexpr#1\p@\relax>\z@\relax%
                     1963
                                    \renewcommand*\svgx@param@scale{#1}%
                     1964
                     1965
                                    \FamilyKeyStateProcessed%
                     1966
                                }{}%
                     1967
                              }%
                     1969
                           }%
                     1970 }
                     1971 \langle /extract \rangle
                     The similar hooks for executing code right before or after the graphic extraction.
extractpretex (opt.)
\svgx@param@pretex
                     1972 (*main)
extractapptex (opt.)
\svgx@param@apptex
extractpostex (opt.)
```

1911

\Ifstr{#1}{inherit}{%

```
1973 \svg@dummy@key{extractpretex}
1974 \svg@dummy@key{extractapptex}
1975 \svg@dummy@key{extractpostex}
1976 \langle /main \rangle
1977 (*extract)
1978 \newcommand*\svgx@param@pretex{\svg@param@pretex}
1979 \DefineFamilyKey{SVG}{extractpretex}{%
      \svg@ifvalueisrelax{#1}{%
1980
        \let\svgx@param@pretex\relax%
1981
      }{%
1982
        \Ifstr{#1}{inherit}{%
1983
          \renewcommand*\svgx@param@pretex{\svg@param@pretex}%
1984
1985
1986
          \renewcommand*\svgx@param@pretex{#1}%
1987
        }%
1988
      }%
1989
      \FamilyKeyStateProcessed%
1990 }
1991 \newcommand*\svgx@param@apptex{\svg@param@apptex}
1992 \DefineFamilyKey{SVG}{extractapptex}{%
      \svg@ifvalueisrelax{#1}{%
1993
        \let\svgx@param@apptex\relax%
1994
      }{%
1995
1996
        \Ifstr{#1}{inherit}{%
          \renewcommand*\svgx@param@apptex{\svg@param@apptex}%
1997
1998
1999
           \renewcommand*\svgx@param@apptex{#1}%
2000
        }%
2001
      }%
2002
      \FamilyKeyStateProcessed%
2003 }
2004 \DefineFamilyKey{SVG}{extractpostex}{%
      \svg@deprecated@key[svg-extract]{extractpostex=#1}{extractapptex=#1}%
2006 }
2007 (/extract)
```

#### C.1.5. Miscellaneous options

clean (opt.) With option clean files generated during the extraction process can be deleted. Setting true clear (opt.) will remove all files, false won't clear any file. Additionally, a specific file list of suffixes can be given.

```
2008 (*main)
2009 \svg@dummy@key[true]{clean}
2010 \svg@dummy@key[true]{clear}
2011 (/main)
2012 (*extract)
2013 \newcommand*\svgx@clean{}
2014 \DefineFamilyKey{SVG}{clean}[true]{%
      \FamilySetBool{SVG}{clean}{@svg@tempswa}{#1}%
      \ifx\FamilyKeyState\FamilyKeyStateProcessed%
2016
2017
        \if@svg@tempswa%
2018
          \renewcommand*\svgx@clean{log,aux,dvi,out,ps,eps,pdf,\svgx@latex@ext}%
2019
        \else%
          \renewcommand*\svgx@clean{}%
2020
        \fi%
2021
      \else%
2022
2023
        \renewcommand*\svgx@clean{#1}%
2024
        \FamilyKeyStateProcessed%
2025
      \fi%
2026 }
2027 \DefineFamilyKey{SVG}{clear}[true]{\FamilyOptions{SVG}{clean=#1}}
2028 (/extract)
```

exclude (opt.) If it is desired not to include but only extract graphics with package **svg-extract**, option exclude can be used.

```
2029 (*main)
2030 \svg@dummy@key[true]{exclude}
2031 (/main)
2032 (*extract)
2033 \DefineFamilyKey{SVG}{exclude}[true]{%
      \FamilySetBool{SVG}{exclude}{@svg@tempswa}{#1}%
2035
      \ifx\FamilyKeyState\FamilyKeyStateProcessed%
2036
        \if@svg@tempswa%
2037
          \renewcommand*\svg@input[2][]{%
2038
             \if@svgx@run\else%
2039
               \PackageWarning{svg-extract}{%
                 The image '##2' was\MessageBreak%
2040
                 neither extracted nor included%
2041
              }%
2042
             \fi%
2043
2044
          }%
2045
        \else%
2046
          \renewcommand*\svg@input{\svg@@input}%
2047
        \fi%
2048
      \fi%
2049 }
2050 (/extract)
```

## C.2. User commands

extractpretex (param.)
extractapptex (param.)
extractruns (param.)

```
The parameters angle and origin are definied as pendants to the keys provided by
            \includesvg
                          \includegraphics.
        extract (param.)
extractpreamble (param.)
                          2051 (*extract)
  extractformat (param.)
                          2052 \newcommand*\svgx@param@angle{0}
   extractwidth (param.)
                          2053 \svg@local@param@def{%
  {\tt extractheight} \ ({\rm param.})
                                 \DefineFamilyKey[.param]{SVG}{extractangle}{%
                          2054
 extractdistort (param.)
                          2055
                                   \FamilyKeyStateUnknownValue%
   extractscale (param.)
                          2056
                                   \svg@ifvalueisrelax{#1}{%
   {\tt extractangle} \ ({\rm param.})
                          2057
                                      \renewcommand*\svgx@param@angle{0}%
  extractpretex (param.)
                          2058
                                      \FamilyKeyStateProcessed%
  extractapptex (param.)
                          2059
                                   }{%
                                      \Ifstr{#1}{inherit}{%
                          2060
    extractruns (param.)
                                        \renewcommand*\svgx@param@angle{\svg@param@angle}%
       latexexe (param.)
                          2062
                                        \FamilyKeyStateProcessed%
       latexopt (param.)
                          2063
                                     }{%
       latexext (param.)
                                        \Ifisdimension{#1\p@}{%}
                          2064
       {\tt dvipsopt}\ ({\rm param.})
                                          \renewcommand*\svgx@param@angle{#1}%
                          2065
     pstoepsopt (param.)
                          2066
                                          \FamilyKeyStateProcessed%
     pstopdfopt (param.)
                          2067
                                        }{}%
    pdftoepsopt (param.)
                                     }%
                          2068
     {\tt pdftopsopt}\ ({\rm param.})
                          2069
                                   }%
        convert (param.)
                                 }%
                          2070
                          2071 }
  convertformat (param.)
     convertdpi (param.)
                          2072 (/extract)
  magicksetting (param.)
 nasjakapsfeams(estart
                          Some dummys for package svg.
   \svghid&§PBambleend
                          2073 (*main)
       gsdevice (param.)
                          2074 \newcommand*\svghidepreamblestart{%
           clean (param.)
                          2075
                                 \PackageWarning{svg}{%
        exclude (param.)
                                   The macro '\string\svghidepreamblestart' is only meant\MessageBreak%
                          2076
      \includeinkscape
                                   to be used together with package 'svg-extract'. \MessageBreak%
                          2077
        extract (param.)
                          2078
                                   Nevertheless, nothing will happen%
extractpreamble (param.)
                          2079
                                 }%
  extractformat (param.)
                          2080 }
   extractwidth (param.)
  extractheight (param.)
 extractdistort (param.)
   extractscale (param.)
                                                                        61
   extractangle (param.)
```

```
2081 \newcommand*\svghidepreambleend{%
2082 \PackageWarning{svg}{%
2083 The macro '\string\svghidepreambleend' is only meant\MessageBreak%
2084 to be used together with package 'svg-extract'. \MessageBreak%
2085 Nevertheless, nothing will happen%
2086 }%
2087 }
2088 \( /main \)
```

These two macros can be used to hide some parts of the preamble during reading the preamble of the main document.

```
2089 \( \svghidepreamblestart \) 2090 \let\svghidepreambleend\\ \relax 2091 \let\svghidepreambleend\\ \relax 2092 \( /\extract \)
```

# C.3. Auxiliary macros

\svg@extract

The macro \svg@extract does the actual job of both extracting and converting independent graphic files. Since it is necessary to run it with --shell-escape enabled, the command raises a warning if it is not activated. Afterwards, the package is finished.

```
2093 (*main)
2094 \newcommand*\svg@extract[1]{}
2095 (/main)
2096 (*extract)
2097 \ifnum\pdf@shellescape=\@ne\relax\else%
      \renewcommand*\svg@extract[1]{%
2098
        \if@svgx@run%
2099
2100
           \begingroup%
2101
             \edef\svg@tempa{#1}%
2102
             \svg@quotes@remove{\svg@tempa}%
2103
             \PackageWarning{svg-extract}{%
              You didn't enable 'shell escape' (or 'write18')\MessageBreak%
2104
               so it wasn't possible to run the extraction for\MessageBreak%
2105
              file '\svg@tempa'\MessageBreak%
2106
2107
            }%
          \endgroup%
2108
2109
        \fi%
      }%
2110
2111
      \expandafter\endinput%
2112 \fi
2113 (/extract)
```

\svgx@stream@in \svgx@read@line \svgx@stream@out \if@svgx@preamble@write Both an input stream and an output stream are necessary for this as well as a token register, which is used to store all commands which should be executed on shell.

```
2114 \newread\svgx@stream@in
2115 \newcommand*\svgx@read@line{}
2116 \newwrite\svgx@stream@out
2117 \newif\if@svgx@preamble@write
```

\svg@extract

If flag --shell-escape is enabled, the command is defined with its intended functionality. It runs all necessary processes to extract and convert graphic files.

```
2118 \renewcommand*\svg@extract[1]{%
```

If option extract is enabled...

```
2119 \if@svgx@run%
```

... the macro \svgx@get@out@sec is used to get the current level numbering within the document and the counter for extracted graphics is stepped. After that, a separate auxiliary IATEX file is created for extracting independent graphic files. The macro \svgx@get@out@sec is used to get the current level numbering within the document. The specified preamble is read for this task, if it exists. It is first searched in the same folder as the SVG file and if it wasn't found, in any other valid folder for SVG files.

```
2120
        \if@svgx@out@sec%
2121
          \svgx@get@out@sec%
        \fi%
2122
2123
        \stepcounter{svgx@out@count}%
2124
        \begingroup%
2125
          \def\svg@tempa##1.##2\@nil{%
2126
             \IfArgIsEmpty{##2}{\edef\svgx@preamble{##1.\svgx@latex@ext}}{}%
2127
           \expandafter\svg@tempa\svgx@preamble.\@nil%
2128
          \IfFileExists{\svg@file@path\svgx@preamble}{%
2129
2130
            \@svg@file@foundtrue%
2131
          }{%
             \svg@get@path[]{\svgx@preamble}{\svg@out@path}%
2132
            \def\svg@tempa###1.###2\@nil{%
2133
2134
               \edef\svgx@preamble{\svg@file@name.###2}%
2135
            }%
             \expandafter\svg@tempa\svgx@preamble\@nil%
2136
2137
          }%
2138
          \edef\svg@tempa{%
            \endgroup%
2139
2140
            \if@svg@file@found%
2141
               \ifx\svg@file@path\@empty%
                 \def\noexpand\svgx@preamble{./\svgx@preamble}%
2142
               \else%
2143
                 \def\noexpand\svgx@preamble{\svg0file@path\svgx@preamble}%
2144
               \fi%
2145
            \fi%
2146
          }%
2147
2148
        \svg@tempa%
2149
        \begingroup%
          \endlinechar=\m@ne%
2150
2151
          \IfFileExists{\svgx@preamble}{%
2152
            \PackageInfo{svg-extract}{%
2153
               The preamble file '\svgx@preamble'\MessageBreak%
2154
               is used for the generation of the auxiliary file\MessageBreak%
2155
               '\svgx@out@name.\svgx@latex@ext'%
            }%
2156
```

The catcodes for # need to be changed to prevent doubling when reading the line.

```
2157 \catcode'\#=12\relax%
2158 \immediate\openout\svgx@stream@out=\svgx@out@name.\svgx@latex@ext%
2159 \immediate\openin\svgx@stream@in=\svgx@preamble%
2160 \@svg@tempswatrue%
2161 \@svgx@preamble@writetrue%
2162 \def\svgx@read@line{}%
```

The given preamble file is read line by line and written to the separate auxiliary IATEX file \svgx@out@name.\svgx@latex@ext via the output stream.

```
2163 \@whilesw\if@svg@tempswa\fi{%}
2164 \immediate\read\svgx@stream@in to\svgx@read@line%
2165 \ifx\svgx@read@line\@empty%
2166 \ifeof\svgx@stream@in\@svg@tempswafalse\fi%
2167 \else%
```

With \svghidepreamblestart and \svghidepreambleend it is possible for the user to omit certain parts of the preamble. Therefor the two macros \svgx@read@preamble@till and

```
2168 \svgx@read@preamble@till{\svghidepreamblestart}{}%
2169 \svgx@read@preamble@from{\svghidepreambleend}{}%
```

If the desired end of the preamble (\svgxQendpreamble) was found, the readout is terminated by switching \ifQsvgQtempswa to false.

```
2170 \svgx\text{Qread\text{Qpreamble\text{Qsvg\text{Qtempswafalse}}}} \
2171 \if\text{if\text{Qsvg\text{Qpreamble\text{Qwrite}}}}
```

During the readout process, it is searched with \svgx@documentclass for the appearance of \documentclass and \if@svgx@classfound is set to true if it was found.

```
2172 \if@svgx@classfound\else%
2173 \expandafter\svgx@documentclass%
2174 \svgx@read@line\documentclass\documentclass\@nil%
2175 \fi%
```

Writing out the—maybe manipulated—read in line.

```
\ifx\svgx@read@line\@empty\else%
2177
                     \immediate\write\svgx@stream@out{%
2178
                       \unexpanded\expandafter{\svgx@read@line}%
2179
                     }%
                   \fi%
2180
                 \fi%
2181
               \fi%
2182
             }%
2183
             \immediate\closein\svgx@stream@in%
2184
2185
             \immediate\closeout\svgx@stream@out%
             \catcode'\#=6\relax%
2186
```

Once the separate auxiliary LATEX file is written, it is read in again and its content is stored in \svg@tempa, since it is necessary to prepend some stuff to the preamble, for example a maybe not existent document class.

```
2187
            \immediate\openin\svgx@stream@in=\svgx@out@name.\svgx@latex@ext%
2188
             \def\svg@tempa{}%
            \loop\unless\ifeof\svgx@stream@in%
2189
2190
               \readline\svgx@stream@in to\svgx@read@line%
2191
               \ifx\svgx@read@line\@empty\else%
2192
                 \edef\svg@tempa{%
                   \unexpanded\expandafter{\svg@tempa}%
2193
2194
                   \unexpanded\expandafter{\svgx@read@line}^^J%
                }%
2195
2196
               \fi%
            \repeat%
2197
2198
            \immediate\closein\svgx@stream@in%
          }{%
2199
```

If a file was given that doesn't exist, a warning is issued.

```
2200
             \svg@quotes@remove{\svgx@preamble}%
             \ifx\svgx@preamble\@empty\else%
2201
               \verb|\PackageWarning{svg-extract}{%}|
2202
2203
                 The preamble file '\svgx@preamble'\MessageBreak%
2204
                 does not exist%
               }%
2205
2206
             fi%
2207
             \def\svg@tempa{}%
2208
```

After the preamble was read in and stored in \svg@tempa, the separate auxiliary LATEX file is written again. Some information are written right at the beginning of the file.

```
2209 \immediate\openout\svgx@stream@out=\svgx@out@name.\svgx@latex@ext%
2210 \immediate\write\svgx@stream@out{%
2211 \@percentchar\@percentchar\space This file was generated by package
2212 'svg-extract'^^J%
2213 \@percentchar\@percentchar\space from source '\jobname'^^J%
2214 \@percentchar\@percentchar\space It's intended to be compiled with
2215 '\svgx@latex@exe\ifx\svgx@latex@opt\@empty\else\space\svgx@latex@opt\fi'
2216 }%
```

With the intention of passing the correct paper dimensions, the calculating of the paper size is executed with \AtBeginDocument even before the document class, so that this is definitely the first thing to happen at the beginning of the document. Additionally, it is ensured that the \special command is definitely used with the correct paper size, when creating a DVI file.

```
2217
          \immediate\write\svgx@stream@out{%
            \string\AtBeginDocument{\@percentchar^^J%
2218
2219
              \space\space\string\svgxsetpapersize\@percentchar^^J%
              \ifxetex\else\ifpdf\else%
2220
2221
                 \space\space\string\AtBeginDvi{\string\special{%
                     papersize=\string\the\string\paperwidth,%
2222
                       \string\the\string\paperheight%
2223
                }}\@percentchar^^J%
2224
2225
              fi\fi
            }^^J%
2226
2227
            \string\PassOptionsToPackage{hidelinks}{hyperref}%
2228
```

If no document class was found during reading the preamble file, then class \article is used.

```
2229 \if@svgx@classfound\else%
2230 \immediate\write\svgx@stream@out{\string\documentclass{article}}%
2231 \fi%
```

And now the stored preamble.

```
2232 \ifx\svg@tempa\@empty\else%
2233 \immediate\write\svgx@stream@out{\unexpanded\expandafter{\svg@tempa}}%
2234 \fi%
```

After the given preamble was written, package **svg-extract** will be loaded in case it was forgotten.

```
2235 \immediate\write\svgx@stream@out{\string\usepackage{svg-extract}}%
```

Now all parameters relevant for the extraction are evaluated and appended.

```
\def\svg@tempa##1{%
2236
            \immediate\write\svgx@stream@out{\string\svgsetup{##1}}%
2237
          ጉ%
2238
2239
          \if@svg@ink@latex\else%
2240
            \svg@tempa{inkscapelatex=false}%
2241
          \ifdim\svgx@param@width>\z@\relax%
2242
            \svg@tempa{width=\svgx@param@width}%
2243
2244
          \fi%
2245
          \ifdim\svgx@param@height>\z@\relax%
2246
             \svg@tempa{height=\svgx@param@height}%
          \fi%
2247
          \if@svgx@param@distort%
2248
            \svg@tempa{distort=true}%
2249
2250
          \fi%
2251
          \ifdim\dimexpr\svgx@param@scale\p@\relax=\p@\relax\else%
```

```
2252
            \svg@tempa{scale=\svgx@param@scale}%
2253
          \fi%
2254
          \def\svg@tempb{\svg@param@pretex}%
          \ifx\svgx@param@pretex\svg@tempb\relax%
2255
            \let\svgx@param@pretex\svg@param@pretex%
2256
2257
2258
          \ifx\svgx@param@pretex\relax\else%
            \svg@tempa{pretex=\unexpanded\expandafter{\svgx@param@pretex}}%
2259
          \fi%
2260
          \def\svg@tempb{\svg@param@apptex}%
2261
          \ifx\svgx@param@apptex\svg@tempb\relax%
2262
            \let\svgx@param@apptex\svg@param@apptex%
2263
2264
2265
          \ifx\svgx@param@apptex\relax\else%
2266
            \svg@tempa{apptex=\unexpanded\expandafter{\svgx@param@apptex}}%
2267
```

Parameter lastpage is only considered for including PDF files with LATEX support.

```
\let\svg@tempa\@empty%
2268
          \if@svg@ink@latex%
2269
2270
            \Ifstr{\svg@ink@format}{pdf}{%
2271
               \ifnum\value{svg@param@lastpage}>\z@\relax%
                 \edef\svg@tempa{lastpage=\the\value{svg@param@lastpage}}%
2272
2273
2274
                 \ifnum\value{svg@param@lastpage}=\z@\relax%
2275
                   \def\svg@tempa{lastpage=true}%
2276
                 \else%
                   \def\svg@tempa{lastpage=false}%
2277
                 \fi%
2278
2279
               \fi%
2280
            }{}%
2281
          \fi%
```

The rotation angle, if given.

```
2282 \ifdim\dimexpr\svgx@param@angle\p@\relax=\z@\relax\else%
2283 \edef\svg@tempa{%
2284 angle=\svgx@param@angle\ifx\svg@tempa\@empty\else,\svg@tempa\fi%
2285 }%
2286 \fi%
```

As we are now at the end of the preamble and just before the beginning of the document, the paper dimension are set again to make sure, that these settings are active at the end of the preamble. Additionally, it is executed again at the very end of \AtBeginDocument to ensure, that no other package used this hook for manipulating the paper size.

```
2287 \ifx\svg@tempa\@empty%
2288 \def\svg@tempa{\string\svgxsetbox{#1}}%
2289 \else%
2290 \edef\svg@tempa{\noexpand\string\noexpand\svgxsetbox[\svg@tempa]{#1}}%
2291 \fi%
2292 \immediate\write\svgx@stream@out{\svg@tempa}%
```

Package xr is used to evaluate possible labels within the included Inkscape LATEX file.

```
2293
          \if@svg@ink@latex%
2294
            \IfFileExists{xr.sty}{%
               \immediate\write\svgx@stream@out{%
2295
                 \string\usepackage{xr}^^J%
2296
                 \string\externaldocument{\jobname}^^J%
2297
              }%
2298
2299
            }{}%
2300
          \fi%
          \immediate\write\svgx@stream@out{%
2301
            \string\begin{document}^^J%
2302
```

```
2303 \string\pagestyle{empty}^^J%
2304 \string\svgxoutputbox\@percentchar^^J%
2305 \string\end{document}%
2306 }%
2307 \immediate\closeout\svgx@stream@out%
2308 \endgroup%
```

After creating the separate auxiliary IATEX file, the actual extraction and conversion can be done.

As the extraction maybe needs to include the main auxiliary file with \externaldocument provided by package **xr** it is necessary to do all related stuff after the main auxiliary file was written. This is done with \afterReadingMainAux provided by package **scrIfile**.

```
2315 \svg@quotes@remove{\svgx@out@path}%
2316 \svg@quotes@remove{\svgx@out@name}%
```

All generated files will be moved to the desired output folder, which is given by option extractpath. Therefor, this folder is created.

```
2317 \edef\svg@tempb{%

2318 \noexpand\svg@shell@mkdir{\svgx@out@path}%

2319 }%

2320 \expandafter\AfterReadingMainAux\expandafter{\svg@tempb}%
```

First of all the separate auxiliary IATEX file is compiled with the detected IATEX engine (\svgx@latex@exe) as often as defined by counter option extractruns.

```
2321
          \edef\svg@tempb{%
            \noexpand\PackageInfo{svg-extract}{%
2322
2323
              Running LaTeX (\svgx@latex@exe) for graphic extraction%
2324
              \ifx\svgx@latex@opt\@empty\else%
2325
                 \MessageBreak with added options '\svgx@latex@opt'%
              \fi%
2326
2327
            }%
2328
          }%
2329
          \expandafter\AfterReadingMainAux\expandafter{\svg@tempb}%
2330
          \edef\svg@tempb{%
2331
            \noexpand\ShellEscape{%
2332
               \svgx@latex@exe\space\svgx@latex@opt\space%
               "\svgx@out@name.\svgx@latex@ext"%
2333
            }%
2334
2335
          }%
2336
          \loop\ifnum\value{svgx@runs}>\z@\relax%
            \expandafter\AfterReadingMainAux\expandafter{\svg@tempb}%
2337
2338
            \advance\c@svgx@runs\m@ne%
2339
          \repeat%
```

All files requested with option extractformat are created with internal conversion tools supplied by most LaTeX distributions if necessary.

```
2348
                                                                      \expandafter\AfterReadingMainAux\expandafter{\svg@tempb}%
2349
                                                          }%
2350
                                                           \@svg@tempswafalse%
                                                           \ifxetex\else\ifpdf\else%
2351
2352
                                                                      \@svg@tempswatrue%
2353
                                                           \fi\fi%
2354
                                                          \if@svg@tempswa%
                                                                      \svg@tempa{dvips}{dvi}{ps}%
2355
                                                                      \label{lem:list_eps} $$\sup_{\text{svgx@format}}{\sup_{\text{pstoeps}}{ps}_{eps}}}{\} % $$\sum_{\text{pstoeps}}{ps}_{eps}}{\} % $$\sum_{\text{pstoeps}}{ps}_{e
2356
                                                                      \svgx@ifinlist{pdf}{\svgx@format}{\svg@tempa{pstopdf}{ps}{pdf}}{}%
2357
                                                           \else%
2358
                                                                      \svgx@ifinlist{eps}{\svgx@format}{\svg@tempa{pdftoeps}{pdf}{eps}}{}%
2359
2360
                                                                       \svgx@ifinlist{ps}{\svgx@format}{\svg@tempa{pdftops}{pdf}{ps}}{}%
2361
```

Now the desired conversion tool is invoked if requested.

```
2362 \if@svgx@cnv@run%
```

If no density was given at all, the density for PNG files is set to 300dpi by default.

```
2363 \ifx\svgx@cnv@dpi\relax%

2364 \ifx\svgx@cnv@dpi@png\@undefined%

2365 \def\svgx@cnv@dpi@png{300}%

2366 \fi%

2367 \fi%
```

The first given file type with option extractformat is used as source for the conversion process.

```
2368 \expandafter\svgx@cnv@get@informat\expandafter{\svgx@format}%
```

The conversion is done for each desired file type given in a list by option convertformat.

```
\@for\svg@tempa:=\svgx@cnv@format\do{%
              \ifx\svg@tempa\@empty\else%
2370
                 \expandafter\svgx@ifinlist\expandafter{\svg@tempa}{\svgx@format}{%
2371
2372
                   \PackageWarning{svg-extract}{%
2373
                     File type '\svg@tempa' was specified for option\MessageBreak%
2374
                     'extractformat' (\svgx@format) as well as for \MessageBreak%
                     option 'convertformat' (\svgx@cnv@format) so the\MessageBreak%
2375
                     conversion won't be done%
2376
                  }%
2377
                }{%
2378
2379
                   \edef\svg@tempb{%
2380
                     \noexpand\PackageInfo{svg-extract}{%
                       Converting '\svgx@out@name.\svgx@cnv@informat'\MessageBreak%
2381
                       to '\svgx@out@name.\svg@tempa'%
2382
2383
                     }%
2384
                   }%
                   \expandafter\AfterReadingMainAux\expandafter{\svg@tempb}%
2385
                   \edef\svg@tempb{%
2386
                     \noexpand\ShellEscape{%
2387
                       \svgx@cnv@cmd{\svgx@out@name}{\svgx@cnv@informat}{\svg@tempa}%
2388
2389
                     }%
                   }%
2390
                   \expandafter\AfterReadingMainAux\expandafter{\svg@tempb}%
2391
                }%
2392
              \fi%
2393
            }%
2394
2395
          \fi%
```

As both extraction and conversion are done, all files are moved to the desired output folder (extractpath).

```
2396 \edef\svg@tempa{\svgx@format\if@svgx@cnv@run,\svgx@cnv@format\fi}%
2397 \@for\svg@tempb:=\svg@tempa\do{%
```

```
2398 \ifx\svg@tempb\@empty\else%
2399 \edef\svg@tempb{%
2400 \noexpand\svgx@move{\svgx@out@name}{\svg@tempb}{\svgx@out@path}%
2401 }%
2402 \expandafter\AfterReadingMainAux\expandafter{\svg@tempb}%
2403 \fi%
2404 }%
```

At the very end, all unwanted auxiliary files are deleted.

```
\@for\svg@tempa:=\svgx@clean\do{%
2405
2406
            \expandafter\svgx@ifinlist\expandafter{\svg@tempa}{\svg@tempb}{}{%
2407
               \edef\svg@tempb{%
                 \noexpand\IfFileExists{"\svgx@out@name".\svg@tempa}{%
2408
                   \noexpand\svg@shell@rm{\svgx@out@name.\svg@tempa}%
2409
                }{}%
2410
              }%
2411
               \expandafter\AtEndDocument\expandafter{%
2412
                 \expandafter\AfterReadingMainAux\expandafter{\svg@tempb}%
2413
              }%
2414
            }%
2415
2416
          }%
2417
        }%
2418
      \fi%
2419 }
```

\svgx@get@out@sec

The macro \svgx@get@out@sec reads all sectioning counters in order to get the numbering of the current sectioning level. The value is stored in \svgx@out@sec.

```
2420 \newcommand*\svgx@get@out@sec{%
2421
      \begingroup%
2422
         \def\svg@tempa{}%
2423
        \@for\svg@tempb:={%
2424
          part, chapter, section, subsection, subsubsection, paragraph, subparagraph%
2425
        }\do{%
           \ifx\svg@tempb\@empty\else%
2426
2427
             \scr@ifundefinedorrelax{the\svg@tempb}{}{%
2428
               \ifnum\value{\svg@tempb}>\z@\relax%
2429
                 \edef\svg@tempa{\svg@tempb}%
2430
             }%
2431
2432
          \fi%
        }%
2433
         \edef\svg@tempb{%
2434
          \endgroup%
2435
2436
          \ifx\svg@tempa\@empty\else%
2437
             \def\noexpand\svgx@out@sec{\csname the\svg@tempa\endcsname}%
2438
          \fi%
        }%
2439
2440
      \svg@tempb%
2441 }
```

\svgx@documentclass \if@svgx@classfound This delimited macro is used to find the occurrence of \documentclass within a read in line. The delimiter \documentclass is used twice in order to ignore the possible occurrence of white space or anything else right before \documentclass.

```
2442 \newif\if@svgx@classfound  
2443 \newcommand*\svgx@documentclass{}  
2444 \def\svgx@documentclass#1\documentclass#2\documentclass#3\@nil{%  
2445 \IfArgIsEmpty{#2}{}{\@svgx@classfoundtrue}%  
2446 }
```

\svgx@read@preamble@till \svgx@read@preamble@from \svgx@read@preamble@skip These macros are used to skip some parts of a read in preamble file.

2447 \newcommand\*\svgx@read@preamble@till[2]{\%

```
2448 \svgx@read@preamble@skip#1\@nil{till}{#2}%
2449 }
2450 \newcommand*\svgx@read@preamble@from[2]{%
2451 \svgx@read@preamble@skip#1\@nil{from}{#2}%
2452 }
```

In principle, the functionality is the same as for \svgx@documentclass.

```
2453 \newcommand*\svgx@read@preamble@skip{}
2454 \def\svgx@read@preamble@skip#1\@nil#2#3{%
```

A given token is used to create the macro \svg@tempa delimited by the token itself which is used twice to get any stuff right before or after the occurrence.

```
2455 \def\svg@tempa##1{%
2456 \def\svg@tempa###1##1###2##1###3\@ni1{%
2457 \IfArgIsEmpty{###3}{}{%
```

Write everything which was found right before the macro which starts hiding area to the output stream and stop writing with \ifGsvgx@preamble@write.

```
2458 \Ifstr{#2}{till}{%

2459 \IfArgIsEmpty{####1}{}{%

2460 \immediate\write\svgx@stream@out{####1}%

2461 }%

2462 \@svgx@preamble@writefalse%

2463 }{%
```

Write everything which was found right after the macro which ends the hiding area and start writing again with \if@svgx@preamble@write.

```
\Ifstr{#2}{from}{%
2464
                  \IfArgIsEmpty{####2}{%
2465
2466
                    \def\svgx@read@line{}%
2467
                 }{%
2468
                    \def\svgx@read@line{####2}%
2469
                 }%
2470
                  \@svgx@preamble@writetrue%
2471
               }{}%
             }%
2472
```

Additional stuff which should be done.

```
2473 #3%
2474 }%
2475 }%
2476 }%
```

Creating the macro \svg@tempa delimited by the first argument.

```
2477 \edef\svg@tempb{\expandafter\detokenize\expandafter{#1}}%
2478 \expandafter\svg@tempa\expandafter{\svg@tempb}%
```

Calling the created macro.

```
2479 \edef\svg@tempb{%
2480 \expandafter\detokenize\expandafter{\svgx@read@line}\svg@tempb\svg@tempb\%
2481 }%
2482 \expandafter\svg@tempa\svg@tempb\@nil%
2483 }
```

\svgx@cnv@informat \svgx@cnv@get@informat The first list entry from argument (\svgx@format) is extracted by \svgx@cnv@get@informat.

```
2484 \newcommand*\svgx@cnv@informat{}
2485 \newcommand*\svgx@cnv@get@informat[1]{%
2486 \begingroup%
2487 \def\svg@tempa##1,##2\@nil{%
```

```
2488
                       \def\svg@tempa{##1}%
            2489
                     }%
                     \svg@tempa#1,\@nil%
            2490
                     \edef\svg@tempa{%
            2491
            2492
                       \endgroup%
            2493
                       \def\noexpand\svgx@cnv@informat{\svg@tempa}%
            2494
                     ጉ%
            2495
                   \svg@tempa%
            If the first argument (\svgx@format) was empty, \svgx@cnv@informat is set to the a file
            type, which is generated anyway.
                   \ifx\svgx@cnv@informat\@empty%
                     \renewcommand*\svgx@cnv@informat{pdf}%
            2497
            2498
                     \ifxetex\else\ifpdf\else%
            2499
                       \renewcommand*\svgx@cnv@informat{ps}%
            2500
                     \fi\fi%
            2501
                   \fi%
            2502 }
            If the file doesn't exist
\svgx@move
            2503 \newcommand*\svgx@move[3]{%
                   \begingroup%
            2504
                     \IfFileExists{"#1".#2}{%
            2505
                       \svg@shell@mv{#1.#2}{#3#1.#2}%
            2506
            2507
                     }{%
                       \edef\svg@tempa{#2}%
            2508
                       \@svg@tempswafalse%
            2509
            2510
                       \expandafter\svgx@ifinlist\expandafter{\svg@tempa}{\svgx@cnv@format}{%
            2511
                         \@svg@tempswatrue%
            2512
                         \def\svg@tempb{conversion}%
            2513
                       }{%
                         \expandafter\svgx@ifinlist\expandafter{\svg@tempa}{pdf,ps,eps}{%
            2514
                           \@svg@tempswatrue%
            2515
            2516
                           \def\svg@tempb{extraction}%
            2517
                         }{}%
            2518
                       }%
                       \if@svg@tempswa%
            2519
                         \edef\svg@tempb{%
            2520
                           The graphic file \svg@tempb\space failed\MessageBreak%
            2521
            2522
                           for '#1.#2'\MessageBreak%
                           Troubleshooting: Please check the log file how\MessageBreak%
            2523
                           the invocation of the extraction took place and\MessageBreak%
            2524
            2525
                           try to execute it yourself in the terminal%
                         }%
            2526
                       \else%
            2527
            2528
                         \def\svg@tempb{%
                           The extraction to format '#2' failed\MessageBreak%
            2529
                           for '#1.#2'\MessageBreak%
            2530
                           Only file types 'pdf,ps,eps' are supported for\MessageBreak%
            2531
            2532
                           key 'exportformat'%
                         }%
            2533
                       \fi%
            2534
            2535
                       \PackageWarning{svg-extract}{\svg@tempb}%
                     ጉ%
            2536
            2537
                   \endgroup%
```

# C.4. Commands for the separate auxiliary LETEX-file

2538 }

For the extraction of independent graphics, an auxiliary LATEX file is needed. Within this file, the following commands are used to include the desired graphic.

\svgxsetbox \svgx@setbox \if@svgx@standalone Within the preamble of the auxiliary LATEX file, the desired graphic is used to setup a box, which is used both to define the papersize as well as for the output itself. The macro \svgx@setbox is executed twice, the first time in the preamble and the second time at the very end of \AtBeginDocument if package etoolbox was loaded.

The switch \if@svgx@standalone is defined for enabling classes to implement a different behavoiur for svg-extract in standalone mode. for example, TUD-Script-classes are using this switch.

```
2539 \newif\if@svgx@standalone
2540 \newcommand*\svgxsetbox[2][]{%
     \@svgx@standalonetrue%
2542
     \svgx@setbox{#1}{#2}%
2543
     \scr@ifundefinedorrelax{AtEndPreamble}{%
2544
       \let\svg@tempa\@firstofone%
     }{%
2545
2546
       \def\svg@tempa{\AtEndPreamble}%
     }%
2547
2548
     2549 }
2550 \newcommand*\svgx@setbox[2]{%
2551
     \sbox\svg@box{\svg@@input[{#1},draft=false]{#2}}%
     \svgxsetpapersize%
2553 }
```

\svgxsetpapersize

This macro sets all well known length macros for defining the paper size as well as the type area to the size of \svg@box.

```
2554 \newcommand*\svgxsetpapersize{%
2555 \setlength\paperwidth{\the\wd\svg@box}%
```

Due to the fact, that the lengths for stock- and mediasizes are maybe set to \relax, these macros are checked with \scr@ifundefinedorrelax.

```
\scr@ifundefinedorrelax{stockwidth}{}{%
2556
        \setlength\stockwidth{\paperwidth}%
2557
      }%
2558
      \scr@ifundefinedorrelax{mediawidth}{}{%
2559
2560
        \setlength\mediawidth{\paperwidth}%
2561
      \setlength\textwidth{\paperwidth}%
2562
      \setlength\paperheight{\the\dimexpr\ht\svg@box+\dp\svg@box\relax}%
2563
2564
      \scr@ifundefinedorrelax{stockheight}{}{%
2565
        \setlength\stockheight{\paperheight}%
      }%
2566
2567
      \scr@ifundefinedorrelax{mediaheight}{}{%
2568
        \setlength\mediaheight{\paperheight}%
2569
2570
      \setlength\textheight{\paperheight}%
```

Any other length regarding the layout is set to have no influence at all. Hence the document has the same size as the graphic.

```
2571
      \hoffset=-1in%
2572
      \oddsidemargin=\z0%
2573
      \evensidemargin=\z0%
2574
      \voffset=-1in%
      \topmargin=\z0%
2575
      \theta = z0\%
2576
      \headsep=\z@%
2577
      \t = \z 0\%
2578
2579
      \footskip=\z0%
2580
      \marginparsep=\z0%
2581
      \marginparwidth=\z0%
2582
      \marginparpush=\z0%
2583 }
```

\svgxoutputbox \if@svgx@beamer With \svgxoutputbox the created box is displayed.

```
2585 \newif\if@svgx@beamer
2586 \@ifclassloaded{beamer}{\@svgx@beamertrue}{}%
2587 \newcommand*\svgxoutputbox{%
2588
      \begingroup%
         \setlength\parindent{\z0}%
2589
         \setlength\parskip{\z@}%
2590
        \verb|\setlength| parfillskip{\z@}%
2591
2592
        \if@svgx@beamer%
          \setbeamertemplate{navigation symbols}{}%
2593
          \begin{frame}[plain]%
2594
          \usebox\svg@box%
2595
2596
          \end{frame}%
2597
        \else%
          \usebox\svg@box%
2598
        \fi%
2599
         \endgraf%
2600
2601
      \endgroup%
2602 }
```

# D. Processing Options

Setting the default options and processing the given ones during when loading the packages.

```
2603 (*main)
2604 \FamilyExecuteOptions{SVG}{%
     inkscape=true,inkscapeversion=auto,inkscapepath=basesubdir,%
      inkscapelatex=true,inkscapearea=drawing,distort=false,%
2607
     usexcolor=true,usetransparent=true%
2608 }
2609 (/main)
2610 (*extract)
2611 \FamilyExecuteOptions{SVG}{%
     extract=true,extractpath=basesubdir,%
      extractruns=2,extractname=namenumbered,extractdistort=false,%
2613
2614
      convert=magick,convert=false,%
      gsdevice={png=png16m},gsdevice={jpeg=jpeg},gsdevice={jpg=jpeg},%
2615
      gsdevice={tif=tiff48nc},gsdevice={tiff=tiff48nc},%
2617
      gsdevice={eps=eps2write},gsdevice={ps=ps2write}%
2618 }
2619 (/extract)
2620 \FamilyProcessOptions{SVG}
```

# Index

Numbers written in italic refer to the page where the corresponding entry is described. Numbers underlined refer to the code line of the definition.

Α	\if@svgx@classfound 2442
$\mathtt{apptex} \; (\mathtt{opt.}) \; \ldots \ldots \qquad \qquad$	\if@svgx@cnv@run <u>1528</u>
	\if@svgx@out@sec <u>1836</u>
С	\if@svgx@preamble@write 2114
clean (opt.)	\if@svgx@run 1324
clear (opt.)	\if@svgx@standalone 2539
convert (opt.)	\includeinkscape $7-8$ , $10$ , $\overline{761}$ , $\underline{2051}$
convertdensity (opt.) $\underline{1631}$	angle (param.)
convertdpi (opt.)	apptex (param.)
convertformat (opt.)	$\verb clean  (param.)$
counters:	$\mathtt{convert} \; (\mathtt{param.})  \dots \qquad \underline{2051}$
svg@param@lastpage	$\texttt{convertdpi} \; (\texttt{param.}) \; \dots \dots \qquad \underline{2051}$
svgx@out@count 1836	$\texttt{convertformat} \; (\texttt{param.})  \dots  \underline{2051}$
svgx@runs <u>1424</u>	distort (param.)
D	draft (param.)
distort (opt.) 6, <u>555</u>	dvipsopt (param.) 2051
draft (opt.)	exclude (param.) <u>2051</u>
dvipsopt (opt.)	extract (param.) <u>2051</u>
<b>avipsops</b> (opt.)	extractangle (param.) $10, \underline{2051}$
E	extractapptex (param.)
end (opt.)	extractdistort (param.) 2051
eps (opt.) <u>1356</u>	extractformat (param.)
exclude (opt.)	extractheight (param.) 2051
<b>ext</b> (opt.)	extractpreamble (param.)         2051           extractpretex (param.)         2051
extension (opt.) $\underline{517}$	extractruns (param.)
$\mathtt{extract} \; (\mathtt{opt.}) \; \ldots \; \ldots \; \delta,  \underline{1324}$	extractscale (param.)
extractapptex (opt.) $\dots \dots 9, \underline{1972}$	extractwidth (param.) 2051
extractdistort (opt.) $0, 1876$	gsdevice (param.) 2051
extractformat (opt.) $9, \underline{1356}$	gsopt (param.) 2051
extractheight (opt.) $9, 1876$	height (param.) 7, 802
extractkeepaspectratio (opt.) 1876 extractname (opt.) 8, 1836	inkscapeformat (param.) $7, 802$
extractpath (opt.)	inkscapelatex (param.) $7, 802$
extractpostex (opt.)	lastpage (param.)
extractpreamble (opt.) 9, 1397	latexexe (param.) 2051
extractpreambleend (opt.) $9$ , $\overline{1397}$	latexext (param.)
extractpretex (opt.) 9, <u>1972</u>	latexopt (param.)
extractruns (opt.)	magickoperator (param.) 2051
extractscale (opt.) $9, \underline{1876}$	magicksetting (param.)
extractwidth (opt.)	origin (param.)
•	pdftopsopt (param.)
G 17.00	pretex (param.)
gsdevice (opt.)	pstoepsopt (param.) 2051
gsopt (opt.)	pstopdfopt (param.) 2051
<b>Spoke</b> (open)	<b>scale</b> (param.)
Н	width (param.)
$\texttt{height} \; (\texttt{opt.}) \; \ldots \ldots \qquad \qquad \boldsymbol{6},  \underline{555}$	\includesvg $6, 8, 10, \underline{709}, \underline{2051}$
	angle (param.)
1	apptex (param.) $6, \underline{712}$
\if@svg@draft	clean (param.)
\if@svg@file@found	convert (param.)
\if@svg@ink@run	convertdpi (param.)         2051           convertformat (param.)         2051
\if @svg@param@distort	convertformat (param.)         2051           distort (param.)         6,712
\if @svg@quotes@found	draft (param.)
\if@svg@tempswa 17	dvipsopt (param.)
\if@svg@use@transparent 286	exclude (param.) 2051
\if@svg@use@xcolor 286	extract (param.) 2051
\if@svgx@beamer 2585	extractangle (param.) $\dots \dots 10, \overline{2051}$

extractapptex (param.) 2051	on (opt.)
extractdistort (param.) 2051	options:
extractformat (param.) 2051	apptex
extractheight (param.) 2051	clean 10, <u>2008</u>
extractpreamble (param.) 2051	clear <u>2008</u>
extractpretex (param.) 2051	convert
extractruns (param.)	convertdensity 1631
extractscale (param.)	convertdpi
extractwidth (param.)	convertformat
gsdevice (param.)	distort
gsopt (param.)	draft
height (param.)	dvipsopt
inkscape (param.) $6, \frac{712}{712}$	end
inkscapearea (param.) 6, 712	eps
inkscapedpi (param.) $6$ , $712$ inkscapeformat (param.) $6$ , $712$	exclude
inkscapelatex (param.) 6, 712	ext
inkscapeopt (param.) $6, \frac{112}{712}$	extract
lastpage (param.)	extractapptex 9, 1972
latexexe (param.)	extractdistort 9, 1876
latexext (param.) 2051	extractformat $9, 1356$
latexopt (param.) 2051	extractheight 9, <u>1876</u>
magickoperator (param.) 2051	extractkeepaspectratio 1876
magicksetting (param.) 2051	extractname 8, 1836
origin (param.)	extractpath $8, 1802$
pdftoepsopt (param.) 2051	extractpostex $\dots \dots \dots$
pdftopsopt (param.) 2051	extractpreamble $0.00000000000000000000000000000000000$
pretex (param.)	extractpreambleend $\dots \dots g$ , $\overline{1397}$
pstoepsopt (param.) 2051	extractpretex $0, 1972$
pstopdfopt (param.) 2051	extractruns $\dots \dots 10, \underline{1424}$
scale (param.) $6, \underline{712}$	extractscale $9, \underline{1876}$
sygextension (param.) $6, \underline{712}$	extractwidth $\dots 9, \underline{1876}$
width (param.)	gsdevice $\dots 12, \underline{1769}$
$\mathtt{inkscape} \; (\mathtt{opt.})  \dots  \dots  \   4,  \underline{315}$	gsexe
inkscapearea (opt.)	gsopt
${\tt inkscapedensity} \; ({\tt opt.})  \dots \qquad \underline{476}$	$\texttt{height}  \dots  \dots  \qquad \qquad$
inkscapedpi (opt.)	inkscape $\dots 4, \underline{315}$
inkscapeexe (opt.)	inkscapearea $5, \underline{462}$
inkscapeformat (opt.)	inkscapedensity $\dots \frac{476}{178}$
inkscapelatex (opt.) 5, <u>458</u>	inkscapedpi 6, <u>476</u>
inkscapename (opt.) $5, \underline{549}$	inkscapeexe 5, <u>397</u>
inkscapeopt (opt.) $6, \underline{397}$	inkscapeformat $\dots \dots \dots$
inkscapepath (opt.)	inkscapelatex
inkscapeversion (opt.) $5, \underline{397}$	inkscapename
14	
K	inkscapeopt
K keepaspectratio (opt.) 555	inkscapepath $\dots, 4, \frac{524}{322}$
keepaspectratio (opt.)	$\begin{array}{llllllllllllllllllllllllllllllllllll$
	$ \begin{array}{cccccccccccccccccccccccccccccccccccc$
keepaspectratio (opt.) $\dots \dots \dots \underline{555}$	$ \begin{array}{llllllllllllllllllllllllllllllllllll$
$ \begin{array}{cccccccccccccccccccccccccccccccccccc$	$\begin{array}{ccc} \text{inkscapepath} & 4, \underline{524} \\ \text{inkscapeversion} & 5, \underline{397} \\ \text{keepaspectratio} & \underline{555} \\ \text{lastpage} & 6, \underline{635} \\ \text{latex} & \underline{458} \\ \end{array}$
	$\begin{array}{cccc} {\rm inkscapepath} & & 4, \underline{524} \\ {\rm inkscapeversion} & & 5, \underline{397} \\ {\rm keepaspectratio} & & \underline{555} \\ {\rm lastpage} & & 6, \underline{635} \\ {\rm latex} & & \underline{458} \\ {\rm latexexe} & & 10, \underline{1441} \\ \end{array}$
$\begin{tabular}{c cccc} $\tt keepaspectratio\ (opt.) & & & & & & \\ \hline & & & & & & \\ \hline & & & &$	$\begin{array}{cccc} {\rm inkscapepath} & & 4, \underline{524} \\ {\rm inkscapeversion} & & 5, \underline{397} \\ {\rm keepaspectratio} & & \underline{555} \\ {\rm lastpage} & & 6, \underline{635} \\ {\rm latex} & & \underline{458} \\ {\rm latexexe} & & 10, \underline{1441} \\ {\rm latexext} & & 10, \underline{1441} \\ \end{array}$
	$\begin{array}{cccc} {\rm inkscapepath} & & 4, \underline{524} \\ {\rm inkscapeversion} & & 5, \underline{397} \\ {\rm keepaspectratio} & & \underline{555} \\ {\rm lastpage} & & 6, \underline{635} \\ {\rm latex} & & \underline{458} \\ {\rm latexexe} & & 10, \underline{1441} \\ \end{array}$
$\begin{tabular}{c cccc} $\tt keepaspectratio\ (opt.) & & & & & & \\ \hline & & & & & & \\ \hline & & & &$	$\begin{array}{cccccccccccccccccccccccccccccccccccc$
$\begin{tabular}{c cccc} $\tt keepaspectratio\ (opt.) & & & & & & \\ \hline & & & & & & \\ \hline & & & &$	$\begin{array}{cccccccccccccccccccccccccccccccccccc$
L         lastpage (opt.)       6, 635         latex (opt.)       458         latexexe (opt.)       10, 1441         latexext (opt.)       10, 1441         latexopt (opt.)       10, 1441         magickexe (opt.)       11, 1735	$\begin{array}{cccccccccccccccccccccccccccccccccccc$
L         lastpage (opt.)       6, 635         latex (opt.)       458         latexexe (opt.)       10, 1441         latexext (opt.)       10, 1441         latexopt (opt.)       10, 1441         magickexe (opt.)       11, 1735         magickoperator (opt.)       11, 1735	$\begin{array}{cccccccccccccccccccccccccccccccccccc$
L         lastpage (opt.)       6, 635         latex (opt.)       458         latexexe (opt.)       10, 1441         latexext (opt.)       10, 1441         latexopt (opt.)       10, 1441         magickexe (opt.)       11, 1735	$\begin{array}{cccccccccccccccccccccccccccccccccccc$
L         lastpage (opt.)       6, 635         latex (opt.)       458         latexexe (opt.)       10, 1441         latexext (opt.)       10, 1441         latexopt (opt.)       10, 1441         magickexe (opt.)       11, 1735         magickoperator (opt.)       11, 1735	inkscapepath       4,524         inkscapeversion       5,397         keepaspectratio       555         lastpage       6,635         latex       458         latexee       10,1441         latexext       10,1441         latexopt       10,1441         magickexe       11,1735         magickoperator       11,1735         magicksetting       11,1735         name       1836         notransparent       3,286
L         lastpage (opt.)       6, 635         latex (opt.)       458         latexexe (opt.)       10, 1441         latexext (opt.)       10, 1441         latexopt (opt.)       10, 1441         magickexe (opt.)       11, 1735         magickoperator (opt.)       11, 1735         magicksetting (opt.)       11, 1735	inkscapepath       4,524         inkscapeversion       5,397         keepaspectratio       555         lastpage       6,635         latex       458         latexee       10,1441         latexext       10,1441         latexopt       10,1441         magickexe       11,1735         magickoperator       11,1735         name       1836         notransparent       3,286         noxcolor       3,286
$ \begin{array}{c ccccccccccccccccccccccccccccccccccc$	$\begin{array}{cccccccccccccccccccccccccccccccccccc$
L         lastpage (opt.)       6, 635         latex (opt.)       458         latexexe (opt.)       10, 1441         latexext (opt.)       10, 1441         latexopt (opt.)       10, 1441         nagickexe (opt.)       11, 1735         magickoperator (opt.)       11, 1735         magicksetting (opt.)       11, 1735         name (opt.)       1836	inkscapepath       4,524         inkscapeversion       5,397         keepaspectratio       555         lastpage       6,635         latex       458         latexexe       10,1441         latexext       10,1441         latexopt       10,1441         magickexe       11,1735         magickoperator       11,1735         name       1836         notransparent       3,286         off       8,395,1352         on       8,395,1352         path       1802         pdf       1356
L         lastpage (opt.)       6, 635         latex (opt.)       458         latexexe (opt.)       10, 1441         latexept (opt.)       10, 1441         latexopt (opt.)       10, 1441         magickexe (opt.)       11, 1735         magickoperator (opt.)       11, 1735         magicksetting (opt.)       11, 1735         nome (opt.)       1836         notransparent (opt.)       3, 286         noxcolor (opt.)       3, 286	inkscapepath       4,524         inkscapeversion       5,397         keepaspectratio       555         lastpage       6,635         latex       458         latexexe       10,1441         latexext       10,1441         latexopt       10,1441         magickexe       11,1735         magickoperator       11,1735         name       1836         notransparent       3,286         off       8,395,1352         on       8,395,1352         path       1802         pdf       1356         pdflatex       1441
L         lastpage (opt.)       6, 635         latex (opt.)       458         latexexe (opt.)       10, 1441         latexext (opt.)       10, 1441         latexopt (opt.)       10, 1441         latexopt (opt.)       11, 1735         magickexe (opt.)       11, 1735         magicksetting (opt.)       11, 1735         name (opt.)       11, 1735         notransparent (opt.)       3, 286         noxcolor (opt.)       3, 286         noxcolor (opt.)       3, 286	inkscapepath       4,524         inkscapeversion       5,397         keepaspectratio       555         lastpage       6,635         latex       458         latexexe       10,1441         latexext       10,1441         latexopt       10,1441         magickexe       11,1735         magickoperator       11,1735         name       1836         notransparent       3,286         off       8,395,1352         on       8,395,1352         path       1802         pdf       1356         pdflatex       1441         pdftoepsopt       10,1479
L         lastpage (opt.)       6, 635         latex (opt.)       458         latexexe (opt.)       10, 1441         latexopt (opt.)       10, 1441         latexopt (opt.)       10, 1441         magickexe (opt.)       11, 1735         magickoperator (opt.)       11, 1735         magicksetting (opt.)       11, 1735         notransparent (opt.)       3, 286         noxcolor (opt.)       3, 286	inkscapepath       4,524         inkscapeversion       5,397         keepaspectratio       555         lastpage       6,635         latex       458         latexexe       10,1441         latexext       10,1441         latexopt       10,1441         magickexe       11,1735         magickoperator       11,1735         name       1836         notransparent       3,286         off       8,395,1352         on       8,395,1352         path       1802         pdf       1356         pdflatex       1441

pdftopsopt $10, \underline{1479}$	inkscapearea-\includesvg $6, \underline{712}$
png	inkscapedpi-\includesvg $6, \frac{712}{0.00}$
postex	inkscapeformat -\includeinkscape $\frac{7}{802}$
preamble	inkscapeformat – \includesvg $6, \frac{712}{600}$
pretex	inkscapelatex \includeinkscape 7, 802
pstoepsopt	inkscapelatex-\includesvg $6$ , $712$ inkscapeopt-\includesvg $6$ , $712$
scale $6, \underline{555}$	lastpage \includeinkscape \dots 7, 802
sygextension	lastpage \includesinkscape \tag{7, \frac{502}{502}}
svgpath	latexexe-\includeinkscape 2051
tex	latexexe-\includesvg $\frac{2051}{1}$
usetransparent	$\begin{array}{cccccccccccccccccccccccccccccccccccc$
usexcolor	$latexext-\cludesvg \dots 2051$
width $6, 555$	$\texttt{latexopt-} \\ \texttt{\lambda} \\ \texttt{includeinkscape} \\ \ldots \\ \underline{2051}$
_	$\texttt{latexopt-} \\ \texttt{\lambda} \\ \texttt{includesvg} \\ \ldots \\ \underline{2051} \\ $
P	magickoperator-\includeinkscape $2051$
parameters:	magickoperator - \includesvg $\frac{2051}{2057}$
angle-\includeinkscape 7, <u>802</u>	magicksetting-\includeinkscape 2051
angle - \includesvg	magicksetting-\includesvg 2051
apptex-\includeinkscape $7, 802$ apptex-\includesvg $6, 712$	origin-\includeinkscape 7, 802
apptex-\includesvg $6$ , $712$ clean-\includeinkscape $2051$	origin-\includesvg 7, 744
clean - \includesvg	pdftoepsopt-\includeinkscape 2051 pdftoepsopt-\includesvg 2051
convert - \includeinkscape 2051	pdftopsopt \includesinkscape 2001
convert - \includesvg 2051	pdftopsopt-\includesvg 2051
convertdpi-\includeinkscape 2051	pretex-\includeinkscape $7, 802$
convertdpi-\includesvg 2051	pretex-\includesvg $\dots \dots 6, \overline{712}$
$\verb convertformat-\include  inkscape & \underline{2051} \\$	pstoepsopt - \includeinkscape $\frac{2051}{}$
$\texttt{convertformat-} \\ \texttt{includesvg} \\ \\  \\ \underline{2051}$	$\verb pstoepsopt-\includesvg  \underline{2051}$
$\texttt{distort-} \\ \texttt{includeinkscape}  \dots  \textcolor{red}{7},  \underline{802}$	${\tt pstopdfopt-\line ludeinkscape} \ \dots \ \ \underline{2051}$
$distort-\coloner-line ludesvg \dots 6, \underline{712}$	${\tt pstopdfopt-\cludesvg} \ \dots \ \underline{2051}$
draft-\includeinkscape 7, <u>802</u>	$scale-\includeinkscape \dots 7, 802$
$\frac{\text{draft}-\text{\text{includesvg}} \dots \dots \dots \dots 6, \underline{712}}{2000}$	$scale-\includesvg$ $6, \frac{712}{712}$
dvipsopt \includeinkscape 2051	sygextension—\includesvg $6, \frac{712}{6000}$
$\begin{array}{cccccccccccccccccccccccccccccccccccc$	width \includeinkscape 7, 802
exclude \includesvg 2051	width-\includesvg         6, 712           path (opt.)         1802
extract - \includeinkscape 2051	pdf (opt.)
extract -\includesvg 2051	pdflatex (opt.)
extractangle - \includeinkscape $10, \frac{2051}{2051}$	pdftoepsopt (opt.)
extractangle-\includesvg $10, \overline{2051}$	pdftops (opt.)
extractapptex-\includeinkscape $2051$	pdftopsopt (opt.)
$\verb extractapptex-\includesvg  \underline{2051}$	png (opt.)
extractdistort - \includeinkscape $\frac{2051}{}$	${\tt postex} \; ({\tt opt.}) \; \dots \qquad \underline{612}$
extractdistort - \includesvg 2051	$\texttt{preamble} \; (\texttt{opt.})  \dots \qquad \underline{1397}$
extractformat $-$ \includeinkscape $\frac{2051}{2051}$	pretex (opt.)
extractformat - \includesvg 2051	pstoepsopt (opt.)
extractheight-\includeinkscape 2051 extractheight-\includesvg 2051	pstopdfopt (opt.)
extractheight-\includesvg 2051 extractpreamble-\includeinkscape 2051	S
extractpreamble \includesvg \frac{2051}{2051}	scale (opt.)
extractpretex-\includeinkscape 2051	\setsvg
extractpretex-\includesvg 2051	\svg@append@input@path 1166
extractruns - \includeinkscape $\frac{2051}{}$	\svg@box 901
$\verb extractruns-\includesvg                                 $	$\verb \svg@deactivate@dq \underline{46} $
extractscale-\includeinkscape . $\underline{2051}$	$\verb \svg@deprecated@key  \underline{278}$
extractscale-\includesvg $\underline{2051}$	\svg@deprecated@param $\dots \dots \underline{673}$
extractwidth-\includeinkscape . 2051	\svg@dummy@key <u>1297</u>
extractwidth-\includesvg 2051	\svg@extension@parse 143
gsdevice \includeinkscape 2051	\svg@extension@@parse 143
gsdevice -\includesvg	\svg@extract
gsopt-\includeinkscape         2051           gsopt-\includesvg         2051	$ \begin{array}{llllllllllllllllllllllllllllllllllll$
height-\includeinkscape 7, 802	\svg@file@missing
height \langle includes vg $\dots$ $6$ , $\frac{602}{712}$	\svg@file@name
inkscape-\includesvg $6, \frac{712}{712}$	\svg@file@path 1207
• • • • • • • • • • • • • • • • • • • •	- •

svgsetup $4$ , $8$ , $695$ svgx@clean $2008$	<b>W</b> width (opt.)
vgpath (opt.)	
svgpath 4, <u>697</u>	usexcolor (opt.)
syghidepreamblestart $9, \frac{2073}{2073}$	usetransparent (opt.) 3, 2
$\begin{array}{cccccccccccccccccccccccccccccccccccc$	U
vgextension (opt.)	tex (opt.) <u>4</u>
$ ext{svg@tempb} \dots  ext{17} $ $ ext{svg@wrn@scale} \dots  ext{963}$	т
$\frac{17}{17}$	\svgxsetpapersize 25
svg@shell@@rm	\svgxsetbox
svg@shell@rm	\svgxoutputbox
svg@shell@@mv	\svgxQuseformatkey $\dots $
svg@shell@mv <u>211</u>	\svgx@stream@out 21
svg@shell@@mkdir $\underline{211}$	\svgx@stream@in 21
svg@shell@mkdir	$\verb \svgx@setformatkey  \underline{170}$
$svg@set@input@path \dots 1166$	\svgx@setbox
svg@sanitize@dq	svgx@runs (counter) 14
svg@remove@leadingchar 90	\svgx@read@preamble@till 24
svg@quotes@@remove	\svgx@read@preamble@skip 24
svg@quotes@remove	\svgx@read@preamble@from 24
svg@quotes@Check	\svgx@read@line 21
$ ext{svg@picture@saved} \dots  ext{1247} $ $ ext{svg@quotes@check} \dots  ext{57}$	\svgx@pstopdf@exe 14' \svgx@pstopdf@opt 14'
$\frac{1256}{1247}$	\svgx@pstoeps@opt <u>14'</u>
svg@pictur@patched	\svgxQpstoepsQexe
svg@patches <u>1247</u>	\svgx@preamble <u>13</u>
svg@param@width	\svgx@pdftops@opt 14'
svg@param@scale	\svgx@pdftops@exe <u>14</u> '
$svg@param@pretex \dots \underline{612}$	$\verb \svgx@pdftoeps@opt  \underline{14} $
vg@param@lastpage (counter) 635	$\verb \svgx@pdftoeps@exe  \underline{14'}$
$svg@param@apptex \dots 612$	\svgx@param@width <u>18</u>
svg@out@path $524$	\svgx@param@scale <u>18</u>
$svg@out@name \dots 549$	\svgx@param@pretex 19
$svg@out@base \dots 549$	\svgx@param@distort 18'
svg@normalize@@path 238	\svgx@param@apptex 19
svg@normalize@path 238	\svgx@out@sec 18
svg@local@param@use	\svgx@out@path 186
svg@local@param@set	\svgx@out@name
svg@local@param@def	svgx@out@count (counter)
svg@@input 901	\svgx@move
svg@input@path	\svgx@magick@set 173
svg@@ink@ver@detect1000svg@input901	\svgx@magick@exe
$ ext{svg@ink@ver@settings} \dots  ext{1060} \  ext{svg@@ink@ver@detect} \dots  ext{1060}$	\svgx@magick@cmd 173 \svgx@magick@exe 173
$ ext{svg@ink@ver@explore} \dots  ext{1144} $ $ ext{svg@ink@ver@settings} \dots  ext{1060}$	\svgx@latex@opt
$     \text{syg@ink@ver@detect} \dots                                   $	\svgx@latex@ext <u>14</u>
syg@ink@ver	\svgx@latex@exe
svg@ink@run	\svgx@@ifkeyandval
svg@ink@opt	\svgx@ifkeyandval
$svg@ink@mode \dots 315$	\svgx@ifinlist 20
svg@ink@latex	\svgx@gs@opt <u>17</u>
$svg@ink@format$ $\underline{438}$	\svgx@gs@exe <u>17</u> 0
${ t svg@ink@exe}$ ${ t \underline{397}}$	\svgx@gs@device <u>17</u> 0
svg@ink@dpi $\underline{476}$	\svgx@gs@cmd <u>17</u> 0
svg@ink@cmd $493$	\svgx@get@out@sec 24
	\svgx@format 133
$svg@includegraphics@saved \dots 1247$	\svgx@endpreamble 13
$ \frac{1}{1279} $ svg@includegraphics@patched $\frac{1}{1279}$	\svgx@dvips@opt 14
svg@includegraphics@file 1279	\svgx@dvips@exe
svg@ifwindowsdetected 20	\svgx@documentclass 244
svg@ifvalueisrelax $30$	\svgx@cnv@informat 24
svg@iffilenewer	\svgx@cnv@get@informat 24
2.00000bms2 <u>2201</u>	\svgx@cnv@get@dpi 163
svg@get@lastpage         360           svg@get@path         1207	
svg@filename@parse $\frac{106}{980}$ svg@get@lastpage $\frac{980}{1207}$ svg@get@path $\frac{1207}{1207}$	\svgx@cnv@dpi <u>16</u> ; \svgx@cnv@format <u>16</u> 0

# **Change History**

v1.0	inkscapelatex (opt.): new 458
General	inkscapename (opt.): new 549
initial version by Philip Ilten 2	inkscapeopt (opt.): new 397
• •	inkscapepath (opt.): new <u>524</u>
v2.00	lastpage (opt.): new
General	latexexe (opt.): new 1441
new maintainer: Falk Hanisch 2	latexext (opt.): new 1441
package <b>subfig</b> not required anymore 2	latexopt (opt.): new 1441
re-implementation from scratch 2	magickexe (opt.): new 1735
support of subfigures stopped due to the	magickoperator (opt.): new 1735
huge number of packages which deal	magicksetting (opt.): new 1735
with this topic and the large variety	name (opt.):
of implementing this functionality;	deprecated
naming exported graphics after their	support of <b>subfig</b> removed 1836
consecutive numbering can't be	notransparent (opt.): new 286
ensured for all variants of subfigures,	noxcolor (opt.): new
so it's neglected	off (opt.): new
Implementation	on (opt.): new
-	path (opt.): deprecated
(1)	<del>-</del>
convert (opt.): changed/extended <u>1528</u>	· · · · / · · · · · · · · · · · · · · ·
convertdpi (opt.): new <u>1631</u>	pdflatex (opt.): deprecated 1441
convertformat (opt.): new 1604	pdftoepsopt (opt.): new 1479
$draft (opt.): new \dots \underline{649}$	pdftops (opt.): deprecated 1479
dvipsopt (opt.): new $\dots \underline{1479}$	pdftopsopt (opt.): new
end (opt.): deprecated <u>1397</u>	png (opt.): deprecated <u>1604</u>
eps (opt.): deprecated <u>1356</u>	postex (opt.): deprecated
extract (opt.): new $\dots $ $\underline{1324}$	preamble (opt.): deprecated <u>1397</u>
extractapptex (opt.): new $\dots 1972$	pstoepsopt (opt.): new 1479
extractformat (opt.): new $\underline{1356}$	pstopdfopt (opt.): new $\dots $ $\underline{1479}$
extractheight (opt.): new 1876	scale (opt.): new <u>555</u>
extractname (opt.): new <u>1836</u>	\setsvg: deprecated <u>695</u>
extractpath (opt.): new <u>1802</u>	\svghidepreambleend: new $2073$
extractpreamble (opt.): new 1397	\svghidepreamblestart: new $2073$
extractpreambleend (opt.): new 1397	\svgpath: new
extractpretex (opt.): new 1972	$\mathtt{svgpath}$ (opt.): deprecated $\underline{505}$
extractruns (opt.): new 1424	\svgsetup: new <u>695</u>
extractscale (opt.): new 1876	usetransparent (opt.): new 286
extractwidth (opt.): new 1876	usexcolor (opt.): new <u>286</u>
gsdevice (opt.): new	
gsexe (opt.): new	v2.00a
gsopt (opt.): new	Implementation
height (opt.): new	\svgxsetpapersize: Bug fix for
\includeinkscape: new	checking stock- and mediasizes $2554$
\includesvg:	
-	2 006
changes, especially to optional parameters	v2.00b
parameters	Implementation
244	Implementation latex (opt.): new, alternative key for
angle (param.): new	Implementation latex (opt.): new, alternative key for
<b>draft</b> (param.): new	Implementation latex (opt.): new, alternative key for
draft (param.): new	Implementation latex (opt.): new, alternative key for inkscapelatex
draft (param.): new       712         height (param.): new       712         inkscape (param.): new       712	Implementation  latex (opt.): new, alternative key for  inkscapelatex
draft (param.): new       712         height (param.): new       712         inkscape (param.): new       712         inkscapearea (param.): new       712	Implementation latex (opt.): new, alternative key for inkscapelatex
draft (param.): new       712         height (param.): new       712         inkscape (param.): new       712         inkscapearea (param.): new       712         inkscapedpi (param.): new       712	Implementation latex (opt.): new, alternative key for inkscapelatex
draft (param.): new       712         height (param.): new       712         inkscape (param.): new       712         inkscapearea (param.): new       712         inkscapedpi (param.): new       712         inkscapeformat (param.): new       712	Implementation latex (opt.): new, alternative key for inkscapelatex
draft (param.): new       712         height (param.): new       712         inkscape (param.): new       712         inkscapearea (param.): new       712         inkscapedpi (param.): new       712         inkscapeformat (param.): new       712         inkscapelatex (param.): new       712	Implementation latex (opt.): new, alternative key for inkscapelatex
draft (param.): new       712         height (param.): new       712         inkscape (param.): new       712         inkscapearea (param.): new       712         inkscapedpi (param.): new       712         inkscapeformat (param.): new       712         inkscapelatex (param.): new       712         inkscapeopt (param.): new       712         inkscapeopt (param.): new       712	Implementation latex (opt.): new, alternative key for inkscapelatex
draft (param.): new       712         height (param.): new       712         inkscape (param.): new       712         inkscapedpi (param.): new       712         inkscapeformat (param.): new       712         inkscapeformat (param.): new       712         inkscapelatex (param.): new       712         inkscapeopt (param.): new       712         lastpage (param.): new       741	Implementation latex (opt.): new, alternative key for inkscapelatex
draft (param.): new       712         height (param.): new       712         inkscape (param.): new       712         inkscapedpi (param.): new       712         inkscapeformat (param.): new       712         inkscapelatex (param.): new       712         inkscapeopt (param.): new       712         inkscapeopt (param.): new       712         inkscapeopt (param.): new       741         origin (param.): new       744	Implementation latex (opt.): new, alternative key for inkscapelatex
draft (param.): new       712         height (param.): new       712         inkscape (param.): new       712         inkscapedpi (param.): new       712         inkscapeformat (param.): new       712         inkscapelatex (param.): new       712         inkscapeopt (param.): new       712         inkscapeopt (param.): new       712         inkscapeopt (param.): new       741         origin (param.): new       744         scale (param.): new       712	Implementation latex (opt.): new, alternative key for inkscapelatex
draft (param.): new       712         height (param.): new       712         inkscape (param.): new       712         inkscapedpi (param.): new       712         inkscapeformat (param.): new       712         inkscapelatex (param.): new       712         inkscapeopt (param.): new       712         inkscapeopt (param.): new       712         inkscapeopt (param.): new       741         origin (param.): new       744	Implementation  latex (opt.): new, alternative key for  inkscapelatex
draft (param.): new       712         height (param.): new       712         inkscape (param.): new       712         inkscapedpi (param.): new       712         inkscapeformat (param.): new       712         inkscapelatex (param.): new       712         inkscapeopt (param.): new       712         inkscapeopt (param.): new       712         inkscape (param.): new       741         origin (param.): new       744         scale (param.): new       712	Implementation latex (opt.): new, alternative key for inkscapelatex
draft (param.): new       712         height (param.): new       712         inkscape (param.): new       712         inkscapearea (param.): new       712         inkscapedpi (param.): new       712         inkscapeformat (param.): new       712         inkscapelatex (param.): new       712         inkscapeopt (param.): new       712         lastpage (param.): new       741         origin (param.): new       744         scale (param.): new       712         inkscape (opt.): changed/extended       315	Implementation  latex (opt.): new, alternative key for inkscapelatex
draft (param.): new       712         height (param.): new       712         inkscape (param.): new       712         inkscapearea (param.): new       712         inkscapedpi (param.): new       712         inkscapeformat (param.): new       712         inkscapelatex (param.): new       712         inkscapeopt (param.): new       712         lastpage (param.): new       741         origin (param.): new       742         scale (param.): new       712         inkscape (opt.): changed/extended       315         inkscapearea (opt.): new       462	Implementation latex (opt.): new, alternative key for inkscapelatex

$\svg@append@input@path: new 1166$	\svg@sanitize@dq: $new \dots 51$
\svg@get@path:	$\verb \svg@set@input@path : usage of$
using \svg@set@input@path $\underline{1207}$	\svg@deactivate@dq $\dots $ $\underline{1166}$
using $\tau$ $\frac{1207}{}$	<pre>svgextension (opt.):</pre>
$\svg@set@input@path: new 1166$	usage of \svgQquotes@remove $517$
svgextension (opt.): new due to user	usage of \svg@remove@leadingchar $517$
request	\svgpath: parse argument for enclosing
v2.02	braces and provide those if
V2.02 General	necessary
fixed errors with active double quotes	\svgx@setbox: new <u>2539</u>
from <b>babel</b> in path arguments 2	\svgxsetbox: late execution of
multiple dots within file names possible 2	\svgxsetpapersize $2539$
	v2.02a
option distort (or keepaspectratio) added for distortion of included	V2.02a General
graphics	
option extractdistort added for	fix bug for package <b>polyglossia</b> which
distortion of extracted graphics 9	fakes <b>babel</b> poorly
package <b>trimspaces</b> required	Implementation
parameter extractangle for	\svg@deactivate@dq: bug fix for
\includesvg and \includeinkscape	polyglossia
implemented in order to rotate	v2.02b
graphics during extractions 10	General V2.02B
Implementation	
distort (opt.): new	fix bug for package tikzscale which changes \includegraphics globally 2
extractdistort (opt.): new 1876	changes \includegraphics globally 2 Implementation
extractname (opt.): usage of	1
\svg@quotes@remove 1836	\svg@patches: fix bug for package tikzscale: store original definitions of
extractpath (opt.): usage of	\picture and \includegraphics
\svg@sanitize@dq <u>1802</u>	right after loading package svg . 1247
\if@svgx@standalone: new 2539	right after loading package svg . 1241
\includeinkscape:	v2.02c
usage of \svg@extension@parse 761	General
extractangle (param.): new 2051	fix bugs with kernel $(2019/10/01)$
extractdistort (param.): new 2051	regarding file name parsing 2
	regarding me name parsing
\includesvg:	
\includesvg: switched to \svg@filename@parse 709	v2.02d
switched to \svg@filename@parse 709	v2.02d General
switched to \svg@filename@parse 709 angle (param.): validation of	General
switched to \svg@filename@parse 709 angle (param.): validation of argument	General conditional invocation of <i>Inkscape</i>
switched to \svg@filename@parse 709  angle (param.): validation of  argument	General conditional invocation of <i>Inkscape</i> export based on file modification
switched to \svg@filename@parse         709           angle (param.): validation of         744           argument         712           extractangle (param.): new         2051	General conditional invocation of <i>Inkscape</i> export based on file modification date implemented for XeTeX 2
switched to \svg@filename@parse 709 angle (param.): validation of argument	General conditional invocation of <i>Inkscape</i> export based on file modification date implemented for XeT <sub>E</sub> X 2 fix bugs with kernel (2019/10/01)
switched to \svg@filename@parse 709 angle (param.): validation of argument	General conditional invocation of <i>Inkscape</i> export based on file modification date implemented for XeT <sub>E</sub> X 2 fix bugs with kernel (2019/10/01) regarding file name parsing, see
switched to \svg@filename@parse 709 angle (param.): validation of argument	General conditional invocation of <i>Inkscape</i> export based on file modification date implemented for XeT <sub>E</sub> X 2 fix bugs with kernel (2019/10/01)
switched to \svg@filename@parse 709 angle (param.): validation of argument	General conditional invocation of <i>Inkscape</i> export based on file modification date implemented for XeTeX
switched to \svg@filename@parse 709 angle (param.): validation of argument	General conditional invocation of <i>Inkscape</i> export based on file modification date implemented for XeTeX
switched to \svg@filename@parse 709 angle (param.): validation of argument	General conditional invocation of <i>Inkscape</i> export based on file modification date implemented for XeTeX
switched to \svg@filename@parse 709 angle (param.): validation of argument	General conditional invocation of Inkscape export based on file modification date implemented for XeTEX
switched to \svg@filename@parse 709 angle (param.): validation of argument	General conditional invocation of Inkscape export based on file modification date implemented for XeTEX
switched to \svg@filename@parse 709 angle (param.): validation of argument	General conditional invocation of Inkscape export based on file modification date implemented for XeTEX
switched to \svg@filename@parse       709         angle (param.): validation of       744         distort (param.): new       712         extractangle (param.): new       2051         extractdistort (param.): new       2051         inkscape (opt.): usage of       315         inkscapepath (opt.): usage of       524         keepaspectratio (opt.): new       555         \svg@append@input@path: avoid       duplicates in \input@path       1166         \svg@deactivate@dq: new       46	General conditional invocation of Inkscape export based on file modification date implemented for XeTEX
switched to \svg@filename@parse         709           angle (param.): validation of         744           distort (param.): new         712           extractangle (param.): new         2051           extractdistort (param.): new         2051           inkscape (opt.): usage of         315           inkscapepath (opt.): usage of         524           keepaspectratio (opt.): new         555           \svg@append@input@path: avoid         duplicates in \input@path         1166           \svg@deactivate@dq: new         46           \svg@extension@parse: new         143	General conditional invocation of Inkscape export based on file modification date implemented for XeTEX
switched to \svg@filename@parse         709           angle (param.): validation of         744           distort (param.): new         712           extractangle (param.): new         2051           extractdistort (param.): new         2051           inkscape (opt.): usage of         315           inkscapepath (opt.): usage of         524           keepaspectratio (opt.): new         555           \svg@append@input@path: avoid         duplicates in \input@path         1166           \svg@deactivate@dq: new         46           \svg@extension@parse: new         143           \svg@extension@parse: new         143	General  conditional invocation of Inkscape export based on file modification date implemented for XeTEX
switched to \svg@filename@parse         709           angle (param.): validation of         744           argument         742           distort (param.): new         2051           extractangle (param.): new         2051           extractdistort (param.): new         2051           inkscape (opt.): usage of         315           inkscapepath (opt.): usage of         \$24           keepaspectratio (opt.): new         555           \svg@append@input@path: avoid         duplicates in \input@path         1166           \svg@deactivate@dq: new         46           \svg@extension@parse: new         143           \svg@extension@oparse: new         143           \svg@file@missing: notify svg file	General conditional invocation of Inkscape export based on file modification date implemented for XeTEX
switched to \svg@filename@parse 709 angle (param.): validation of argument	General  conditional invocation of Inkscape export based on file modification date implemented for XeTEX
switched to \svg@filename@parse 709 angle (param.): validation of argument	General conditional invocation of Inkscape export based on file modification date implemented for XeTEX
switched to \svg@filename@parse 709 angle (param.): validation of argument	General conditional invocation of Inkscape export based on file modification date implemented for XeTeX
switched to \svg@filename@parse 709 angle (param.): validation of argument	General conditional invocation of Inkscape export based on file modification date implemented for XeTeX
switched to \svg@filename@parse 709 angle (param.): validation of argument	General conditional invocation of Inkscape export based on file modification date implemented for XeTeX
switched to \svg@filename@parse 709 angle (param.): validation of argument	General conditional invocation of Inkscape export based on file modification date implemented for XeTeX
switched to \svg@filename@parse 709 angle (param.): validation of argument	General conditional invocation of Inkscape export based on file modification date implemented for XeTeX
switched to \svg@filename@parse 709 angle (param.): validation of argument	General conditional invocation of Inkscape export based on file modification date implemented for XeTeX
switched to \svg@filename@parse 709 angle (param.): validation of argument	General conditional invocation of Inkscape export based on file modification date implemented for XeTeX
switched to \svg@filename@parse 709 angle (param.): validation of argument	General conditional invocation of Inkscape export based on file modification date implemented for XeTeX

\svg@ink@ver@settings: new $\underline{1060}$	v2.02i
\svg@@ink@ver@detect: new $\underline{1060}$	General
v2.02g General	update for package <b>scrlfile</b> v3.32 and kernel $(2020/10/01)$
fix for multiple dots in file names $(#27)$ 2	
Implementation \svg@@input: parsing of file extension discarded; meanwhile taken over by	${\tt v2.02j}$ General
the kernel	bug fix for automatic version detection of $Inkscape$ with MiKTEX 2
General	Implementation
fix for package transparent $(\#28)$ 2	\svg@ink@ver@explore: new $\underline{1144}$