The packages svg and svg-extract

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

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

v2.02e (2020/01/13)

The **svg** package is intended for the automated integration of SVG graphics into IATEX documents. Therefor the capabilities provided by *Inkscape*—or more precisely its command line tool—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 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.

Contents

I.	User documentation	2
1.	Introduction	2
2.	Usage of package svg	3
2.1.	General settings	3
2.2.	Options for the invocation of <i>Inkscape</i>	4
2.3.	Options for the graphic inclusion	5
2.4.	Including SVG files	6
2.5.	Including already exported SVG files	6
3.	Usage of package svg-extract	7
3.1.	General settings	7
3.2.	Extract independent grahic files	8
3.3.	Convert extracted grahic files	9
3.3.1.	Settings for the invocation of ImageMagick	11
3.3.2.	Settings for the invocation of Ghostscript	11
4.	Example	11
5 .	Troubleshooting and reporting issues	13
6.	Include SVG files created with ROOT	13
II.	Implementation	16
Α.	Initialization	16
A.1.	Packages	16

File: svg.dtx Date: 2020/01/13 v2.02e

A.2.	Helper macros	16
В.	Including SVG files with package svg	16
B.1.	Options	16
B.1.1.	The invocation of <i>Inkscape</i>	17
B.1.2.	Setting input folder and file	20
B.1.3.	Setting output folder	21
B.1.4.	Options for the inclusion of graphics	21
B.2.	Handling path information	23
B.3.	Optional Parameters for user commands	31
B.4.	User commands	31
B.5.	Auxiliary macros	34
B.6.	Patches	39
C.	Extracting independent graphic files with svg-extract	41
C.1.	Options	41
C.1.1.	Controlling the extract process	41
	Invoking external application for graphic conversion	45
C.1.3.	9	48
C.1.4.	Options for the extraction of graphics	49
C.1.5.	Miscellaneous options	52
C.2.	User commands	52
C.3.	Auxiliary macros	53
C.4.	Commands for the separate auxiliary LATEX-file	66
D.	Processing Options	67
E.	Macros for file access	68
Index		69
Change 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 L^AT_EX, and an image component that can be imported as a PDF file. For further information see the documentation of **svg-inkscape** on CTAN¹. The procedure described therein is taken up and consistently expanded. Thus, it is now possible to include a SVG file into a L^AT_EX document where the text within the SVG graphic will be rendered natively by L^AT_EX.

Both packages svg and svg-extract rely heavily upon executing commands from the shell using the \ShellEscape command—or respectively the old known \write18—for executing a variety of commands directly to the system. So it is necessary to include the flag --shell-escape when compiling documents using 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.

File I: svg.dtx Date: 2020/01/13 v2.02e

¹http://www.ctan.org/pkg/svg-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 functionallity. 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 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 assinged to usexcolor and usetransparent, while noxcolor and notransparent don't accept any value.

```
\usepackage[\langle options \rangle] \{ svg \}
```

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. Settings with $\space{logorithms}\$ are done in the current scope which means globally or within the current group.

```
\symbol{sysetup}{\langle options \rangle}
```

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

File I: svg.dtx Date: 2020/01/13 v2.02e

\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 terminate with / last. For example:

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

would cause the system to look first in the subdirectory <code>svg/</code> and afterwards in the absolute path <code>/usr/local/svg/</code>. Further, if no path was specified with <code>\svgpath</code> or the desired file wasn't found, all directories given with <code>\graphicspath</code> are searched too. Please keep in mind that the current working directory is browsed first in any case. It's recommended to avoid any spaces and/or quotes respectively <code>\dq</code> both in paths and file names, especially when DVI output is active.

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.

pdf/eps/ps/png

see inkscapeformat=pdf/eps/ps/png

latex/nolatex

see inkscapelatex=true/false

drawing/page

see inkscapearea=drawing/page

 $\langle integer \rangle$ dpi

see inkscapedpi=(integer)

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

The option inkscapepath specifies, where the resulting files of the *Inkscape* export should be located. The subfolder ./svg-inkscape/ within the current working directory is used by default (inkscapepath=basesubdir).

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.

inkscapename (opt.) The file names of the *Inkscape* export are derived from the name of the base SVG file and can be modified with inkscapename=\(filename\). It's possible to use counters for specifying the name of the exported file. Repeatedly specifying the same file name will overwrite previously created files.

For including a SVG file, *Inkscape* is used to separate the text and image from the SVG file ${\tt inkscapeexe}\ ({\rm opt.})$ itself. In order to execute the command line tool from shell, the path where the executable is located has to be known to the operating system. You can check this by typing inkscape -V into the shell. If this check fails and you don't want to change environment variable path on your OS, you can use option inkscapeexe to set the absolute path where the executable of Inkscape is located. The option is set to inkscapeexe=inkscape by default.

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.

If option inkscapelatex=true is set, the output is split into a seperate PDF/EPS/PS file ${\tt inkscapelatex}\ ({\rm opt.})$ (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.

This option controls which area of the SVG file should be exported, drawing is set by default. inkscapearea (opt.)

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.

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.

You can use this option to pass additional switches to the *Inkscape* command line tool. For further information see the documentation of **Inkscape**².

> 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

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.

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:

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

File I: svg.dtx Date: 2020/01/13 v2.02e

5

inkscapeformat (opt.)

inkscapedpi (opt.)

inkscapeopt (opt.)

svgextension (opt.)

width (opt.) height (opt.)

distort (opt.) scale (opt.)

pretex (opt.) apptex (opt.)

²https://inkscape.org/de/doc/inkscape-man.html

³to provide compatibility for package graphicx, it's possible to use keepaspectratio=true as alias for distort=false and the other way round

draft (opt.) This option can be used with booelan 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 bug⁴ concerning the L^AT_EX export has been reported for *Inkscape* 0.91. It may happen that within the exported L^AT_EX 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 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.)
inkscapeopt (param.)
svgextension (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 any spaces and/or quotes respectively \dq both in paths an file names. Espacially 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 IATEX 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 IATEX 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.

lastpage (param.)

apptex (param.)
draft (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

 $\label{linear} \$

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.

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 {\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.

File I: svg.dtx Date: 2020/01/13 v2.02e

⁴https://bugs.launchpad.net/ubuntu/+source/inkscape/+bug/1417470

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 $\text{I}^{\text{A}}\text{T}_{\text{E}}\text{X}\,2_{\mathcal{E}}$ distributions. In additon, the command line tools 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 $\langle program \rangle$ --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 of 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 L^AT_EX 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 $\includesvg[\langle parameters \rangle] {\langle filename \rangle}$ and $\includeinkscape[\langle parameters \rangle] {\langle filename \rangle}$ are extended. They can be used to control the process of graphic extraction and converting.

extractangle (param.)

With this 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:

```
\label{linear_eq} $$\lim_{\alpha \in \mathbb{R}^+} \operatorname{angle}(filename)$$
```

All option described below can be used togehter with \svgsetup and are then valid in the current scope. There also exist identically named parameters for the optional arguments of

```
\label{localization} $$ \left( \frac{parameters}{svg filename} \right) $$ \left( \frac{parameters}{svg filename} \right) $$
```

These parameters have an effect only once when the specific command is executed and remain unchanged afterwards. These parameters are: extract, extractpreamble, extractformat, extractruns, latexopt, extractwidth, extractheight, extractdistort, extractscale, extractangle, extractpretex, extractapptex, convert, convertformat, convertdpi, magicksetting, magickoperator, gsopt, gsdevice, clean, exclude.

File I: svg.dtx Date: 2020/01/13 v2.02e

3.2. Extract independent grahic 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.

extractpath (opt.)

The path where the extracted and converted files are located can be specified with option extractpath, whereas extractpath=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.

basesubdir/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.

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

It's also possible to change the name for extracted and converted files. The default setting is extractname=filenamenumbered.

filename/name

The name of the exported *Inkscape* file is used and the suffix -extract is attached. filenamenumbered/namenumbered/numberedfilename/numberedname

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

numbered/section/numberedsection/sectionnumbered

The file name is composed by the number of extracted files and the current outline numbering.

$\langle filename \rangle$

You can use any file name, the file extension is derived from option extractformat. 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.

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

The included SVG file can be extracted from the document into a 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,

$\verb|\includesvg[extractformat={pdf,eps,ps}]| \{ \langle svg \ filename \rangle \}|$

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.

 $\begin{tabular}{ll} \mbox{extractwidth } (\mbox{opt.}) \\ \mbox{extractheight } (\mbox{opt.}) \end{tabular}$

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:

extractdistort (opt.)
 extractscale (opt.)
 extractpretex (opt.)

$\verb|\cluster| includes vg[width=\textwidth, extractwidth=500pt]{|\cluster| filename|}|$

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

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. Only option extractdistort is initialized to false and does not inherit from distort by default.

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

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 IATEX file. These parts can be excluded if they are enclosed by \svghidepreamblestart and \svghidepreambleend.

For example, your current LATEX 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\{\langle documentclassname \rangle\}
\usepackage{svg-extract}
\svghidepreamblestart
\usepackage{showframe}
\svghidepreambleend
```

extractruns (opt.)

When extracting independent grahic files by compiling the generated auxiliary LATEX file, it's maybe necessary to do multiple LATEX runs on this file. The number of runs can be controlled with option extractruns. It's set to extractruns=2 by default.

latexexe (opt.) latexopt (opt.)

latexext (opt.)

For the extraction of an independent grahic file, the LATEX program is used which is set by the latexexe option. Depending on the LATEX processor 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 runs, which are performed during the extraction process by the latexopt option. If you are used to utilize a other file extension for LATEX files than .tex, option latexext can be used like latexext=ltx.

dvipsopt (opt.) pstoepsopt (opt.) pstopdfopt (opt.) pdftoepsopt (opt.) pdftopsopt (opt.)

Depending on the used LATEX processor, the file type of the extracted graphic differs. In order to create all formats, requested with option extractformat, several converting tools provided by most of the LATEX 2ε distributions are maybe invoked. These are dvips, ps2eps, ps2pdf and/or pdftops and can't be changed. It's only possible to specify additional switches for every single tool with dvipsopt, pstoepsopt, pstopdfopt, 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).

3.3. Convert extracted grahic files

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

convert (opt.)

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.

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 ${\it ImageMagick}$ is selected.

gs/ghostscript

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

gs64/ghostscript64

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

gs32/ghostscript32

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

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 $\langle option \rangle = \langle outputformat \rangle = \langle value \rangle$ and even $\langle option \rangle = \langle outputformat \rangle + = \langle value \rangle$ 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 a 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.

If you want to reverse any setting, you only have to use \relax as a value, either for a general option $(\langle option \rangle = \relax)$ or a specific one $(\langle option \rangle = \{\langle outputformat \rangle [+] = \relax\})$.

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

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

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

The conversion with *ImageMagick* via the magick or convert command-line tool 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 tool⁵.

3.3.2. Settings for the invocation of Ghostscript

gsexe (opt.)
gsdevice (opt.)
gsopt (opt.)

The conversion with *Ghostscript* is done with command-line tool gs on unix-like operating systems and gswin64c or gswin32c on Windows. The executable can be changed with option gsexe. 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 $Ghostscript^6$.

4. Example

As an minimal example take the following lines of code:

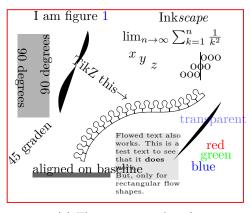
```
\documentclass{article}
\usepackage[T1]{fontenc}
\usepackage{svg}
\usepackage[off]{svg-extract}
\svgsetup{clean=true}
%\pdfsuppresswarningpagegroup=1
\usepackage{relsize}
\usepackage{subcaption}
\begin{document}
\begin{figure}
 \begin{minipage}{.5\linewidth}
   \includesvg[width=\linewidth]{svg-example}%
   \subcaption{This text is too large!}
 \end{minipage}%
 \begin{minipage}{.5\linewidth}
   \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}
```

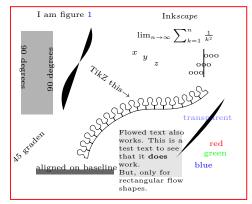
File I: svg.dtx Date: 2020/01/13 v2.02e

⁵http://www.imagemagick.org/script/command-line-processing.php

⁶https://ghostscript.com/doc/current/Use.htm

⁷The image used here is a slightly modified version of the image used in the initial documentation on how to include a SVG file in L^ATEX by Johan B. C. Engelen available as package svg-inkscape on CTAN.





(a) This text is too large!

(b) This text fits better.

Figure 1: An example figure with LATEX support

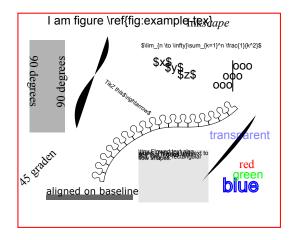


Figure 2: The same example figure without LATEX support

\end{document}

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 $\langle texmf \rangle / doc/latex/svg/examples/$. Second, you have to run the desired IATEX engine with --shell-escape option enabled.

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.

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

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 via 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 from shell 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 pdfLTEX 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_EX version 1.40.15 or later, you can get rid of these messages by using \pdfsuppresswarningpagegroup=1. See also the discussion on LaTeX Stack Exchange⁸ 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 LATEX 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 LATEX 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 **TAttFill**⁹. 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.

File I: svg.dtx Date: 2020/01/13 v2.02e

⁸http://tex.stackexchange.com/questions/76273/

⁹http://root.cern.ch/root/html/TAttText.html

```
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 I^AT_EX 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 $\langle texmf \rangle / doc/latex/svg/examples/$, 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->SetLabelFont(80,"XYZ");
 gStyle->SetTextFont(80);
 gStyle->SetTitleFont(80,""); gStyle->SetTitleFont(80,"XYZ");
 gStyle->SetPalette(1);
                           gStyle->SetOptStat(0);
 // 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->DrawText(0.7, 0.9, "\\larger[2]$z(x,y) = \\frac{1}{\\sigma_x\\sigma_y"
             \[ (y-\mu_y)^2 {2\sigma_y^2} \ \] 
             "\\right)$");
 // Print the plot.
 gPad->Print("root.svg");
```

where the text produced within the 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}}%
```

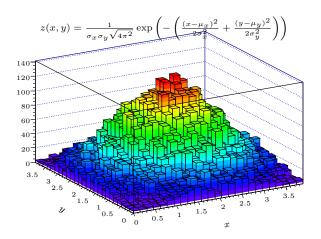


Figure 3: Rendering of a ${f ROOT}$ plot—no more ${\it Comic\ CERNs}$

```
\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 requires package iftex for detecting the used LaTeX engine, scrbase for options processing, pdftexcmds for pdfTeX primitives when using LuaTeX, shellesc and ifplatform for engine independent access to systems commands and files as well as graphicx for the inclusion of PDF files. The usage of packages xcolor and transparent can be switched of with the corresponding options. Package svg-extract only needs package svg itself.

```
1 \langle *base \rangle
2 \RequirePackage{iftex}[2019/11/07]
3 \RequirePackage{scrbase}[2019/12/23]
4 \RequirePackage{pdftexcmds}[2016/05/21]
5 \RequirePackage{shellesc}[2016/06/07]
6 \RequirePackage{trimspaces}[2009/09/17]
7 \RequirePackage{graphicx}[1999/02/16]
8 \langle /base \rangle
9 \langle *extract \rangle
10 \RequirePackage{svg}[2017/03/27]
11 \langle /extract \rangle
```

A.2. Helper macros

\svg@tempa \svg@tempb \svg@box \if@svg@tempswa Internal temporary macros. The catcode for double quotes are also temporarily changed.

```
12 \newcommand*\svg@tempa{}
13 \newcommand*\svg@tempb{}
14 \newbox\svg@box
15 \newif\if@svg@tempswa
16 \edef\svg@catcodecodes@restore{%
17 \catcode'\noexpand\"\the\catcode'\"\relax%
18 }
19 \@makeother\"%
```

B. Including SVG files with package svg

B.1. Options

All options, which can be set either as package options or with \svgsetup, as well as the optional parameters for both user commands \includesvg[$\langle parameters \rangle$] { $\langle svg\ filename \rangle$ } and \includeinkscape[$\langle parameters \rangle$] { $\langle graphic\ filename \rangle$ } are defined with the interface provided by package scrbase.

```
20 \DefineFamily{SVG}
21 \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.

```
22 \newcommand*\svg@deprecated@key[3][svg]{%
23 \PackageWarning{#1}{%
24 The option key '#2' is deprecated.\MessageBreak%
```

File II: svg.dtx Date: 2020/01/13 v2.02e

```
25  It's recommended to use '#3'\MessageBreak%
26  instead%
27  }%
28  \FamilyOptions{SVG}{#3}%
29 }
```

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

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.

```
30 \newif\if@svg@use@xcolor
31 \FamilyBoolKey{SVG}{usexcolor}{@svg@use@xcolor}
32 \DeclareOption{noxcolor}{\FamilyOptions{SVG}{usexcolor=false}}
33 \newif\if@svg@use@transparent
34 \FamilyBoolKey{SVG}{usetransparent}{@svg@use@transparent}
35 \DeclareOption{notransparent}{\FamilyOptions{SVG}{usetransparent=false}}
```

They are only available during the loading process of package \mathbf{svg} .

```
36 \AtEndOfPackage{%
    \RelaxFamilyKey{SVG}{usexcolor}%
37
    \RelaxFamilyKey{SVG}{usetransparent}%
38
    \if@svg@use@xcolor%
39
40
      \RequirePackage{xcolor}[2016/05/11]%
41
    \else%
      \AfterPackage*{xcolor}{%
42
        \PackageWarning{svg}{Package 'xcolor' was loaded anyway}%
43
      }%
44
45
    \fi%
    \if@svg@use@transparent%
46
      \RequirePackage{transparent}[2016/05/16]%
47
48
      \AfterPackage*{transparent}{%
49
        \PackageWarning{svg}{Package 'transparent' was loaded anyway}%
50
      }%
51
52
    \fi%
53 }
```

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.

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) or the command line call can be forced with every LATEX run (inkscape=forced). Additionally, option inkscape can be used as wrapper for options inkscapeformat, inkscapelatex, inkscapearea and inkscapedpi, which are declared later.

```
54 \newcommand*\svg@ink@mode{}
55 \DefineFamilyKey{SVG}{inkscape}[true]{%
56
    \lowercase{\svg@sanitize@dq\svg@tempb{#1}}%
    \FamilySetNumerical{SVG}{inkscape}{svg@tempa}{%
57
      {false}{0},{off}{0},{no}{0},%
58
      {true}{1}, {on}{1}, {yes}{1}, {onlynewer}{1}, {newer}{1}, %
59
60
      {force}{2},{forced}{2},{overwrite}{2},%
      {pdf}{3},{eps}{4},{ps}{5},{png}{6},%
61
      {drawing}{7},{crop}{7},%
62
      {page}{8},{nocrop}{8},%
63
      {tex}{9},{latex}{9},{exportlatex}{9},{latexexport}{9},%
64
      {notex}{10}, {nolatex}{10}, {noexportlatex}{10}, {nolatexexport}{10}, %
65
```

```
66 {latexnoexport}{10},{raw}{10},{plain}{10},{simple}{10}%
67 }{\svg@tempb}%
68 \ifx\FamilyKeyState\FamilyKeyStateProcessed%
```

Setting the mode for invoking *Inkscape*...

```
69 \ifnum\svg@tempa<\thr@@\relax%
70 \let\svg@ink@mode\svg@tempa%
71 \else%
```

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

```
\ifcase\svg@tempa\relax\or\or\or\ pdf
72
73
           \FamilyOptions{SVG}{inkscapeformat=pdf}%
74
        \or% eps
75
          \FamilyOptions{SVG}{inkscapeformat=eps}%
76
        \or% ps
          \FamilyOptions{SVG}{inkscapeformat=ps}%
77
        \or% png
78
          \FamilyOptions{SVG}{inkscapeformat=png}%
79
        \or% drawing
80
81
          \FamilyOptions{SVG}{inkscapearea=drawing}%
82
        \or% page
          \FamilyOptions{SVG}{inkscapearea=page}%
83
84
85
          \FamilyOptions{SVG}{inkscapelatex=true}%
86
        \or% notex
          \FamilyOptions{SVG}{inkscapelatex=false}%
87
        \fi%
88
      \fi%
89
```

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

```
90 \else% dpi

91 \def\svg@tempa##1dpi##2\@nil{%

92 \Ifstr{##2}{dpi}{\FamilyOptions{SVG}{inkscapedpi=##1}}{}%

93 }%

94 \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.

```
95 \ifx\FamilyKeyState\FamilyKeyStateProcessed\else%
```

Splitting executable from options with delimitted 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.

```
96
        \svg@quotes@remove[{#1}]{\svg@tempb}%
97
        \def\svg@tempa##1-##2\@nil{%
          \IfArgIsEmpty{##2}{\def\svg@tempb{}}{%
98
            99
            \svg@tempa##2\@nil%
100
          }%
101
102
          \edef\svg@tempa{\trim@spaces{##1}}%
103
104
        \edef\svg@tempb{%
          \noexpand\svg@tempa\svg@tempb-\noexpand\@nil%
105
106
        }%
107
        \svg@tempb%
        \if@svg@quotes@found%
108
          \edef\svg@tempa{"\svg@tempa"}%
109
        \fi%
110
        \PackageWarning{svg}{%
111
          Setting the executable%
112
          \ifx\svg@tempb\@empty\else%
113
```

```
116
                                    \MessageBreak%
                                    for Inkscape should be done with options\MessageBreak%
                       117
                                    'inkscapeexe=\svg@tempa'%
                       118
                                    \ifx\svg@tempb\@empty\else%
                        119
                       120
                                      \MessageBreak and 'inkscapeopt=\svg@tempb'%
                       121
                                    \fi.\MessageBreak%
                                    Nevertheless, this was done by now anyway%
                       122
                                  }%
                       123
                                  \edef\svg@tempa{%
                        124
                                    \noexpand\FamilyOptions{SVG}{inkscapeexe=\svg@tempa}%
                        125
                                    \ifx\svg@tempb\@empty\else%
                        126
                        127
                                      \label{lem:lyOptions} $$ \operatorname{SVG}_{inkscapeopt=\svg@tempb}% $$ $$ \operatorname{SVG}_{inkscapeopt=\svg@tempb}% $$ $$ $$ $$ $$
                        128
                                    fi%
                        129
                                  }%
                        130
                                  \svg@tempa%
                        131
                               \fi%
                        132
                             \fi%
                       133 }
                      Package options which can be used to switch functionality on or off during the loading of
             on (opt.)
           off (opt.)
                      package svg.
                        134 \DeclareOption{on}{\FamilyOptions{SVG}{inkscape=true}}
                       135 \DeclareOption{off}{\FamilyOptions{SVG}{inkscape=false}}
inkscapeformat (opt.)
                       With option inkscapeformat the output format of the Inkscape export function, which
    \svg@ink@format
                      is called via \ShellEscape, can be configured. It is set to pdf or, if dvi output could be
                       detected, to eps during initialization.
                       136 \newcommand*\svg@ink@format{pdf}
                       137 \ifxetex\else\ifpdf\else
                       138 \renewcommand*\svg@ink@format{eps}
                       139 \fi\fi
                       140 \DefineFamilyKey{SVG}{inkscapeformat}{%
                       141
                             \lowercase{\def\svg@tempa{#1}}%
                       142
                             \FamilySetNumerical{SVG}{inkscapeformat}{svg@tempa}{%
                        143
                                {pdf}{0},{eps}{1},{ps}{2},{png}{3}%
                        144
                             }{\svg@tempa}%
                        145
                             \ifx\FamilyKeyState\FamilyKeyStateProcessed%
                        146
                               \ifcase\svg@tempa\relax% latex
                                  \renewcommand*\svg@ink@format{pdf}%
                        147
                               \or% eps
                        148
                                  \renewcommand*\svg@ink@format{eps}%
                        149
                        150
                               \or% ps
                        151
                                  \renewcommand*\svg@ink@format{ps}%
                        152
                                \or% png
                        153
                                  \renewcommand*\svg@ink@format{png}%
                               \fi%
                        154
                        155
                             \fi%
                        156 }
                       This option controls whether the Inkscape export will be invoked with or without the
 inkscapelatex (opt.)
                       generation of a seperate LATEX file.
         latex (opt.)
           tex (opt.)
                        157 \newif\if@svg@ink@latex
     \svg@ink@latex
                        158 \verb|\FamilyBoolKey{SVG}{inkscapelatex}{@svg@ink@latex}|
                        159 \FamilyBoolKey{SVG}{latex}{@svg@ink@latex}
                       160 \FamilyBoolKey{SVG}{tex}{@svg@ink@latex}
                       The exported area for an Inkscape graphic can be set with this option.
  inkscapearea (opt.)
      \svg@ink@area
```

\space and associated options%

114

115

\fi%

161 \newcommand*\svg@ink@area{}

```
162 \DefineFamilyKey{SVG}{inkscapearea}{%
163
     \FamilySetNumerical{SVG}{inkscapearea}{svg@tempa}{%
164
       {drawing}{0},{crop}{0},%
165
       {page}{1},{nocrop}{1}%
     }{#1}%
166
     \ifx\FamilyKeyState\FamilyKeyStateProcessed%
167
       \ifcase\svg@tempa\relax% drawing
168
         \renewcommand*\svg@ink@area{-D}%
169
       \else% page
170
         \renewcommand*\svg@ink@area{-C}%
171
172
       \fi%
     \fi%
173
174 }
```

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.

```
175 \newcommand*\svg@ink@dpi{}
176 \let\svg@ink@dpi\relax
177 \DefineFamilyKey{SVG}{inkscapedpi}{%
     \FamilyKeyStateUnknownValue%
179
     \svg@ifvalueisrelax{#1}{%
180
       \let\svg@ink@dpi\relax%
       \FamilyKeyStateProcessed%
181
     }{%
182
       \def\svg@tempa##1dpi##2\@nil{\def\svg@tempa{##1}}%
183
       \lowercase{\svg@tempa#1dpi\@nil}%
184
       \Ifnumber{\svg@tempa}{%
185
         \edef\svg@ink@dpi{\svg@tempa}%
186
187
         \FamilyKeyStateProcessed%
188
189
     }%
190 }
191 \DefineFamilyKey{SVG}{inkscapedensity}{\FamilyOptions{SVG}{inkscapedpi=#1}}
```

inkscapeexe (opt.)
 \svg@ink@exe
inkscapeopt (opt.)
 \svg@ink@opt

With these options, the terminal command for invoking *Inkscape* as well as additional options can be defined.

```
192 \newcommand*\svg@ink@exe{inkscape}
193 \DefineFamilyKey{SVG}{inkscapeexe}{%
194 \renewcommand*\svg@ink@exe{#1}%
195 \FamilyKeyStateProcessed%
196 }
197 \newcommand*\svg@ink@opt{}
198 \DefineFamilyKey{SVG}{inkscapeopt}{%
199 \renewcommand*\svg@ink@opt{#1}%
200 \FamilyKeyStateProcessed%
201 }
```

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.

```
202 \DefineFamilyKey{SVG}{svgpath}{%
     \PackageWarning{svg}{%
204
       The key 'svgpath' is deprecated. It's recommended\MessageBreak%
205
       to use '\string\svgpath' instead%
206
     }%
207
     \ifx\svgpath\@undefined%
       \AtEndOfPackage{\svgpath{{#1}}}%
208
209
     \else%
       \svgpath{{#1}}%
210
211
     \fi%
```

```
212 \FamilyKeyStateProcessed%
213 }

svgextension (opt.)

extension (opt.)

ext (opt.)

\svg@file@ext

214 \newcommand*\svg@file@ext{svg}

215 \DefineFamilyKey{SVG}{svgextension}{%}

The extension should be in lower case letters.

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

Remove leading dots from the extension.

```
217 \svg@remove@leadingchar.\svg@file@ext%
218 }
219 \DefineFamilyKey{SVG}{extension}{\FamilyOptions{SVG}{svgextension=#1}}
220 \DefineFamilyKey{SVG}{ext}{\FamilyOptions{SVG}{svgextension=#1}}
```

B.1.3. Setting output folder

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

The option inkscapepath controls, in which folder the results of the *Inkscape* export will be located. With option inkscapename the name of the exported file itself can be changed.

```
221 \newcommand*\svg@out@path{}
222 \newcommand*\svg@out@name{\svg@file@name\svg@file@suffix}
223 \newcommand*\svg@out@base{\svg@out@path\svg@out@name.\svg@ink@format}
224 \DefineFamilyKey{SVG}{inkscapepath}{%
                 \svg@sanitize@dq\svg@tempb{#1}%
                 \FamilySetNumerical{SVG}{inkscapepath}{svg@tempa}{%
226
227
                         {svgpath}{0},{svgdir}{0},%
228
                         {svgsubpath}{1},{svgsubdir}{1},%
                        \label{lossedir} $$\{basepath\}_{2}, \{basedir}_{2}, \{jobpath}_{2}, \{jobdir}_{2}, \%$
229
                         \begin{tabular}{l}{\bf \{3\},\{basesubdir\}\{3\},\{jobsubpath\}\{3\},\{jobsubdir\}\{3\},\{jobsubdir\}\{3\},\{jobsubdir\}\{3\},\{jobsubdir\}\{3\},\{jobsubdir\}\{3\},\{jobsubdir\}\{3\},\{jobsubpath\}\{3\},\{jobsubdir\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpath\}\{3\},\{jobsubpat
230
                 }{\svg@tempb}%
231
232
                 \ifx\FamilyKeyState\FamilyKeyStateProcessed%
                        \ifcase\svg@tempa\relax% svgpath
233
234
                               \renewcommand*\svg@out@path{\svg@file@path}%
235
                        \or% svgsubpath
                               \renewcommand*\svg@out@path{\svg@file@path svg-inkscape/}%
236
237
                        \or% basepath
                               \renewcommand*\svg@out@path{./}%
238
239
                        \or% basesubpath
                               \renewcommand*\svg@out@path{./svg-inkscape/}%
240
                        \fi%
241
                 \else%
242
                         \edef\svg@out@path{\svg@tempb}%
243
                        \svg@normalize@path{\svg@out@path}%
244
                         \FamilyKeyStateProcessed%
245
                 \fi%
246
247 }
248 \DefineFamilyKey{SVG}{inkscapename}{%
                 \renewcommand*\svg@out@name{#1\svg@file@suffix}%
                 \FamilyKeyStateProcessed%
250
251 }
```

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.) These options determine the size of the included graphics. The usage of \relax as value resets the respective option to the default behavior. \svg@param@width height (opt.) $252 \verb|\newcommand*\svg@param@width{\z0}|$ \svg@param@width 253 \DefineFamilyKey{SVG}{width}{% distort (opt.) 254 \FamilyKeyStateUnknownValue% keepaspectratio (opt.) \svg@ifvalueisrelax{#1}{% 255 \if@svg@param@distort 256 \renewcommand*\svg@param@width{\z@}% scale (opt.) 257 \FamilyKeyStateProcessed% \svg@param@scale 258 }{% 259 \FamilySetLengthMacro{SVG}{width}{\svg@param@width}{#1}% 260 $\footnote{MilkeyState}\$ 261 \ifdim\svg@param@width<\z@\relax% 262 \FamilyKeyStateUnknownValue% \fi% 263 264 \fi% }% 265 266 } 267 \newcommand*\svg@param@height{\z@} 268 \DefineFamilyKey{SVG}{height}{% \FamilyKeyStateUnknownValue% 270 \svg@ifvalueisrelax{#1}{% 271 \renewcommand*\svg@param@height{\z@}% 272 \FamilyKeyStateProcessed% 273 }{% 274 \FamilySetLengthMacro{SVG}{height}{\svg@param@height}{#1}% 275 \ifdim\svg@param@height<\z@\relax% 276 277 \FamilyKeyStateUnknownValue% 278 \fi% 279 \fi% 280 281 } 282 \newif\if@svg@param@distort 283 \FamilyBoolKey{SVG}{distort}{@svg@param@distort} 284 \DefineFamilyKey{SVG}{keepaspectratio}[true]{% \FamilySetBool{SVG}{keepaspectratio}{@svg@tempswa}{#1}% 285 286 \if@svg@tempswa% 287 \FamilyOptions{SVG}{distort=false}% 288 289 \FamilyOptions{SVG}{distort=true}% 290 \fi% 291 292 \fi% 293 } 294 \newcommand*\svg@param@scale{1} 295 \DefineFamilyKey{SVG}{scale}{\% \FamilyKeyStateUnknownValue% 296 \svg@ifvalueisrelax{#1}{% 297 298 \renewcommand*\svg@param@scale{1}% \FamilyKeyStateProcessed% 299 300 $\Ifisdimension{#1\p@}{%}$ 301 \ifdim\dimexpr#1\p@\relax>\z@\relax% 302 303 \renewcommand*\svg@param@scale{#1}% 304 \FamilyKeyStateProcessed% \fi% 305 }{}% 306 }% 307 308 } For executing code right before or after the graphic inclusion, two hooks are defined. pretex (opt.)

```
File II: svg.dtx Date: 2020/01/13 v2.02e
```

309 \newcommand*\svg@param@pretex{}

311 \DefineFamilyKey{SVG}{pretex}{%

310 \let\svg@param@pretex\relax

\svg@param@pretex

\svg@param@apptex

apptex (opt.)

postex (opt.)

```
312
     \svg@ifvalueisrelax{#1}{%
313
       \let\svg@param@pretex\relax%
314
       \def\svg@param@pretex{#1}%
315
316
317
     \FamilyKeyStateProcessed%
318 }
319 \newcommand*\svg@param@apptex{}
320 \let\svg@param@apptex\relax
321 \DefineFamilyKey{SVG}{apptex}{%
     \svg@ifvalueisrelax{#1}{%
322
       \let\svg@param@apptex\relax%
323
324
     }{%
325
       \def\svg@param@apptex{#1}%
326
327
     \FamilyKeyStateProcessed%
328 }
329 \DefineFamilyKey{SVG}{postex}{%
     \svg@deprecated@key{postex=#1}{apptex=#1}%
331 }
```

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

For Inkscape 0.91 a bug concerning the LATEX export has been reported (https://bugs.launchpad.net/ubuntu/+source/inkscape/+bug/1417470). Sometimes the LATEX 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.

```
332 \newcounter{svg@param@lastpage}
333 \DefineFamilyKey{SVG}{lastpage}{%
     \FamilySetNumerical{SVG}{lastpage}{svg@tempa}{%
        {false}{0},{off}{0},{no}{0},{ignore}{0},%
336
        {true}{1},{on}{1},{yes}{1},{auto}{1}%
337
     ት{#1}%
     \ifx\FamilyKeyState\FamilyKeyStateProcessed%
338
        \ifcase\svg@tempa\relax% false
339
          \label{lastpage} $$\operatorname{SVG}_{1astpage}_{svg@param@lastpage}_{m@ne}% $$
340
        \or% true
341
          \FamilySetCounter{SVG}{lastpage}{svg@param@lastpage}{\z@}%
342
343
       \fi%
344
     \fi%
345 }
```

draft (opt.)
\if@svg@draft

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

```
346 \newif\if@svg@draft \\ 347 \FamilyBoolKey{SVG}{draft}{@svg@draft} \\ 348 \AtBeginDocument{\if@svg@draft\else\\ifGin@draft\@svg@drafttrue\\fi\fi}
```

B.2. Handling path information

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 IATEX, this is taken into account.

\svg@deactivate@dq

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

```
349 \newcommand*\svg@deactivate@dq{}
350 \AfterPackage+{babel}{%
351 \renewcommand*\svg@deactivate@dq{\bbl@deactivate{"}}%
352 \providecommand*\bbl@deactivate[1]{}%
353 }
```

\svg@sanitize@dq Save expansion of the second argument in the macro from teh first argument with deactivated double quotes.

```
354 \newcommand*\svg@sanitize@dq[2]{%
355 \begingroup%
356 \svg@deactivate@dq%
357 \edef\svg@tempa{\endgroup\def\noexpand#1{#2}}%
358 \svg@tempa%
359}
```

\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.

```
360 \newcommand*\svg@quotes@remove[2][]{%
361
     \begingroup%
362
        \IfArgIsEmpty{#1}{\def\svg@tempb{#2}}{\def\svg@tempb{#1}}%
363
        \svg@sanitize@dq\svg@tempa{\svg@tempb}%
        \expandafter\svg@quotes@check\expandafter{\svg@tempa}%
364
        \expandafter\svg@quotes@@remove\svg@tempa""\@nil%
365
        \edef\svg@tempb{%
366
367
         \endgroup%
         \def\noexpand#2{\svg@tempa}%
368
         \if@svg@quotes@found%
369
            \verb|\noexpand@svg@quotes@foundtrue||
370
371
         \else%
            \noexpand\@svg@quotes@foundfalse%
372
373
          \fi%
374
       ጉ%
375
     \svg@tempb%
376 }
377 \newcommand*\svg@quotes@@remove{}
378 \def\svg@quotes@@remove#1"#2"#3\@nil{%
379
     \IfArgIsEmpty{#2}{%
        \edef\svg@tempa{#1}%
380
     ጉ ና %
381
        \svg@quotes@@remove#1#2#3""\@nil%
382
     }%
383
384 }
```

\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.

```
385 \newif\if@svg@quotes@found
386 \newcommand*\svg@quotes@check[1]{%
387 \expandafter\svg@quotes@check#1"\@nil%
388 }
389 \newcommand*\svg@quotes@check{}
390 \def\svg@quotes@check#1"#2\@nil{%
391 \IfArgIsEmpty{#2}{\@svg@quotes@foundfalse}{\@svg@quotes@foundtrue}%
392 }
```

\svg@remove@leadingchar

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

```
393 \newcommand*\svg@remove@leadingchar[2]{%
394 \begingroup%
395 \svg@sanitize@dq\svg@tempa{#2}%
396 \def\svg@tempb{%
397 \def\svg@tempa####1\@nil{\def\svg@tempa{####1}}%
398 \kernel@ifnextchar#1%
399 {\expandafter\svg@tempa\@gobble}%
```

```
400
            {\svg@tempa}%
401
       }%
402
        \expandafter\svg@tempb\svg@tempa\@nil%
403
        \edef\svg@tempb{%
404
          \endgroup%
405
          \def\noexpand#2{\svg@tempa}%
406
       }%
407
      \svg@tempb%
408 }
```

\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.

```
409 \newcommand*\svg@set@input@path[1][]{%
410 \begingroup%
411 \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.

```
412 \ifx\svg@file@path\@empty\else%

413 \svg@normalize@path{\svg@file@path}%

414 \edef\svg@file@path{{\svg@file@path}}%

415 \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.

```
416 \svg@append@input@path{\svg@file@path}{\svg@input@path}%
417 \svg@append@input@path{\svg@file@path}{\Ginput@path}%
418 \IfArgIsEmpty{#1}{}{\svg@append@input@path{\svg@file@path}{{#1}}}%
419 \svg@append@input@path{\svg@file@path}{\input@path}%
```

Finally, \input@path is set.

```
420 \edef\svg@tempa{%

421 \endgroup%

422 \ifx\svg@file@path\@empty\else%

423 \def\noexpand\input@path{\svg@file@path}%

424 \fi%

425 }%

426 \svg@tempa%

427}
```

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.

```
428 \newcommand*\svg@append@input@path[2]{%
429 \ifx#2\@undefined\else%
430 \edef\svg@tempb{#2}%
431 \expandafter\@tfor\expandafter\svg@tempa\expandafter:\expandafter=%
432 \svg@tempb\do{%
```

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

```
433 \ifx\svg@tempa\@empty\else%
434 \@svg@tempswatrue%
435 \svg@normalize@path{\svg@tempa}%
436 \expandafter\@tfor\expandafter\svg@tempb\expandafter:\expandafter=%
437 #1\do{%
438 \ifx\svg@tempa\svg@tempb%
```

```
439
                 \@svg@tempswafalse%
440
                 \@break@tfor%
441
               \fi%
            }%
442
             \if@svg@tempswa%
443
444
               \edef#1{#1{\svg@tempa}}%
445
            fi%
446
          \fi%
        }%
447
      \fi%
448
449 }
```

\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 \sug@normalize@path.

```
450 \newcommand*\svg@normalize@path[1]{%
     \begingroup%
451
       \svg@quotes@remove[{#1}]{\svg@tempa}%
452
       \ifx\svg@tempa\@empty\relax%
453
454
         \def\svg@tempa{./}%
455
       \expandafter\svg@normalize@@path\svg@tempa//\@nil%
456
       \edef\svg@tempb{%
457
         \endgroup%
458
459
         \if@svg@quotes@found%
           \def\noexpand#1{"\svg@tempa"}%
460
461
         \else%
           \def\noexpand#1{\svg@tempa}%
462
463
         \fi%
       }%
464
465
     \svg@tempb%
466 }
467 \newcommand*\svg@normalize@@path{}
468 \def\svg@normalize@@path#1/#2/\@nil{%
469
     \IfArgIsEmpty{#2}{%
       470
471
       \svg@normalize@@path#2/\@nil%
472
       \edef\svg@tempa{#1/\svg@tempa}%
473
    }%
474
475 }
```

\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 reseting the key. Therefore, \svg@ifvalueisrelax checks, whether \relax was used as value or not.

```
476 \newcommand*\svg@ifvalueisrelax[1]{%
     \begingroup%
477
478
        \def\svg@tempa{#1}%
479
        \def\svg@tempb{\relax}%
        \ifx\svg@tempa\svg@tempb\relax%
480
          \aftergroup\@firstoftwo%
481
482
        \else%
483
          \aftergroup\@secondoftwo%
484
        \fi%
     \endgroup%
485
486 }
```

\svg@get@path
\if@svg@file@found
\svg@file@path
\svg@file@name
\svg@file@base
\svg@file@suffix

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.

```
487 \newif\if@svg@file@found
488 \newcommand*\svg@file@path{}
489 \newcommand*\svg@file@name{}
490 \newcommand*\svg@file@base{}
491 \newcommand*\svg@file@suffix{}
492 \newcommand*\svg@get@path[3][\svg@file@ext]{%
493 \begingroup%
```

A maybe given, unneeded file extension is removed.

```
494 \svg@filename@parse[{#1}]{#2}%

495 \IfArgIsEmpty{#1}{%

496 \edef\svg@tempa{\filename@area\filename@base.\filename@ext}%

497 \}{%

498 \edef\svg@tempa{\filename@area\filename@base.#1}%

499 \}%
```

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.

```
500 \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.

```
501
       \@svg@tempswafalse%
502
       \expandafter\IfFileExists\expandafter{\svg@tempa}{%
         \expandafter\svg@quotes@check\expandafter{\svg@tempa}%
503
         \if@svg@quotes@found\else%
504
           \svg@quotes@remove{\@filef@und}%
505
         \fi%
506
507
         \@svg@tempswatrue%
         \edef\@filef@und{\expandafter\trim@spaces\expandafter{\@filef@und}}%
508
509
         \svg@filename@parse[{#1}]{\@filef@und}%
510
       \edef\svg@tempa{%
511
512
         \endgroup%
513
         \if@svg@tempswa%
514
           \noexpand\@svg@file@foundtrue%
515
           \def\noexpand\svg@file@path{\filename@area}%
           \def\noexpand\svg@file@name{\filename@base}%
516
           \def\noexpand\svg@file@base{\filename@area\filename@base}%
517
         \else%
518
519
            \noexpand\@svg@file@foundfalse%
            \def\noexpand\svg@file@path{}%
520
           \def\noexpand\svg@file@name{#2}%
521
522
           \def\noexpand\svg@file@base{#2}%
523
       }%
524
     \svg@tempa%
525
526 }
```

 $\svg@filename@parse$

As the internal LATEX 2ε 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.

```
527 \newcommand*\svg@filename@parse[2][]{% 528 \begingroup%
```

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

```
\svg@sanitize@dq\svg@tempa{#2}%
529
       \expandafter\filename@parse\expandafter{\svg@tempa}%
530
531 % If there are quotes in the file path, the closing one will be found as first
532 % character in \cs{filename@base} as \cs{filename@area} is splitted at the last
533 % slash. This leading quote is removed from \cs{filename@base} with
534 % \cs{svg@remove@leadingchar}.
        \begin{macrocode}
535 %
536
       \svg@quotes@remove{\filename@area}%
537
       \if@svg@quotes@found%
         \edef\filename@area{"\filename@area"}%
538
         \verb|\svg@remove@leadingchar"\filename@base%| \\
539
540
       \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.

```
541 \ifx\filename@ext\relax\else%
542 \svg@quotes@remove{\filename@ext}%
543 \svg@extension@parse{#1}%
544 \if@svg@quotes@found%
545 \edef\filename@base{\filename@base"}%
546 \fi%
547 \fi%
```

Quotes within \filename@base are normalized.

```
548 \svg@quotes@remove{\filename@base}%
549 \if@svg@quotes@found%
550 \edef\filename@base{"\filename@base"}%
551 \fi%
```

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

```
\edef\svg@tempa{%
552
         \endgroup%
553
         \def\noexpand\filename@area{\filename@area}%
554
         \def\noexpand\filename@base{\filename@base}%
555
556
          \ifx\filename@ext\relax%
            \let\noexpand\filename@ext\noexpand\relax%
557
          \else%
558
            \def\noexpand\filename@ext{\filename@ext}%
559
560
       }%
561
     \svg@tempa%
562
563 }
```

\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 occurence 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.

```
564 \newcommand*\svg@extension@parse[1]{%
565 \IfArgIsEmpty{#1}{}{%
566 \@expandtwoargs\Ifstr%
567 {\detokenize\expandafter{\filename@ext}}{\detokenize\expandafter{#1}}{}%
568 \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.

```
569 \edef\svg@tempa{%

570 \def\noexpand\svg@tempa{}%

571 \let\noexpand\svg@tempb\relax%
```

```
572 \noexpand\svg@extension@@parse%
573 \filename@ext.\noexpand\@nil#1\noexpand\@nil%
574 }%
575 \svg@tempa%
576 \edef\svg@tempa{%
577 \endgroup%
```

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

```
578
           \ifx\svg@tempb\relax%
579
            \let\noexpand\filename@ext\relax%
580
581
           \else%
            \def\noexpand\filename@ext{\svg@tempb}%
582
583
           \fi%
         }%
584
585
       \svg@tempa%
     }%
586
587
    }%
588 }
```

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

```
589 \newcommand*\svg@extension@@parse{}
590 \def\svg@extension@@parse#1.#2\@nil#3\@nil{%
591 \edef\svg@tempa{\svg@tempa.#1}%
592 \IfArgIsEmpty{#2}{}{%
593 \Ifstr{\detokenize{#2}}{\detokenize{#3.}}{%
```

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

```
594 \edef\svg@tempb{#3}%

595 }{%

596 \svg@extension@@parse#2\@nil#3\@nil%

597 }%

598 }%

599}
```

\svg@file@missing

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

```
600 \newcommand*\svg@file@missing[3][]{%
 601
                          \begingroup%
602
                                      \svg@quotes@remove[{#2}]{\svg@tempa}%
                                      \svg@filename@parse[{#1}]{\svg@tempa}%
 603
                                     \IfArgIsEmpty{#1}{%
 604
                                                \svg@quotes@remove[{#3}]{\svg@tempb}%
605
                                               \def\svg@tempa{%
606
                                                         Did you run the export with Inkscape? There's no file\MessageBreak%
607
608
                                                          '\filename@area\filename@base.\filename@ext'\MessageBreak%
 609
                                                         although '\svg@tempb' was found.%
                                               }%
 610
                                    }{%
 611
612
                                                \egli{math} \egl
613
```

Collecting all considered path for the error message.

```
614 \edef\svg@tempb{#3}%
615 \Ifstr{\svg@tempb}{./}{\let\svg@tempb\@empty}{}%
616 \ifx\svg@tempb\@empty%
617 \svg@set@input@path%
618 \else%
619 \svg@set@input@path[\svg@tempb]%
620 \fi%
```

```
621
          \ifx\input@path\@undefined%
 622
            \def\svg@tempb{No additional path was given.}%
 623
          \else%
             \def\svg@tempb{Following folders have additionally been searched:}%
 624
 625
            \expandafter\@tfor\expandafter\svg@tempa\expandafter:\expandafter=%
 626
                 \input@path\do{%
 627
               \edef\svg@tempb{\svg@tempb\noexpand\MessageBreak\svg@tempa}%
            }%
 628
          \fi%
 629
The error message itself.
          \def\svg@tempa{%
 630
            There's no file '\filename@base.\filename@ext'\MessageBreak%
 631
            \ifx\filename@area\@empty%
 632
               neither in the current directory nor any other searched\MessageBreak%
 633
 634
               path given by \string\svgpath\space or \string\graphicspath.%
 635
               \MessageBreak\svg@tempb%
 636
            \else%
               in folder '\filename@area'.%
 637
             \fi%
 638
          }%
 639
 640
        }%
        \PackageError{svg}{%
 641
          File '\filename@base.\filename@ext' is missing%
        }{\svg@tempa}%
 643
 644
      \endgroup%
```

\svg@iffilenewer

645 }

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 definied.

```
646 \newcommand*\svg@iffilenewer[2]{\@gobbletwo}
647 \ifx\pdf@filemoddate\@undefined
648
     \ifx\filemoddate\@undefined\else
649
        \ifx\strcmp\@undefined\else
          \renewcommand*\svg@iffilenewer[2]{%
650
651
            \begingroup%
              \edef\svg@tempa{\filemoddate{#1}}%
652
              \edef\svg@tempb{\filemoddate{#2}}%
653
654
              \ifnum\strcmp{\svg@tempa}{\svg@tempb}>\z@\relax%
                \aftergroup\@firstoftwo%
655
656
657
                \aftergroup\@secondoftwo%
658
              \fi%
659
            \endgroup%
         ጉ%
660
       \fi
661
     \fi
662
663 \else
     \ifx\pdf@strcmp\@undefined\else
664
665
        \renewcommand*\svg@iffilenewer[2]{%
666
          \begingroup%
667
            \edef\svg@tempa{\pdf@filemoddate{#1}}%
668
            \edef\svg@tempb{\pdf@filemoddate{#2}}%
669
            \ifnum\pdf@strcmp{\svg@tempa}{\svg@tempb}>\z@\relax%
              \aftergroup\@firstoftwo%
670
            \else%
671
              \aftergroup\@secondoftwo%
672
            \fi%
673
674
          \endgroup%
675
       }%
676
     \fi
677\fi
```

File II: svg.dtx Date: 2020/01/13 v2.02e

B.3. Optional Parameters for user commands

\svg@local@param@set \svg@local@param@use \svg@local@param@def 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.

```
678 \newcommand*\svg@local@param@set[1]{%
679 \svg@local@param@use%
680 \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.

681 \Ifstr{\svg@ink@format}{png}{\FamilyOptions{SVG}{inkscapelatex=false}}{}%

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

```
\@svg@tempswafalse%
682
     \ifdim\svg@param@width>\z@\relax\ifdim\svg@param@height>\z@\relax%
683
       \@svg@tempswatrue%
684
685
     fi\fi
     \if@svg@tempswa\else%
686
       \FamilyOptions{SVG}{distort=false}%
687
688
689 }
690 \newcommand*\svg@local@param@use{}
691 \newcommand*\svg@local@param@def[1]{%
     \edef\svg@local@param@use{%
692
       \unexpanded\expandafter{\svg@local@param@use}\unexpanded{#1}%
693
694
     }%
695 }
```

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

```
696 \langle *body \rangle
697 \DefineFamilyMember[.param]{SVG}
698 \langle /body \rangle
```

B.4. User commands

\svgsetup \setsvg The macro \svgsetup can be used to change options after loading the package svg both in preamble and the document body. For compatibility reasons, \setsvg is also defined.

```
699 \newcommand*\svgsetup{\FamilyOptions{SVG}} 700 \newcommand*\setsvg{\FamilyOptions{SVG}} \label{eq: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.

```
701 \newcommand*\svg@input@path{}
702 \let\svg@input@path\input@path
703 \newcommand*\svgpath[1]{%
     \def\svg@tempa##1\@nil{%
704
        \ifx\svg@tempb\bgroup%
705
         \def\svg@input@path{#1}%
706
        \else%
707
          \def\svg@input@path{{#1}}%
708
709
       \fi%
710
     }%
711
     \futurelet\svg@tempb\svg@tempa#1\@nil%
712 }
```

File II: svg.dtx Date: 2020/01/13 v2.02e

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

```
713 \newcommand*\includesvg[2][]{%
714 \begingroup%
```

Checking for deprecated commands \svgwidth and \svgscale.

715 \svg@deprecated@param%

```
Most of the optional parameters have the same effect as the identically named options.
      inkscape (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.)
                                 \svg@local@param@set{#1}%
                         716
   inkscapeopt (param.)
  svgextension (param.)
                        The file suffix used by both packages svg and svg-extract.
         width (param.)
        height (param.)
                                 \if@svg@ink@latex%
       distort (param.)
                                   \edef\svg@file@suffix{_\svg@file@ext-tex}%
         scale (param.)
                                 \else%
                         719
        pretex (param.)
                                   \edef\svg@file@suffix{_\svg@file@ext-raw}%
                         720
        apptex (param.)
                         721
                                 \fi%
         draft (param.)
                                 \@onelevel@sanitize\svg@file@suffix%
                         722
```

Searching all given paths for the relevant SVG file.

```
723 \svg@get@path{#2}{}%
724 \if@svg@file@found%
```

Running the export with Inkscape (if necessary) and checking the required files for graphic inclusion.

```
725
                                                                    \svg@ink@run%
 726
                                                                     \IfFileExists{\svg@out@base}{}{%
 727
                                                                                   \@svg@file@foundfalse%
                                                                                   \label{lem:sing} $$\sup_{svg@file@base.\svg@file@ext}% $$ \end{substitute} $$ \end{substitute} $$ \end{substitute} $$\sum_{svg@file@ext}% $$\end{substitute} $$ \end{substitute} $$ \end{substitute} $$\sum_{svg@file@ext}% $$\end{substitute} $$\end{substitute} $$\end{substitute} $$\end{substitute} $$\sum_{svg@file@ext}% $$\end{substitute} $$\end{substitute} $$\end{substitute} $$\end{substitute} $$\end{substitute} $$\sum_{svg@file@ext}% $$\end{substitute} $$\end{substitute} $$\end{substitute} $$\end{substitute} $$\end{substitute} $$\sum_{svg} $$\end{substitute} $$\end{substit} $$\end{substitute} 
728
                                                                    ጉ%
 729
                                                                    \if@svg@ink@latex%
 730
                                                                                   \IfFileExists{\svg@out@base_tex}{}{%
 731
                                                                                                  \@svg@file@foundfalse%
 732
                                                                                                   \svg@file@missing{\svg@out@base_tex}{\svg@file@base.\svg@file@ext}%
 733
                                                                                   }%
 734
                                                                     \fi%
 735
```

Include the resulting graphic file and maybe extract independent files.

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

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

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

739 \fi%

740 \else%
```

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

```
741 \svg@file@missing[\svg@file@ext]{\svg@file@base}{}%
742 \fi%
743 \endgroup%
744}
```

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

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

File II: svg.dtx Date: 2020/01/13 v2.02e

angle (param.) The parameters angle and origin are definied as pendants to the keys provided by origin (param.) \includegraphics.

```
748 \newcommand*\svg@param@angle{0}
749 \svg@local@param@def{%
      \label{lem:condition} $$ \operatorname{SVG}_{angle}_{\%} $$ $$ \operatorname{SVG}_{angle}_{\%} $$
750
        \Ifisdimension{#1\p0}{%}
751
           \renewcommand*\svg@param@angle{#1}%
752
           \FamilyKeyStateProcessed%
753
754
        }{}%
755
     }%
756 }
757 \newcommand*\svg@param@origin{c}
758 \svg@local@param@def{%
      \DefineFamilyKey[.param]{SVG}{origin}[c]{%
759
         \renewcommand*\svg@param@origin{#1}%
760
761
         \FamilyKeyStateProcessed%
     }%
762
763 }
```

\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.

```
764 \newcommand*\includeinkscape[2][]{%
765 \begingroup%
```

Checking for deprecated commands \svgwidth and \svgscale.

```
766 \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.

```
767
                        \svg@filename@parse{#2}%
768
                        \ifx\filename@ext\relax\else%
                              \svg@quotes@remove{\filename@ext}%
769
770
                              \expandafter\lowercase\expandafter{%
771
                                     \verb|\expandafter\filename@ext\expandafter\filename@ext\expandafter\filename@ext\expandafter\filename@ext\expandafter\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\filename\fil
772
                              }%
773
                              \def\svg@tempb##1_tex##2\@nil{%
                                     \IfArgIsEmpty{##1}{}{\def\filename@ext{##1}}%
774
775
                                     \Ifstr{##2}{_tex}{\@svg@tempswatrue}{\@svg@tempswafalse}%
776
                              \@svg@tempswafalse%
777
                              \label{lem:pa:=pdf} $$ \operatorname{pg}\sigma:=\left\{pdf\right\}\left\{ps\right\}\left\{png\right\}\do\left\{\%\right\}. $$
778
779
                                     \begingroup%
                                           \expandafter\svg@tempb\filename@ext_tex\@nil%
780
                                           \svg@extension@parse{\svg@tempa}%
781
                                           \ifx\filename@ext\relax%
782
783
                                                  \def\svg@tempb{\endgroup}%
784
                                           \else%
                                                  \edef\svg@tempb{%
785
786
                                                         \endgroup%
                                                         \noexpand\FamilyOptions{SVG}{inkscapeformat=\svg@tempa}%
787
                                                        \if@svg@tempswa%
788
789
                                                               \noexpand\FamilyOptions{SVG}{inkscapelatex=true}%
790
                                                         \fi%
                                                         \def\noexpand\filename@base{\filename@base}%
791
                                                         \def\noexpand\filename@ext{\filename@ext}%
792
                                                         \noexpand\@svg@tempswatrue%
793
                                                 }%
794
795
                                           \fi%
796
                                     \svg@tempb%
```

Break for loop, if valid extension was found.

```
797 \if@svg@tempswa%
798 \@break@tfor%
799 \fi%
800 }%
```

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

```
801 \if@svg@tempswa\else%

802 \svg@extension@parse{\svg@ink@format}%

803 \fi%

804 \fi%
```

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

```
{\tt 805} \qquad {\tt \svg@local@param@set{\#1}\%}
```

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

```
apptex (param.)
draft (param.)
lastpage (param.)
angle (param.)
```

origin (param.)

scale (param.)

pretex (param.)

 $\label{lem:continuous} $806 \qquad \svg@get@path[\svg@ink@format]{\filename@area\filename@base}{\svg@out@path}\% $807 \qquad \svg@file@found\% $$$

Checking the required files for graphic inclusion.

```
\edef\svg@out@name{\svg@file@name}%
808
         \edef\svg@out@base{\svg@file@path\svg@file@name.\svg@ink@format}%
809
810
         \if@svg@ink@latex%
811
           \IfFileExists{\svg@out@base_tex}{}{%
812
             \@svg@file@foundfalse%
813
              \svg@file@missing{\svg@out@base_tex}{\svg@out@base}%
           }%
814
         \fi%
815
```

Include the resulting graphic file and maybe extract independent files.

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

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

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

819 \fi%

820 \else%
```

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

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

822 \fi%

823 \endgroup%

824 }
```

B.5. Auxiliary macros

\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.

```
825 \newcommand*\svg@deprecated@param{%
826 \@svg@tempswafalse%
827 \ifx\svgwidth\@undefined\else%
828 \edef\svg@tempa{\noexpand\FamilyOptions{SVG}{width=\svgwidth}}%
829 \svg@tempa%
830 \@svg@tempswatrue%
831 \fi%
832 \ifx\svgscale\@undefined\else%
```

```
833
       \edef\svg@tempa{\noexpand\FamilyOptions{SVG}{scale=\svgscale}}%
834
       \svg@tempa%
835
       \@svg@tempswatrue%
836
     \fi%
     \if@svg@tempswa%
837
       \PackageWarning{svg}{%
838
839
         You should specify the image size with parameters\MessageBreak%
840
          'width' and 'height' or 'scale' instead of using\MessageBreak%
          '\string\svgscale' or '\string\svgwidth'%
841
       ጉ%
842
       \let\svgwidth\@undefined%
843
       \let\svgscale\@undefined%
844
     \fi%
845
846 }
```

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

```
847 \newif\if@svg@ink@run
848 \newcommand*\svg@ink@run{%
849 \ifnum\svg@ink@mode>\z@\relax%
850 \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.

```
851 \@svg@ink@runtrue%
852 \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.

```
853 \svg@iffilenewer{\svg@file@base.\svg@file@ext}{\svg@out@base}{}{%
854 \@svg@ink@runfalse%
855 }%
```

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.

```
856
           \if@svg@ink@latex%
857
              \IfFileExists{\svg@out@base_tex}{%
                \ifnum\pdf@shellescape=\@ne\relax\if@svg@ink@run%
858
859
                  \svg@iffilenewer{\svg@out@base_tex}{\svg@out@base}{%
                    \@svg@ink@runfalse%
860
                    \svg@quotes@remove[\svg@out@base]{\svg@tempa}%
861
                    \PackageWarning{svg}{%
862
                      Since the encountered filedate of file\MessageBreak%
863
                      '\svg@tempa_tex' is newer than \MessageBreak%
864
865
                      '\svg@tempa' it's supposed that\MessageBreak%
866
                      you customized this file. To avoid an accidental\MessageBreak%
                      overwriting of this file, the Inkscape export\MessageBreak%
867
                      won't be done. If you want to overwrite the\MessageBreak%
868
869
                      existing file please choose the parameter\MessageBreak%
870
                      'inkscape=force'%
                    }%
871
                  }{}%
872
                \fi\fi%
873
874
             }{\@svg@ink@runtrue}%
875
           \fi%
876
         \fi%
```

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

```
877 \if@svg@ink@run%
878 \ifnum\pdf@shellescape=\@ne\relax%
```

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

```
\ifx\svg@ink@dpi\relax%
879
                \Ifstr{\svg@ink@format}{png}{%
880
                  \FamilyOptions{SVG}{inkscapedpi=300}%
881
                }{}%
882
              \fi%
883
              \PackageInfo{svg}{%
884
                Calling Inkscape%
886
                \ifx\svg@ink@opt\@empty\else%
887
                  \space with added options '\svg@ink@opt'%
888
                \fi%
              }%
889
```

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

```
\svg@quotes@remove[\svg@file@base]{\svg@tempa}%
890
891
              \svg@quotes@remove[\svg@out@name]{\svg@tempb}%
892
              \ShellEscape{\svg@ink@cmd{\svg@tempa}{\svg@tempb}}%
              \IfFileExists{\svg@out@name.\svg@ink@format}{%
893
                \edef\svg@tempb{\svg@tempb.\svg@ink@format}%
894
                \svg@quotes@remove{\svg@out@base}%
895
896
                \svg@shell@mkdir{\svg@out@path}%
                \svg@shell@move{\svg@tempb}{\svg@out@base}%
897
                \if@svg@ink@latex%
898
                  \svg@shell@move{\svg@tempb_tex}{\svg@out@base_tex}%
899
                \fi%
900
901
             }{%
902
                \PackageWarning{svg}{%
903
                  The export with Inkscape failed for file\MessageBreak%
                  '\svg@tempa.\svg@file@ext'\MessageBreak%
904
                  Troubleshooting: Please check in the log file how\MessageBreak%
905
906
                  the invocation of Inkscape took place and try to\MessageBreak%
907
                  execute it yourself in the terminal%
               }%
908
             }%
909
```

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

```
910
              \svg@quotes@remove[\svg@file@base]{\svg@tempa}%
911
              \PackageWarning{svg}{%
912
913
                You didn't enable 'shell escape' (or 'write18')\MessageBreak%
914
                so it wasn't possible to launch the Inkscape export\MessageBreak%
915
                for '\svg@tempa.\svg@file@ext'%
916
              ጉ%
            \fi%
917
         \fi%
918
919
       \endgroup%
920
     \fi%
921 }
```

\svg@ink@cmd The actual call of Inkscape at command line.

```
922 \newcommand*\svg@ink@cmd[2]{%

923 \svg@ink@exe\space-z\space\svg@ink@area\space%

924 \ifx\svg@ink@dpi\relax\else--export-dpi=\svg@ink@dpi\space\fi%

925 \if@svg@ink@latex--export-latex\space\fi%

926 \svg@ink@opt\space%

927 --file="#1.\svg@file@ext"\space%

928 --export-\svg@ink@format="#2.\svg@ink@format"\space%

929 }
```

\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.

```
930 \newcommand*\svg@get@lastpage[1]{%
931
     \Ifstr{\svg@ink@format}{pdf}{%
932
       \begingroup%
933
         \@tempcnta=\m@ne\relax%
         \ifx\XeTeXpdfpagecount\@undefined%
934
            \ifpdf%
935
              \ifx\pdfximage\@undefined%
936
                \ifx\saveimageresource\@undefined\else%
937
                  \saveimageresource{#1}%
938
939
                  \@tempcnta=\lastsavedimageresourcepages\relax%
940
                \pi\%
941
              \else%
942
                \pdfximage{#1}%
943
                \@tempcnta=\pdflastximagepages\relax%
944
            \fi%
945
         \else%
946
            \@tempcnta=\XeTeXpdfpagecount#1\relax%
947
         \fi%
948
         \ifnum\@tempcnta=\m@ne\relax%
949
950
            \PackageWarning{svg}{%
              It wasn't possible to detect the last page\MessageBreak%
951
952
              of '#1'%
953
           }%
954
         \else%
            \PackageInfo{svg}{Last page of '#1' is \the\@tempcnta}%
955
956
         \fi%
         \edef\svg@tempa{%
957
            \endgroup%
958
            \noexpand\FamilyOptions{SVG}{lastpage=\the\@tempcnta}%
959
960
961
       \svg@tempa%
962
     }{}%
963 }
```

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

```
964 \newcommand*\svg@wrn@scale{%
965
     \ifdim\dimexpr\svg@param@scale\p@\relax=\p@\relax\else%
966
        \@svg@tempswafalse%
967
        \ifdim\svg@param@width>\z@\relax%
          \@svg@tempswatrue%
968
        \fi%
969
        \ifdim\svg@param@height>\z@\relax%
970
          \@svg@tempswatrue%
971
972
        \fi%
        \if@svg@tempswa%
973
974
         \PackageWarning{svg}{%
            The parameter 'scale' is only considered if neither\MessageBreak%
975
            'width' nor 'height' are specified%
976
         }%
977
       \fi%
978
     \fi%
979
980 }
```

\svg@input

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.

```
981 \newcommand*\svg@input{\svg@@input}
982 \newcommand*\svg@@input[2][]{%
983 \IfArgIsEmpty{#1}{}{\svg@local@param@set{#1}}%
984 \svg@set@input@path%
985 \if@svg@draft%
986 \@svg@ink@latexfalse%
987 \fi%
```

In order to support file names with multiple dots, the second argument is parsed and only the part after the last dot is stroed in \svg@tempb as extension. Everything before is stored in \svg@tempa.

```
\def\svg@tempb##1.##2\@nil{%
988
989
        \IfArgIsEmpty{##2}{%
990
          \def\svg@tempb{##1}%
991
          \edef\svg@tempa{\svg@tempa.##1}%
992
          \svg@tempb##2\@nil%
993
        ጉ%
994
995
      }%
996
      \edef\svg@tempa{%
997
        \def\noexpand\svg@tempa{}%
        \noexpand\svg@tempb#2.\noexpand\@nil%
998
999
      }%
1000
      \svg@tempa%
```

Afterwards \svg@tempa is defined with the file name within enclosing braces followed by the extension—only if the file name itself contains any dots— and \svg@tempb holds the original file name plus extension without enclosing braces.

```
\svg@remove@leadingchar.\svg@tempa%
1001
1002
      \begingroup%
1003
        \expandafter\filename@parse\expandafter{\svg@tempa}%
        \edef\svg@tempa{%
1004
          \endgroup%
1005
1006
          \ifx\filename@ext\relax%
1007
            \edef\noexpand\svg@tempa{\svg@tempa.\svg@tempb}%
1008
1009
             \edef\noexpand\svg@tempa{{\svg@tempa}.\svg@tempb}%
1010
          \fi%
1011
        }%
1012
      \svg@tempa%
1013
      \edef\svg@tempb{#2}%
```

If the export with *Inkscape* was done with LaTeX 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.

```
\if@svg@ink@latex%
1014
        \svg@patches{\svg@tempa}%
1015
1016
        \ifnum\value{svg@param@lastpage}=\z@\relax%
          \expandafter\svg@get@lastpage\expandafter{\svg@tempb}%
1017
        \fi%
1018
        \edef\svg@tempa{%
1019
1020
          \ifx\svg@param@pretex\relax\else%
             \noexpand\svg@param@pretex%
1021
1022
          \noexpand\input{\svg@tempb_tex}%
1023
1024
          \ifx\svg@param@apptex\relax\else%
1025
             \noexpand\svg@param@apptex%
1026
          \fi%
        }%
1027
```

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

```
1028 \if@svg@param@distort%
```

```
1029 \def\svg@tempb{\resizebox*{\svg@param@width}{\svg@param@height}}%
1030 \else%
1031 \let\svg@tempb\@firstofone%
1032 \fi%
1033 \sbox\svg@box{\svg@tempb{\svg@tempa}}%
```

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

```
\ifdim\dimexpr\svg@param@angle\p@\relax=\z@\relax%
1034
          \let\svg@tempb\@firstofone%
1035
        \else%
1036
           \edef\svg@tempb{%
1037
1038
             \noexpand\rotatebox[origin=\svg@param@origin]{\svg@param@angle}%
          }%
1039
1040
        \fi%
        \svg@tempb{\usebox\svg@box}%
1041
1042
      \else%
```

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

```
\svg@wrn@scale%
1043
        \edef\svg@tempb{%
1044
1045
          draft\if@svg@draft\else=false\fi,%
          scale=\svg@param@scale,%
1046
1047
          keepaspectratio\if@svg@param@distort=false\fi%
1048
        \ifdim\svg@param@height>\z@\relax%
1049
          \edef\svg@tempb{\svg@tempb,height=\svg@param@height}%
1050
1051
1052
        \ifdim\svg@param@width>\z@\relax%
          \edef\svg@tempb{\svg@tempb,width=\svg@param@width}%
1053
1054
        \fi%
        \ifdim\dimexpr\svg@param@angle\p@\relax=\z@\relax\else%
1055
          \edef\svg@tempb{%
1056
            \svg@tempb,origin=\svg@param@origin,angle=\svg@param@angle%
1057
          }%
1058
1059
        \fi%
        \expandafter\includegraphics\expandafter[\svg@tempb]{\svg@tempa}%
1061
1062 }
```

B.6. 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. Those patches are done with \svg@patches.

```
1063 \newcommand*\svg@picture@saved{}
1064 \let\svg@picture@saved\picture
1065 \newcommand*\svg@includegraphics@saved{}
1066 \let\svg@includegraphics@saved\includegraphics
1067 \newcommand*\svg@patches[1]{%
1068 \let\picture\svg@picture@patched%
1069 \let\includegraphics\svg@includegraphics@patched%
1070 \edef\svg@includegraphics@file{#1}%
1071 }
```

\svg@pictur@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.

```
1072 \newcommand*\svg@picture@patched{}
1073 \newcommand*\svg@pictur@patched{}
1074 \long\def\svg@picture@patched#1{\svg@pictur@patched@#1}
```

```
1075 \def\svg@pictur@patched@(#1,#2){%
1076 \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%
1077
        Gscale@div\svg@tempa{#1\p@}{#2\p@}%
1078
        \setlength\unitlength{\svg@param@height}%
1079
        \setlength\unitlength{\svg@tempa\unitlength}%
1080
        \ifdim\svg@param@width>\z@\relax%
1081
1082
          \ifdim\unitlength>\svg@param@width\relax%
1083
            \setlength\unitlength{\svg@param@width}%
1084
          \fi%
1085
        \fi%
1086
      \else%
```

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

```
1087 \ifdim\svg@param@width>\z@\relax%

1088 \setlength\unitlength{\svg@param@width}%

1089 \else%

1090 \setlength\unitlength{\svg@param@scale\unitlength}%

1091 \fi%

1092 \fi%
```

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

```
1093 \svg@picture@saved(#1,#2)%
1094 }
```

\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.

```
1095 \DefineFamily{SVGpatch}
1096 \DefineFamilyMember{SVGpatch}
1097 \newcounter{svg@param@currpage}
1098 \setcounter{svg@param@currpage}{\m@ne}
1099 \FamilyCounterKey{SVGpatch}{page}{svg@param@currpage}
1100 \DefineFamilyKey{SVGpatch}{width}{\FamilyKeyStateProcessed}
1101 \newcommand*\svg@includegraphics@file{}
1102 \newcommand*\svg@includegraphics@patched[2][]{%
1103 \FamilyOptions{SVGpatch}{#1}%
```

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

```
1104 \ifnum\value{svg@param@lastpage}<\z@\relax%
1105 \FamilySetCounter{SVGpatch}{page}{svg@param@currpage}{%
1106 \the\value{svg@param@lastpage}%
1107 }%
1108 \fi%</pre>
```

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.

1109 \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.

```
1110 \svg@includegraphics@saved[{#1}]{\svg@includegraphics@file}%
1111 \fi%
1112 }
```

C. Extracting independent graphic files with svg-extract

C.1. Options

For package **svg-extract** the user interface is extended. The following options can either be set with \svgsetup or be used as local 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.

```
1113 (*base)
1114 \DefineFamilyMember[.dummy]{SVG}
1115 \newcommand*\svg@dummy@key[2][]{%
      \@ifpackageloaded{svg-extract}{}{%
1117
        \IfArgIsEmpty{#1}{%
1118
          \DefineFamilyKey[.dummy]{SVG}{#2}{%
1119
             \PackageWarning{svg}{%
               The option key '#2' can only\MessageBreak%
1120
               be used with package 'svg-extract', but
\MessageBreak%
1121
               you didn't load it%
1122
             }%
1123
             \FamilyKeyStateProcessed%
1124
1125
1126
1127
          \label{lem:lykey[.dummy]} $$ \operatorname{SVG}_{\#2}[{\#1}]_{\%} $$
1128
             \PackageWarning{svg}{%
1129
               The option key '#2' can only\MessageBreak%
               be used with package 'svg-extract', but\MessageBreak%
1130
               you didn't load it%
1131
             }%
1132
             \FamilyKeyStateProcessed%
1133
1134
          }%
1135
```

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

```
1136 \AfterPackage{svg-extract}{\RelaxFamilyKey[.dummy]{SVG}{#2}}%

1137 }%

1138 }

1139 \langle /\base \rangle
```

C.1.1. Controlling the extract process

extract (opt.)
\if@svgx@run

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

```
1140 \land \
```

```
1148
                             {false}{0},{off}{0},{no}{0},%
                     1149
                             {true}{1},{on}{1},{yes}{1},{onlynewer}{1},{newer}{1},%
                     1150
                              {overwrite}{1},{force}{1},{forced}{1},%
                     1151
                              {pdf}{2},{eps}{3},{ps}{4}%
                           }{\svg@tempa}%
                     1152
                           \ifx\FamilyKeyState\FamilyKeyStateProcessed%
                     1153
                             \ifcase\svg@tempa\relax% false
                     1154
                     1155
                                \@svgx@runfalse%
                             \or% true
                     1156
                               \@svgx@runtrue%
                     1157
                              \or% pdf
                     1158
                               \FamilyOptions{SVG}{extractformat=pdf}%
                     1159
                              \or% eps
                     1160
                     1161
                                \FamilyOptions{SVG}{extractformat=eps}%
                     1162
                              \or% ps
                     1163
                                \FamilyOptions{SVG}{extractformat=ps}%
                     1164
                              fi%
                     1165
                           \fi%
                     1166 }
                     1167 (/extract)
                     Package options which can be used to switch functionality on or off during the loading of
           on (opt.)
          off (opt.)
                     package svg-extract.
                     1168 \langle *extract \rangle
                     1169 \DeclareOption{on}{\FamilyOptions{SVG}{extract=true}}
                     1170 \DeclareOption{off}{\FamilyOptions{SVG}{extract=false}}
                     1171 (/extract)
                     Option extractformat controls the output format (pdf/eps/ps). It is set to pdf or, if dvi
extractformat (opt.)
      \svgx@format
                     output could be detected, to eps during initialization.
          pdf (opt.)
                     1172 (*base)
          eps (opt.)
                     1173 \svg@dummy@key{extractformat}
                     1174 \svg@dummy@key[true]{pdf}
                     1175 \svg@dummy@key[true]{eps}
                     1176 (/base)
                     1177 (*extract)
                     1178 \newcommand*\svgx@format{pdf}
                     1179 \ifxetex\else\ifpdf\else
                     1180 \renewcommand*\svgx@format{eps}
                     1181 \fi\fi
                     1182 \DefineFamilyKey{SVG}{extractformat}{%
                     1183
                           \lowercase{\edef\svgx@format{#1}}%
                     1184
                           \FamilyKeyStateProcessed%
                     1185 }
                     1186 \DefineFamilyKey{SVG}{pdf}[true]{%
                     1187
                           \FamilySetBool{SVG}{pdf}{@svg@tempswa}{#1}%
                     1188
                           \ifx\FamilyKeyState\FamilyKeyStateProcessed%
                     1189
                              \if@svg@tempswa%
                                \svgx@ifinlist{pdf}{\svgx@format}{}{%
                     1190
                                  \edef\svgx@format{\svgx@format,pdf}%
                     1191
                                }%
                     1192
                     1193
                                \svg@deprecated@key{pdf}{extractformat={\svgx@format}}%
                     1194
                              \else%
                                \FamilyKeyStateUnknownValue%
                     1195
                              \fi%
                     1196
                     1197
                           \fi%
                     1198 }
                     1199 \DefineFamilyKey{SVG}{eps}[true]{%
                     1200
                           \FamilySetBool{SVG}{eps}{@svg@tempswa}{#1}%
                           \ifx\FamilyKeyState\FamilyKeyStateProcessed%
                     1201
                     1202
                              \if@svg@tempswa%
                     1203
                                \svgx@ifinlist{eps}{\svgx@format}{}{%
                     1204
                                  \edef\svgx@format{\svgx@format,eps}%
```

```
1205
                                     }%
                          1206
                                     \svg@deprecated@key{eps}{extractformat={\svgx@format}}%
                          1207
                                   \else%
                          1208
                                     \FamilyKeyStateUnknownValue%
                                   \fi%
                          1209
                          1210
                                 \fi%
                          1211 }
                          1212 (/extract)
                          For the extraction process, a preamble is necessary for a separate auxiliary IATEX file.
   extractpreamble (opt.)
                          By default, the preamble of the main document is used, which end is detected at
          {\tt preamble}\;({\rm opt.})
                          \begin{document}.
         \svgx@preamble
extractpreambleend (opt.)
                          1213 (*base)
               end (opt.)
                          1214 \svg@dummy@key{extractpreamble}
      \svgx@endpreamble
                          1215 \svg@dummy@key{preamble}
                          1216 \svg@dummy@key{extractpreambleend}
                          1217 \svg@dummy@key{end}
                          1218 (/base)
                          1219 (*extract)
                          1220 \newcommand*\svgx@preamble{\jobname.\svgx@latex@ext}%
                          1221 \DefineFamilyKey{SVG}{extractpreamble}{%
                                 \renewcommand*\svgx@preamble{#1}%
                          1223
                                 \FamilyKeyStateProcessed%
                          1224 }
                          1225 \DefineFamilyKey{SVG}{preamble}{%
                                \svg@deprecated@key[svg-extract]{preamble=#1}{extractpreamble=#1}%
                          1226
                          1227 }
                          1228 \newcommand*\svgx@endpreamble{}
                          1229 \expandafter\def\expandafter\svgx@endpreamble\expandafter{%
                                \csname begin\endcsname{document}%
                          1231 }
                          1232 \DefineFamilyKey{SVG}{extractpreambleend}{%
                          1233
                                \renewcommand*\svgx@endpreamble{#1}%
                          1234
                                 \FamilyKeyStateProcessed%
                          1235 }
                          1236 \DefineFamilyKey{SVG}{end}{%
                                 \svg@deprecated@key[svg-extract]{end=#1}{extractpreambleend=#1}%
                          1237
                          1238 }
                          1239 (/extract)
       extractruns (opt.)
                          With this option, the number of LATEX runs for the separate auxiliary file can be set.
      svgx@runs (counter)
                          1241 \svg@dummy@key{extractruns}
                          1242 (/base)
                          1243 (*extract)
                          1244 \newcounter{svgx@runs}
                          1245 \DefineFamilyKey{SVG}{extractruns}{%
                                 \FamilySetCounter{SVG}{extractruns}{svgx@runs}{#1}%
                          1246
                                 \ifx\FamilyKeyState\FamilyKeyStateProcessed%
                          1247
                                   \ifnum\value{svgx@runs}<\@ne\relax%
                          1248
                          1249
                                     \PackageWarning{svg-extract}{%
                                       The count for runs has to be at least one%
                          1250
                          1251
                          1252
                                     \FamilySetCounter{SVG}{extractruns}{svgx@runs}{\@ne}%
                          1253
                                   \fi%
                          1254
                                 \fi%
                          1255 }
                          1256 (/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 compiler.
          pdflatex (opt.)
        \svgx@latex@exe
                          1257 (*base)
          latexext (opt.)
        \svgx@latex@ext
          latexopt (opt.)
                          File II: svg.dtx Date: 2020/01/13 v2.02e
        \svgx@latex@opt
```

```
1259 \svg@dummy@key{pdflatex}
                     1260 \svg@dummy@key{latexext}
                     1261 \svg@dummy@key{latexopt}
                     1262 (/base)
                     1263 (*extract)
                     1264 \ifxetex
                     1265 \newcommand*\svgx@latex@exe{xelatex}
                     1266 \else\ifluatex
                     1267 \ifpdf
                             \newcommand*\svgx@latex@exe{lualatex}
                     1268
                     1269
                           \else
                     1270
                             \newcommand*\svgx@latex@exe{lualatex --output-format=dvi}
                     1271
                     1272 \leq ifpdf
                     1273 \newcommand*\svgx@latex@exe{pdflatex}
                     1274 \else
                     1275 \newcommand*\svgx@latex@exe{latex}
                     1276 \fi\fi\fi
                     1277 \DefineFamilyKey{SVG}{latexexe}{\%
                           \renewcommand*\svgx@latex@exe{#1}%
                           \FamilyKeyStateProcessed%
                     1279
                     1280 }
                     1281 \DefineFamilyKey{SVG}{pdflatex}{%
                           \svg@deprecated@key[svg-extract]{pdflatex=#1}{latexexe=#1}%
                     1284 \verb|\newcommand*\svgx@latex@ext{tex}|
                     1285 \DefineFamilyKey{SVG}{latexext}{%
                     1286
                           \renewcommand*\svgx@latex@ext{#1}%
                     1287
                           \FamilyKeyStateProcessed%
                     1288 }
                     1289 \newcommand*\svgx@latex@opt{}
                     1290 \DefineFamilyKey{SVG}{latexopt}{%
                     1291
                           \renewcommand*\svgx@latex@opt{#1}%
                           \FamilyKeyStateProcessed%
                     1292
                     1293 }
                     1294 (/extract)
                     Options and macros for calling convert commands, which are supplied by most LATEX 2\varepsilon distri-
     dvipsopt (opt.)
                     butions. These are used to generate all files, which are supported by option extractformat,
   \svgx@dvips@exe
   \svgx@dvips@opt
                     as they don't need an additional application.
   pstoepsopt (opt.)
                     1295 (*base)
 \svgx@pstoeps@exe
                     1296 \svg@dummy@key{dvipsopt}
\svgx@pstoeps@opt
                     1297 \svg@dummy@key{pstoepsopt}
   pstopdfopt (opt.)
                     1298 \svg@dummy@key{pstopdfopt}
\svgx@pstopdf@exe
                     1299 \svg@dummy@key{pdftoepsopt}
\svgx@pstopdf@opt
                     1300 \svg@dummy@key{pdftopsopt}
  pdftoepsopt (opt.)
                     1301 \svg@dummy@key{pdftops}
\svgx@pdftoeps@exe
                     1302 (/base)
                     1303 (*extract)
\svgx@pdftoeps@opt
                     1304 \newcommand*\svgx@dvips@exe{dvips}
   pdftopsopt (opt.)
                     1305 \newcommand*\svgx@dvips@opt{}
\svgx@pdftops@exe
                     1306 \DefineFamilyKey{SVG}{dvipsopt}{%
\svgx@pdftops@opt
                           \renewcommand*\svgx@dvips@opt{#1}%
                     1307
      {\tt pdftops}\ ({\rm opt.})
                     1308
                           \FamilyKeyStateProcessed%
                     1309 }
                     1310 \newcommand*\svgx@pstoeps@exe{ps2eps}
                     1311 \newcommand*\svgx@pstoeps@opt{-B -C}
                     1312 \DefineFamilyKey{SVG}{pstoepsopt}{%
                     1313
                           \renewcommand*\svgx@pstoeps@opt{#1}%
                     1314
                           \FamilyKeyStateProcessed%
                     1315 }
                     1316 \newcommand*\svgx@pstopdf@exe{ps2pdf}
                     1317 \newcommand*\svgx@pstopdf@opt{}
```

1258 \svg@dummy@key{latexexe}

```
1318 \DefineFamilyKey{SVG}{pstopdfopt}{%
1319
      \renewcommand*\svgx@pstopdf@opt{#1}%
1320
      \FamilyKeyStateProcessed%
1321 }
1322 \newcommand*\svgx@pdftoeps@exe{pdftops -eps}
1323 \newcommand*\svgx@pdftoeps@opt{}
1324 \DefineFamilyKey{SVG}{pdftoepsopt}{%
1325
      \renewcommand*\svgx@pdftoeps@opt{#1}%
1326
      \FamilyKeyStateProcessed%
1327 }
1328 \newcommand*\svgx@pdftops@exe{pdftops}
1329 \newcommand*\svgx@pdftops@opt{}
1330 \DefineFamilyKey{SVG}{pdftopsopt}{%
      \renewcommand*\svgx@pdftops@opt{#1}%
1332
      \FamilyKeyStateProcessed%
1333 }
1334 \DefineFamilyKey{SVG}{pdftops}{%
1335
      \PackageWarning{#1}{%
        The option key 'pdftops' is deprecated.\MessageBreak%
1336
        You should use either 'pdftoepsopt' or\MessageBreak%
1337
        'pdftopsopt' instead. See the manual for\MessageBreak%
1338
        more. Nothing was done%
1339
1340
     }%
      \FamilyKeyStateProcessed%
1341
1342 }
1343 (/extract)
```

C.1.2. Invoking external application for graphic conversion

Besides the use of a conversion tool supplied by \LaTeX 2ε , 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.

```
1344 (*base)
1345 \svg@dummy@key[true]{convert}
1346 (/base)
1347 (*extract)
1348 \newif\if@svgx@cnv@run
1349 \newcommand*\svgx@cnv@cmd{}
1350 \DefineFamilyKey{SVG}{convert}[true]{%
      \FamilySetNumerical{SVG}{convert}{svg@tempa}{%
1351
        {false}{0}, {off}{0}, {no}{0}, %
1352
        {true}{1},{on}{1},{yes}{1},{onlynewer}{1},{newer}{1},%
1353
        {overwrite}{1},{force}{1},{forced}{1},%
1354
        \{magick\}\{2\}, \{imagemagick\}\{2\}, \{convert\}\{2\}, \%
1355
1356
        {gs}{3},{ghostscript}{3},%
        {gs64}{4},{ghostscript64}{4},%
1357
        {gs32}{5},{ghostscript32}{5}%
1358
1359
1360
      \ifx\FamilyKeyState\FamilyKeyStateProcessed%
1361
        \ifcase\svg@tempa\relax% false
          \@svgx@cnv@runfalse%
1362
        \or% true
1363
          \@svgx@cnv@runtrue%
1364
        \or% magick
1365
          \@svgx@cnv@runtrue%
1366
          \renewcommand*\svgx@cnv@cmd{\svgx@magick@cmd}%
1367
1368
        \or% gs
          \@svgx@cnv@runtrue%
1369
1370
          \renewcommand*\svgx@cnv@cmd{\svgx@gs@cmd}%
1371
        \or% gs64
1372
          \@svgx@cnv@runtrue%
```

```
1373
          \renewcommand*\svgx@cnv@cmd{\svgx@gs@cmd}%
1374
          \svgx@onlywindows{%
1375
             \renewcommand*\svgx@gs@exe{gswin64c}%
          }%
1376
        \or% gs32
1377
1378
          \@svgx@cnv@runtrue%
1379
          \renewcommand*\svgx@cnv@cmd{\svgx@gs@cmd}%
1380
          \svgx@onlywindows{%
            \renewcommand*\svgx@gs@exe{gswin32c}%
1381
          ጉ%
1382
        \fi%
1383
```

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.

```
1384 \else%
```

Same doing like with option inkscape.

```
\def\svg@tempa##1-##2\@nil{%
1385
          \IfArgIsEmpty{##2}{\def\svg@tempb{}}{%
1386
1387
            \def\svg@tempa##1###1\@nil{\def\svg@tempb{####1}}%
1388
            \svg@tempa#1\@nil\%
          }%
1389
          \def\svg@tempa{##1}%
1390
        }%
1391
        \svg@tempa#1-\@nil%
1392
1393
        \PackageWarning{svg-extract}{%
1394
          Setting the executable%
1395
          \ifx\svg@tempb\@empty\else%
1396
            \space and associated options%
1397
          \fi%
1398
          \MessageBreak%
1399
          for ImageMagick should be done with options\MessageBreak%
1400
          'magickexe=\svg@tempa'%
          \ifx\svg@tempb\@empty\else%
1401
            \MessageBreak and 'magicksetting' and/or 'magickoperator'%
1402
          \fi.\MessageBreak%
1403
          Nevertheless, this was done by now%
1404
1405
          \ifx\svg@tempb\@empty\else%
1406
             , whereby \MessageBreak 'magicksetting=\svg@tempb' was used%
1407
          \fi%
1408
        }%
        \FamilyOptions{SVG}{convert=magick}%
1409
1410
        \edef\svg@tempa{%
          \noexpand\FamilyOptions{SVG}{magickexe=\svg@tempa}%
1411
          \ifx\svg@tempb\@empty\else%
1412
            \noexpand\FamilyOptions{SVG}{magicksetting=\svg@tempb}%
1413
          \fi%
1414
        }%
1415
        \svg@tempa%
1416
1417
      \fi%
1418 }
1419 (/extract)
```

convertformat (opt.)
\svgx@cnv@format

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

```
png (opt.) \\ 1420 \ \langle *base \rangle \\ 1421 \ svg@dummy@key{convertformat} \\ 1422 \ svg@dummy@key[true]{png} \\ 1423 \ \langle /base \rangle \\ 1424 \ \langle *extract \rangle \\ 1425 \ newcommand*\svgx@cnv@format{png} \\ 1426 \ DefineFamilyKey{SVG}{convertformat}{%} \\ \\ \label{eq:png}
```

File II: svg.dtx Date: 2020/01/13 v2.02e

```
1427
      \lowercase{\edef\svgx@cnv@format{#1}}%
1428
      \ifx\svgx@cnv@format\@empty\else%
1429
        \@svgx@cnv@runtrue%
1430
      \FamilyKeyStateProcessed%
1431
1432 }
1433 \DefineFamilyKey{SVG}{png}[true]{%
1434
      \FamilySetBool{SVG}{png}{@svg@tempswa}{#1}%
      \ifx\FamilyKeyState\FamilyKeyStateProcessed%
1435
        \if@svg@tempswa%
1436
          \svgx@ifinlist{png}{\svgx@cnv@format}{}{%
1437
            \edef\svgx@cnv@format{\svgx@cnv@format,png}%
1438
          }%
1439
1440
          \svg@deprecated@key{png}{convertformat={\svgx@cnv@format}}%
1441
        \else%
1442
          \FamilyKeyStateUnknownValue%
1443
        fi%
      \fi%
1444
1445 }
1446 (/extract)
```

convertdpi (opt.)
convertdensity (opt.)
\svgx@cnv@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 recornized by \svgx@ifkeyandval.

```
1447 (*base)
1448 \svg@dummy@key{convertdpi}
1449 \svg@dummy@key{convertdensity}
1450 (/base)
1451 (*extract)
1452 \newcommand*\svgx@cnv@dpi{}
1453 \let\svgx@cnv@dpi\relax
1454 \DefineFamilyKey{SVG}{convertdpi}{%
      \FamilyKeyStateUnknownValue%
1455
1456
      \svgx@ifkeyandval{#1}{%
1457
        \svgx@cnv@get@dpi{##2}%
1458
        \ifx\svg@tempa\relax\else%
1459
           \expandafter\edef\csname svgx@cnv@dpi@##1\endcsname{\svg@tempa}%
1460
          \FamilyKeyStateProcessed%
        \fi%
1461
      }{%
1462
        \svgx@cnv@get@dpi{##1}%
1463
        \ifx\svg@tempa\relax\else%
1464
          \edef\svgx@cnv@dpi{\svg@tempa}%
1465
1466
          \FamilyKeyStateProcessed%
1467
        \fi%
      }%
1468
1469 }
1470 \DefineFamilyKey{SVG}{convertdensity}{\FamilyOptions{SVG}{convertdpi=#1}}
1471 (/extract)
```

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

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.

```
1472 \ \langle *base \rangle \\ 1473 \ svg@dummy@key\{magickexe\} \\ 1474 \ svg@dummy@key\{magicksetting\} \\ 1475 \ svg@dummy@key\{magickoperator\} \\ 1476 \ \langle /base \rangle \\ 1477 \ \langle *extract \rangle \\ 1478 \ newcommand* \ svgx@magick@exe\{\}
```

```
1479 \DefineFamilyKey{SVG}{magickexe}{%
                  1480
                        \renewcommand*\svgx@magick@exe{#1}%
                  1481
                        \FamilyKeyStateProcessed%
                  1482 }
                  1483 \newcommand*\svgx@magick@set{}
                  1484 \DefineFamilyKey{SVG}{magicksetting}{%
                  1485 \svgx@setformatkey{#1}{svgx@magick@set}%
                  1486
                        \FamilyKeyStateProcessed%
                  1487 }
                  1488 \newcommand*\svgx@magick@opr{}
                  1489 \DefineFamilyKey{SVG}{magickoperator}{\%}
                       \svgx@setformatkey{#1}{svgx@magick@opr}%
                        \FamilyKeyStateProcessed%
                  1492 }
                  1493 (/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.)
                  1494 (*base)
\svgx@gs@device
                  1495 \svg@dummy@key{gsexe}
                  1496 \svg@dummy@key{gsopt}
                  1497 \svg@dummy@key{gsdevice}
                  1498 (/base)
                  1499 (*extract)
                  1500 \newcommand*\svgx@gs@exe{}
                  1501 \DefineFamilyKey{SVG}{gsexe}{%
                  1502
                       \renewcommand*\svgx@gs@exe{#1}%
                  1503
                        \FamilyKeyStateProcessed%
                  1504 }
                  1505 \newcommand*\svgx@gs@opt{}
                  1506 \DefineFamilyKey{SVG}{gsopt}{%
                        \svgx@setformatkey{#1}{svgx@gs@opt}%
                  1507
                  1508
                        \FamilyKeyStateProcessed%
                  1509 }
                  1510 \newcommand*\svgx@gs@device{}
                  1511 \DefineFamilyKey{SVG}{gsdevice}{%
                       \svgx@setformatkey{#1}{svgx@gs@device}%
                  1513
                        \FamilyKeyStateProcessed%
                  1514 }
```

C.1.3. Setting output folder

 $1515 \langle /extract \rangle$

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

```
well as the conversion of ImageMagick or Ghostscript will be located. With option
             path (opt.)
                        extractname the name of the extracted and maybe converted file itself can be changed.
     extractname (opt.)
             name (opt.)
                        1516 (*base)
       \svgx@out@path
                        1517 \svg@dummy@key{extractpath}
       \svgx@out@name
                         1518 \svg@dummy@key{path}
     \if@svgx@out@sec
                         1519 \svg@dummy@key{extractname}
svgx@out@count (counter)
                         1520 \svg@dummy@key{name}
                         1521 (/base)
                         1522 (*extract)
                         1523 \verb|\newcommand*\svgx@out@path{}|
                         1524 \DefineFamilyKey{SVG}{extractpath}{%
                               \svg@sanitize@dq\svg@tempb{#1}%
                         1525
                               \FamilySetNumerical{SVG}{extractpath}{svg@tempa}{%
                         1526
                         1527
                                 {svgpath}{0},{svgdir}{0},%
                                 {svgsubpath}{1},{svgsubdir}{1},%
                         1528
                                 {basepath}{2}, {basedir}{2}, {jobpath}{2}, {jobdir}{2}, %
                         1529
```

The option extractpath controls, in which folder the results both of the extraction as

```
1530
        {basesubpath}{3}, {basesubdir}{3}, {jobsubpath}{3}, {jobsubdir}{3}%
1531
      }{\svg@tempb}%
1532
      \ifx\FamilyKeyState\FamilyKeyStateProcessed%
1533
        \ifcase\svg@tempa\relax% svgpath
          \renewcommand*\svgx@out@path{\svg@file@path}%
1534
        \or% svgsubpath
1535
          \renewcommand*\svgx@out@path{\svg@file@path svg-extract/}%
1536
1537
        \or% basepath
          \renewcommand*\svgx@out@path{./}%
1538
        \or% basesubpath
1539
          \renewcommand*\svgx@out@path{./svg-extract/}%
1540
        \fi%
1541
      \else%
1542
        \edef\svgx@out@path{\svg@tempb}%
1543
1544
        \svg@normalize@path{\svgx@out@path}%
1545
        \FamilyKeyStateProcessed%
1546
      \fi%
1547 }
1548 \DefineFamilyKey{SVG}{path}{%
      \svg@deprecated@key[svg-extract]{path=#1}{extractpath=#1}%
1549
1550 }
1551 \newcounter{svgx@out@count}
1552 \newcommand*\svgx@out@name{}
1553 \newif\if@svgx@out@sec
1554 \DefineFamilyKey{SVG}{extractname}{%
      \svg@quotes@remove[{#1}]{\svg@tempb}%
1555
      \FamilySetNumerical{SVG}{extractname}{svg@tempa}{%
1556
        {filename}{0},{name}{0},%
1557
1558
        {filenamenumbered}{1},{namenumbered}{1},%
1559
        {numberedfilename}{1}, {numberedname}{1}, %
        \{numbered\}\{2\}, \{section\}\{2\}, \{numberedsection\}\{2\}, \{sectionnumbered\}\{2\}\%\}
1560
1561
      }{\svg@tempb}%
      \@svgx@out@secfalse%
1562
      \ifx\FamilyKeyState\FamilyKeyStateProcessed%
1563
        \ifcase\svg@tempa\relax% filename
1564
          \renewcommand*\svgx@out@name{\svg@out@name-extract}%
1565
1566
        \or% filenamenumbered
          \renewcommand*\svgx@out@name{\the\value{svgx@out@count}-\svg@out@name}%
1567
        \or% numbered
1568
1569
          \renewcommand*\svgx@out@name{\the\value{svgx@out@count}-\svgx@out@sec}%
1570
          \@svgx@out@sectrue%
        \fi%
1571
1572
      \else%
        \if@svg@quotes@found%
1573
          \edef\svgx@out@name{"\svg@tempb"}%
1574
1575
          \edef\svgx@out@name{\svg@tempb}%
1576
        fi%
1577
        \FamilyKeyStateProcessed%
1578
1579
      \fi%
1580 }
1581 \DefineFamilyKey{SVG}{name}{%}
      \svg@deprecated@key[svg-extract]{name=#1}{extractname=#1}%
1582
1583 }
1584 (/extract)
```

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 reseting 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.

```
1585 (*base)
```

File II: svg.dtx Date: 2020/01/13 v2.02e

```
1586 \svg@dummy@key{extractwidth}
1587 \svg@dummy@key{extractheight}
1588 \svg@dummy@key{extractdistort}
1589 \svg@dummy@key{extractkeepaspectratio}
1590 \svg@dummy@key{extractscale}
1591 (/base)
1592 (*extract)
1593 \newcommand*\svgx@param@width{\svg@param@width}
1594 \DefineFamilyKey{SVG}{extractwidth}{%
      \FamilyKeyStateUnknownValue%
1595
      \svg@ifvalueisrelax{#1}{%
1596
        \renewcommand*\svgx@param@width{\z@}%
1597
1598
        \FamilyKeyStateProcessed%
1599
1600
        \Ifstr{#1}{inherit}{%
1601
          \renewcommand*\svgx@param@width{\svg@param@width}%
1602
          \FamilyKeyStateProcessed%
1603
        }{%
          \FamilySetLengthMacro{SVG}{extractwidth}{\svgx@param@width}{#1}%
1604
          \ifx\FamilyKeyState\FamilyKeyStateProcessed%
1605
            \ifdim\svgx@param@width<\z@\relax%
1606
               \FamilyKeyStateUnknownValue%
1607
            \fi%
1608
          \fi%
1609
        }%
1610
      }%
1611
1612 }
1613 \verb|\newcommand*\svgx@param@height{\svg@param@height}|
1614 \DefineFamilyKey{SVG}{extractheight}{%
1615
      \FamilyKeyStateUnknownValue%
1616
      \svg@ifvalueisrelax{#1}{%
        \renewcommand*\svgx@param@height{\z@}%
1617
        \FamilyKeyStateProcessed%
1618
1619
        \Ifstr{#1}{inherit}{%
1620
          \renewcommand*\svgx@param@height{\svg@param@height}%
1621
          \FamilyKeyStateProcessed%
1622
1623
          \FamilySetLengthMacro{SVG}{extractheight}{\svgx@param@height}{#1}%
1624
          \ifx\FamilyKeyState\FamilyKeyStateProcessed%
1625
            \ifdim\svgx@param@height<\z@\relax%
1626
               \verb|\FamilyKeyStateUnknownValue%||
1627
            \fi%
1628
1629
          \fi%
1630
        }%
1631
1632 }
1633 \newif\if@svgx@param@distort
1634 \DefineFamilyKey{SVG}{extractdistort}[true]{%
1635
      \FamilyKeyStateUnknownValue%
      \svg@ifvalueisrelax{#1}{%
1636
        \@svgx@param@distortfalse%
1637
        \FamilyKeyStateProcessed%
1638
1639
1640
        \Ifstr{#1}{inherit}{%
          \renewcommand*\if@svgx@param@distort{\if@svg@param@distort}%
1641
          \FamilyKeyStateProcessed%
1642
1643
1644
          \FamilySetBool{SVG}{extractdistort}{@svgx@param@distort}{#1}%
1645
        }%
      }%
1646
1647 }
1648 \DefineFamilyKey{SVG}{extractkeepaspectratio}[true]{%
      \FamilySetBool{SVG}{extractkeepaspectratio}{@svg@tempswa}{#1}%
1649
1650
      \ifx\FamilyKeyState\FamilyKeyStateProcessed%
1651
        \if@svg@tempswa%
```

```
1652
                                 \FamilyOptions{SVG}{extractdistort=false}%
                      1653
                      1654
                                 \FamilyOptions{SVG}{extractdistort=true}%
                      1655
                               \fi%
                             \else%
                      1656
                               \FamilyOptions{SVG}{extractdistort=#1}%
                      1657
                      1658
                      1659 }
                      1660 \newcommand*\svgx@param@scale{\svg@param@scale}
                      1661 \DefineFamilyKey{SVG}{extractscale}{%
                             \verb|\FamilyKeyStateUnknownValue%||
                      1662
                             \svg@ifvalueisrelax{#1}{%
                      1663
                               \renewcommand*\svgx@param@scale{1}%
                      1664
                      1665
                               \FamilyKeyStateProcessed%
                      1666
                      1667
                               \Ifstr{#1}{inherit}{%
                      1668
                                 \renewcommand*\svgx@param@scale{\svg@param@scale}%
                      1669
                                 \FamilyKeyStateProcessed%
                      1670
                               }{%
                                 \Ifisdimension{#1\p@}{%}
                      1671
                                    \ifdim\dimexpr#1\p@\relax>\z@\relax%
                      1672
                                      \renewcommand*\svgx@param@scale{#1}%
                      1673
                                      \FamilyKeyStateProcessed%
                      1674
                      1675
                                    fi%
                                 }{}%
                      1676
                               }%
                      1677
                      1678
                             }%
                      1679 }
                      1680 (/extract)
                      The similar hooks for executing code right before or after the graphic extraction.
extractpretex (opt.)
\svgx@param@pretex
                      1681 (*base)
{\tt extractapptex} \ ({\rm opt.})
                      1682 \svg@dummy@key{extractpretex}
\svgx@param@apptex
                      1683 \svg@dummy@key{extractapptex}
{\tt extractpostex} \ ({\rm opt.})
                      1684 \svg@dummy@key{extractpostex}
                      1685 (/base)
                      1686 (*extract)
                      1687 \newcommand*\svgx@param@pretex{\svg@param@pretex}
                      1688 \DefineFamilyKey{SVG}{extractpretex}{%
                      1689
                             \svg@ifvalueisrelax{#1}{%
                      1690
                               \let\svgx@param@pretex\relax%
                      1691
                             }{%
                               \Ifstr{#1}{inherit}{%
                      1692
                                  \renewcommand*\svgx@param@pretex{\svg@param@pretex}%
                      1693
                      1694
                                  \renewcommand*\svgx@param@pretex{#1}%
                      1695
                      1696
                               }%
                      1697
                      1698
                             \FamilyKeyStateProcessed%
                      1699 }
                      1700 \newcommand*\svgx@param@apptex{\svg@param@apptex}
                          \label{lem:lykey} $$ \operatorname{SVG}_{\operatorname{extractapptex}}_{\operatorname{cxt}} $$
                             \svg@ifvalueisrelax{#1}{%
                      1702
                               \let\svgx@param@apptex\relax%
                      1703
                      1704
                      1705
                               \Ifstr{#1}{inherit}{%
                                 \renewcommand*\svgx@param@apptex{\svg@param@apptex}%
                      1706
                      1707
                      1708
                                  \renewcommand*\svgx@param@apptex{#1}%
                      1709
                               }%
                      1710
                            }%
                      1711
                             \FamilyKeyStateProcessed%
                      1712 }
                      1713 \DefineFamilyKey{SVG}{extractpostex}{%
```

```
1714 \svg@deprecated@key[svg-extract]{extractpostex=#1}{extractapptex=#1}% 1715 } 1716 \langle / \text{extract} \rangle
```

C.1.5. Miscellaneous options

clean (opt.)
clear (opt.)
\svgx@clean

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

```
1717 (*base)
1718 \svg@dummy@key[true]{clean}
1719 \svg@dummy@key[true]{clear}
1720 (/base)
1721 \langle *extract \rangle
1722 \newcommand*\svgx@clean{}
1723 \DefineFamilyKey{SVG}{clean}[true]{%
      \FamilySetBool{SVG}{clean}{@svg@tempswa}{#1}%
1725
      \ifx\FamilyKeyState\FamilyKeyStateProcessed%
1726
        \if@svg@tempswa%
1727
          \renewcommand*\svgx@clean{log,aux,dvi,out,ps,eps,pdf,\svgx@latex@ext}%
1728
        \else%
1729
          \renewcommand*\svgx@clean{}%
        \fi%
1730
      \else%
1731
        \renewcommand*\svgx@clean{#1}%
1732
        \FamilyKeyStateProcessed%
1733
1734
      \fi%
1735 }
1736 \DefineFamilyKey{SVG}{clear}{\FamilyOptions{SVG}{clean=#1}}
1737 (/extract)
```

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

```
1738 (*base)
1739 \svg@dummy@key[true]{exclude}
1740 (/base)
1741 (*extract)
1742 \DefineFamilyKey{SVG}{exclude}[true]{%
      \FamilySetBool{SVG}{exclude}{@svg@tempswa}{#1}%
1743
      \ifx\FamilyKeyState\FamilyKeyStateProcessed%
1744
        \if@svg@tempswa%
1745
          \renewcommand*\svg@input[2][]{%
1746
             \if@svgx@run\else%
1747
               \PackageWarning{svg-extract}{%
1748
                 The image '##2' was\MessageBreak%
1749
1750
                 neither extracted nor included%
               }%
1751
            \pi
1752
          }%
1753
        \else%
1754
          \renewcommand*\svg@input{\svg@@input}%
1755
        \fi%
1756
1757
      \fi%
1758 }
1759 (/extract)
```

C.2. User commands

\includesvg The parameters angle and origin are definied as pendants to the keys provided by extract (param.) \includegraphics.

```
extractpreamble (param.)
extractformat (param.)
extractwidth (param.)
extractheight (param.)
extractdistort (param.)
extractscale (param.)
extractangle (param.)
extractpretex (param.)
```

```
1760 (*extract)
1761 \newcommand*\svgx@param@angle{0}
1762 \svg@local@param@def{%
      \DefineFamilyKey[.param]{SVG}{extractangle}{%
1763
        \FamilyKeyStateUnknownValue%
1764
1765
        \svg@ifvalueisrelax{#1}{%
1766
          \renewcommand*\svgx@param@angle{0}%
          \FamilyKeyStateProcessed%
1767
        }{%
1768
          \Ifstr{#1}{inherit}{%
1769
             \renewcommand*\svgx@param@angle{\svg@param@angle}%
1770
             \FamilyKeyStateProcessed%
1771
          }{%
1772
1773
             \Ifisdimension{#1\p0}{%}
1774
               \renewcommand*\svgx@param@angle{#1}%
1775
               \FamilyKeyStateProcessed%
1776
            }{}%
          }%
1777
1778
        }%
      }%
1779
1780 }
1781 (/extract)
Some dummys for package svg.
```

\svghidepreamblestart \svghidepreambleend

```
1782 (*base)
1783 \newcommand*\svghidepreamblestart{%
1784
     \PackageWarning{svg}{%
        The macro '\string\svghidepreamblestart' is only meant\MessageBreak%
1785
        to be used together with package 'svg-extract'.\MessageBreak%
1786
1787
        Nevertheless, nothing will happen%
1788
     }%
1789 }
1790 \newcommand*\svghidepreambleend{%
      \PackageWarning{svg}{%
1791
        The macro '\string\svghidepreambleend' is only meant\MessageBreak%
1792
        to be used together with package 'svg-extract'.\MessageBreak%
1793
1794
        Nevertheless, nothing will happen%
1795
     }%
1796 }
1797 (/base)
```

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

```
1798 (*extract)
1799 \let\svghidepreamblestart\relax
1800 \let\svghidepreambleend\relax
1801 (/extract)
```

C.3. Auxiliary macros

\svg@extract \svgx@stream@in \svgx@read@line \svgx@stream@out \if@svgx@preamble@write

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.

```
1802 (*base)
1803 \newcommand*\svg@extract[1]{}
1804 (/base)
1805 (*extract)
1806 \ifnum\pdf@shellescape=\@ne\relax\else%
      \renewcommand*\svg@extract[1]{%
1807
        \if@svgx@run%
1808
1809
          \begingroup%
```

```
1810
            \edef\svg@tempa{#1}%
1811
            \svg@quotes@remove{\svg@tempa}%
1812
            \PackageWarning{svg-extract}{%
              You didn't enable 'shell escape' (or 'write18')\MessageBreak%
1813
               so it wasn't possible to run the extraction for\MessageBreak%
1814
              file '\svg@tempa'\MessageBreak%
1815
            }%
1816
1817
          \endgroup%
1818
        \fi%
     ጉ%
1819
      \expandafter\endinput%
1820
1821 \fi
```

If --shell-escape is enabled, the command is defined with its intended functionality. Some macros and a input stream as well as a output stream are necessary for this.

```
1822 \newread\svgx@stream@in
1823 \newcommand*\svgx@read@line{}
1824 \newwrite\svgx@stream@out
1825 \newif\if@svgx@preamble@write
1826 \renewcommand*\svg@extract[1]{%
```

If option extract is enabled...

```
1827 \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 LATEX 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.

```
\if@svgx@out@sec%
1828
          \svgx@get@out@sec%
1829
1830
        \stepcounter{svgx@out@count}%
1831
1832
        \begingroup%
          \def\svg@tempa##1.##2\@nil{%
1833
            \IfArgIsEmpty{##2}{\edef\svgx@preamble{##1.\svgx@latex@ext}}{}%
1834
          }%
1835
          \expandafter\svg@tempa\svgx@preamble.\@nil%
1836
          \IfFileExists{\svg@file@path\svgx@preamble}{%
1837
            \@svg@file@foundtrue%
1838
          }{%
1839
1840
            \svg@get@path[]{\svgx@preamble}{\svg@out@path}%
            \def\svg@tempa###1.###2\@ni1{%
1842
               \edef\svgx@preamble{\svg@file@name.###2}%
1843
            }%
1844
            \expandafter\svg@tempa\svgx@preamble\@nil%
          }%
1845
          \edef\svg@tempa{%
1846
            \endgroup%
1847
            \if@svg@file@found%
1848
               \ifx\svg@file@path\@empty%
1849
                 \def\noexpand\svgx@preamble{./\svgx@preamble}%
1850
1851
               \else%
                 \def\noexpand\svgx@preamble{\svg@file@path\svgx@preamble}%
1852
1853
               \fi%
1854
            \fi%
          }%
1855
1856
        \svg@tempa%
        \begingroup%
1857
          \endlinechar=\m@ne%
1858
          \IfFileExists{\svgx@preamble}{%
1859
1860
            \PackageInfo{svg-extract}{%
```

File II: svg.dtx Date: 2020/01/13 v2.02e

```
The preamble file '\svgx@preamble'\MessageBreak% is used for the generation of the auxiliary file\MessageBreak% '\svgx@out@name.\svgx@latex@ext'% }%
```

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

The given preamble file is read line by line and written to the separate auxiliary IATEX file \svgxQoutQname.\svgxQlatexQext via the output stream.

```
1871 \@whilesw\if@svg@tempswa\fi{%

1872 \immediate\read\svgx@stream@in to\svgx@read@line%

1873 \ifx\svgx@read@line\@empty%

1874 \ifeof\svgx@stream@in\@svg@tempswafalse\fi%

1875 \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 \svgx@read@preamble@from are toggling the switch \if@svgx@preamble@write

```
1876 \svgx@read@preamble@till{\svghidepreamblestart}{}%
1877 \svgx@read@preamble@from{\svghidepreambleend}{}%
```

If the desired end of the preamble (\svgx@endpreamble) was found, the readout is terminated by switching \if@svg@tempswa to false.

```
1878 \svgx@read@preamble@till{\svgx@endpreamble}{\@svg@tempswafalse}%
1879 \if@svgx@preamble@write%
```

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.

```
1880 \if@svgx@classfound\else%
1881 \expandafter\svgx@documentclass%
1882 \svgx@read@line\documentclass\documentclass\@nil%
1883 \fi%
```

Writing out the—maybe manipulated—read in line.

```
\ifx\svgx@read@line\@empty\else%
1884
                     \immediate\write\svgx@stream@out{%
1885
1886
                        \unexpanded\expandafter{\svgx@read@line}%
1887
                     }%
1888
                   \fi%
                 \fi%
               \fi%
1890
             }%
1891
1892
             \immediate\closein\svgx@stream@in%
             \immediate\closeout\svgx@stream@out%
1893
             \catcode'\#=6\relax%
1894
```

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.

```
1895 \immediate\openin\svgx@stream@in=\svgx@out@name.\svgx@latex@ext%
1896 \def\svg@tempa{}%
1897 \loop\unless\ifeof\svgx@stream@in%
1898 \readline\svgx@stream@in to\svgx@read@line%
1899 \ifx\svgx@read@line\@empty\else%
```

```
1900
                 \edef\svg@tempa{%
1901
                   \unexpanded\expandafter{\svg@tempa}%
1902
                   \unexpanded\expandafter{\svgx@read@line}^^J%
                 }%
1903
               \fi%
1904
1905
             \repeat%
1906
             \immediate\closein\svgx@stream@in%
1907
          }{%
```

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

```
\svg@quotes@remove{\svgx@preamble}%
1908
1909
             \ifx\svgx@preamble\@empty\else%
1910
               \PackageWarning{svg-extract}{%
                 The preamble file '\svgx@preamble'\MessageBreak%
1911
                 does not exist%
1912
1913
              }%
            \pi
1914
             \def\svg@tempa{}%
1915
1916
```

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.

```
1917
          \immediate\openout\svgx@stream@out=\svgx@out@name.\svgx@latex@ext%
1918
          \immediate\write\svgx@stream@out{%
1919
            \@percentchar\@percentchar\space This file was generated by package
            'svg-extract'^^J%
1920
            \Opercentchar\Opercentchar\space from source '\jobname'^^J%
1921
            \@percentchar\@percentchar\space It's intended to be compiled with
1922
1923
            '\svgx@latex@exe\ifx\svgx@latex@opt\@empty\else\space\svgx@latex@opt\fi'
          ጉ%
1924
```

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.

```
\immediate\write\svgx@stream@out{%
1925
1926
            \string\AtBeginDocument{\@percentchar^^J%
1927
              \space\space\string\svgxsetpapersize\@percentchar^^J%
1928
              \ifxetex\else\ifpdf\else%
1929
                \space\space\string\AtBeginDvi{\string\special{%
1930
                     papersize=\string\the\string\paperwidth,%
1931
                       \string\the\string\paperheight%
                }}\@percentchar^^J%
1932
              fi\fi
1933
            }^^J%
1934
            \string\PassOptionsToPackage{hidelinks}{hyperref}%
1935
1936
```

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

```
1937 \if@svgx@classfound\else%
1938 \immediate\write\svgx@stream@out{\string\documentclass{article}}%
1939 \fi%
```

And now the stored preamble.

```
1940 \ifx\svg@tempa\@empty\else%
1941 \immediate\write\svgx@stream@out{\unexpanded\expandafter{\svg@tempa}}%
1942 \fi%
```

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

```
1943 \immediate\write\svgx@stream@out{\string\usepackage{svg-extract}}%
```

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

```
\def\svg@tempa##1{%
1945
                                    \immediate\write\svgx@stream@out{\string\svgsetup{##1}}%
1946
                              \if@svg@ink@latex\else%
1947
                                     \svg@tempa{inkscapelatex=false}%
1948
                              \fi%
1949
1950
                              \ifdim\svgx@param@width>\z@\relax%
                                    \svg@tempa{width=\svgx@param@width}%
1951
                              \fi%
1952
                              \ifdim\svgx@param@height>\z@\relax%
1953
                                    \svg@tempa{height=\svgx@param@height}%
1954
                              \fi%
1955
                               \if@svgx@param@distort%
1956
                                     \svg@tempa{distort=true}%
1957
1958
                              \fi%
                              \ifdim\dimexpr\svgx@param@scale\p@\relax=\p@\relax\else%
1959
1960
                                     \svg@tempa{scale=\svgx@param@scale}%
                              \fi%
1961
                              \def\svg@tempb{\svg@param@pretex}%
1962
                              \ifx\svgx@param@pretex\svg@tempb\relax%
1963
                                    \let\svgx@param@pretex\svg@param@pretex%
1964
1965
                              \fi%
1966
                              \ifx\svgx@param@pretex\relax\else%
                                    \verb|\svg@tempa{pretex=\unexpanded\expandafter{\svgx@param@pretex}}||% \end{| constraints of the property of th
1967
1968
                              \def\svg@tempb{\svg@param@apptex}%
1969
1970
                              \ifx\svgx@param@apptex\svg@tempb\relax%
1971
                                    \let\svgx@param@apptex\svg@param@apptex%
1972
                              \fi%
                              \ifx\svgx@param@apptex\relax\else%
1973
                                    \svg@tempa{apptex=\unexpanded\expandafter{\svgx@param@apptex}}%
1974
1975
                              \fi%
```

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

```
\let\svg@tempa\@empty%
1976
1977
         \if@svg@ink@latex%
1978
           \Ifstr{\svg@ink@format}{pdf}{%
             1979
1980
               \edef\svg@tempa{lastpage=\the\value{svg@param@lastpage}}%
             \else%
1981
               \ifnum\value{svg@param@lastpage}=\z@\relax%
1982
                 \def\svg@tempa{lastpage=true}%
1983
1984
               \else%
                 \def\svg@tempa{lastpage=false}%
1985
               \fi%
1986
1987
             \fi%
           }{}%
1988
         \fi%
1989
```

The rotation angle, if given.

```
1990 \ifdim\dimexpr\svgx@param@angle\p@\relax=\z@\relax\else%

1991 \edef\svg@tempa{%

1992 angle=\svgx@param@angle\ifx\svg@tempa\@empty\else,\svg@tempa\fi%

1993 }%

1994 \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.

```
1995 \ifx\svg@tempa\@empty%
1996 \def\svg@tempa{\string\svgxsetbox{#1}}%
1997 \else%
1998 \edef\svg@tempa{\noexpand\string\noexpand\svgxsetbox[\svg@tempa]{#1}}%
1999 \fi%
2000 \immediate\write\svgx@stream@out{\svg@tempa}%
```

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

```
\if@svg@ink@latex%
2001
            \IfFileExists{xr.sty}{%
2002
2003
               \immediate\write\svgx@stream@out{%
2004
                 \string\usepackage{xr}^^J%
2005
                 \string\externaldocument{\jobname}^^J%
2006
               }%
2007
            }{}%
2008
          \fi%
2009
          \immediate\write\svgx@stream@out{%
            \string\begin{document}^^J%
2010
            \t \t ing\pagestyle{empty}^^J\%
2011
            \string\svgxoutputbox\@percentchar^^J%
2012
2013
             \string\end{document}%
2014
2015
          \immediate\closeout\svgx@stream@out%
2016
        \endgroup%
```

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

```
2017 \Ifstr{\svgx@format\svgx@cnv@format}{}{%
2018 \PackageWarning{svg-extract}{%
2019 Both keys 'extractformat' and 'convertformat' are\MessageBreak%
2020 empty, so nothing to do so far%
2021 }%
2022 }{%
```

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.

```
2023 \svg@quotes@remove{\svgx@out@path}%
2024 \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.

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

```
2029 \edef\svg@tempb{%
2030 \noexpand\PackageInfo{svg-extract}{%
2031 Running LaTeX (\svgx@latex@exe) for graphic extraction%
2032 \ifx\svgx@latex@opt\@empty\else%
2033 \MessageBreak with added options '\svgx@latex@opt'%
2034 \fi%
2035 }%
```

```
2036
          }%
2037
          \expandafter\AfterReadingMainAux\expandafter{\svg@tempb}%
2038
          \edef\svg@tempb{%
            \noexpand\ShellEscape{%
2039
              \svgx@latex@exe\space\svgx@latex@opt\space%
2040
2041
              "\svgx@out@name.\svgx@latex@ext"%
2042
            }%
2043
          }%
          \loop\ifnum\value{svgx@runs}>\z@\relax%
2044
            \expandafter\AfterReadingMainAux\expandafter{\svg@tempb}%
2045
2046
            \advance\c@svgx@runs\m@ne%
2047
          \repeat%
```

All files requested with option extractformat are created with internal conversion tools supplied by most \LaTeX 2ε distributions if necessary.

```
\def\svg@tempa##1##2##3{%
2048
            \edef\svg@tempb{%
2049
2050
              \noexpand\ShellEscape{%
                 \Onameuse{svgx0##10exe}\space\Onameuse{svgx0##10opt}\space%
2051
2052
                 "\svgx@out@name.##2"%
              }%
2053
2054
            }%
2055
            \AfterReadingMainAux{\PackageInfo{svg-extract}{Running ##1}}%
2056
            \expandafter\AfterReadingMainAux\expandafter{\svg@tempb}%
2057
2058
          \@svg@tempswafalse%
          \ifxetex\else\ifpdf\else%
2059
            \@svg@tempswatrue%
2060
2061
          \fi\fi%
2062
          \if@svg@tempswa%
2063
            \svg@tempa{dvips}{dvi}{ps}%
2064
            \svgx@ifinlist{eps}{\svgx@format}{\svg@tempa{pstoeps}{ps}{eps}}{}%
2065
            \svgx@ifinlist{pdf}{\svgx@format}{\svg@tempa{pstopdf}{ps}{pdf}}{}%
2066
          \else%
2067
            \svgx@ifinlist{eps}{\svgx@format}{\svg@tempa{pdftoeps}{pdf}{eps}}{}%
2068
            \svgx@ifinlist{ps}{\svgx@format}{\svg@tempa{pdftops}{pdf}{ps}}{}%
2069
```

Now the desired conversion tool is invoked if requested.

```
2070 \if@svgx@cnv@run%
```

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

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

```
2076 \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{%
2077
2078
              \ifx\svg@tempa\@empty\else%
2079
                \expandafter\svgx@ifinlist\expandafter{\svg@tempa}{\svgx@format}{%
2080
                  \PackageWarning{svg-extract}{%
2081
                    File type '\svg@tempa' was specified for option\MessageBreak%
2082
                    'extractformat' (\svgx@format) as well as for \MessageBreak%
2083
                    option 'convertformat' (\svgx@cnv@format) so the\MessageBreak%
                    conversion won't be done%
2084
                  }%
2085
```

```
2086
                }{%
2087
                   \edef\svg@tempb{%
2088
                     \noexpand\PackageInfo{svg-extract}{%
                       Converting '\svgx@out@name.\svgx@cnv@informat'\MessageBreak%
2089
                       to '\svgx@out@name.\svg@tempa'%
2090
2091
                     }%
2092
                   ጉ%
                   \expandafter\AfterReadingMainAux\expandafter{\svg@tempb}%
2093
                   \edef\svg@tempb{%
2094
                     \noexpand\ShellEscape{%
2095
                       \svgx@cnv@cmd{\svgx@out@name}{\svgx@cnv@informat}{\svg@tempa}%
2096
                     }%
2097
2098
2099
                   \expandafter\AfterReadingMainAux\expandafter{\svg@tempb}%
2100
                }%
2101
               \fi%
            }%
2102
2103
          \fi%
```

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

```
2104
          \edef\svg@tempa{\svgx@format\if@svgx@cnv@run,\svgx@cnv@format\fi}%
2105
          \@for\svg@tempb:=\svg@tempa\do{%
2106
            \ifx\svg@tempb\@empty\else%
              \edef\svg@tempb{%
2107
2108
                 \noexpand\svgx@move{\svgx@out@name}{\svg@tempb}{\svgx@out@path}%
2109
              \expandafter\AfterReadingMainAux\expandafter{\svg@tempb}%
2110
2111
            \fi%
          }%
2112
```

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

```
\@for\svg@tempa:=\svgx@clean\do{%
2113
2114
             \expandafter\svgx@ifinlist\expandafter{\svg@tempa}{\svg@tempb}{}{%
2115
               \edef\svg@tempb{%
2116
                 \noexpand\IfFileExists{"\svgx@out@name".\svg@tempa}{%
2117
                   \noexpand\svg@shell@rm{\svgx@out@name.\svg@tempa}%
2118
                 }{}%
               }%
2119
               \expandafter\AtEndDocument\expandafter{%
2120
                 \expandafter\AfterReadingMainAux\expandafter{\svg@tempb}%
2121
              ጉ%
2122
            }%
2123
          }%
2124
2125
        }%
2126
      \fi%
2127 }
2128 (/extract)
```

\svgx@get@out@sec \svgx@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.

```
2129 \newcommand*\svgx@out@sec{unknown}
2130 \newcommand*\svgx@get@out@sec{%
2131
     \begingroup%
       \def\svg@tempa{}%
2132
2133
       \@for\svg@tempb:={%
2134
        2135
      }\do{%
2136
        \ifx\svg@tempb\@empty\else%
2137
          \scr@ifundefinedorrelax{the\svg@tempb}{}{%
2138
            \ifnum\value{\svg@tempb}>\z@\relax%
2139
              \edef\svg@tempa{\svg@tempb}%
2140
            \fi%
```

```
2141
             }%
2142
           \fi%
2143
        }%
         \edef\svg@tempb{%
2144
           \endgroup%
2145
2146
           \ifx\svg@tempa\@empty\else%
2147
             \def\noexpand\svgx@out@sec{\csname the\svg@tempa\endcsname}%
2148
           \fi%
        ጉ%
2149
2150
      \svg@tempb%
2151 }
```

\svgx@documentclass \if@svgx@classfound This delimited macro is used to find a 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.

\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.

```
2157 \newcommand*\svgx@read@preamble@till[2]{%
2158 \svgx@read@preamble@skip#1\@nil{till}{#2}%
2159 }
2160 \newcommand*\svgx@read@preamble@from[2]{%
2161 \svgx@read@preamble@skip#1\@nil{from}{#2}%
2162 }
```

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

```
2163 \newcommand*\svgx@read@preamble@skip{}
2164 \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.

```
2165 \def\svg@tempa##1{%
2166 \def\svg@tempa###1###2##1###3\@nil{%
2167 \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.

```
2168 \Ifstr{#2}{till}{%
2169 \IfArgIsEmpty{####1}{}{%
2170 \immediate\write\svgx@stream@out{####1}%
2171 }%
2172 \@svgx@preamble@writefalse%
2173 }{%
```

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}{%
2174
                 \IfArgIsEmpty{####2}{%
2175
                   \def\svgx@read@line{}%
2176
                 }{%
2177
                   \def\svgx@read@line{####2}%
2178
2179
                 }%
2180
                 \@svgx@preamble@writetrue%
2181
               }{}%
2182
             }%
```

Additional stuff which should be done.

```
2183 #3%
2184 }%
2185 }%
2186 }%
```

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

```
2187 \edef\svg@tempb{\expandafter\detokenize\expandafter{#1}}%
2188 \expandafter\svg@tempa\expandafter{\svg@tempb}%
```

Calling the created macro.

```
2189 \edef\svg@tempb{%
2190 \expandafter\detokenize\expandafter{\svgx@read@line}\svg@tempb\svg@tempb\%
2191 }%
2192 \expandafter\svg@tempa\svg@tempb\@nil%
2193 }
```

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

```
2194 \newcommand*\svgx@cnv@informat{}
2195 \newcommand*\svgx@cnv@get@informat[1]{%
2196
      \begingroup%
        \def\svg@tempa##1,##2\@nil{%
2197
          \def\svg@tempa{##1}%
2198
2199
        \svg@tempa#1,\@nil%
2200
2201
        \edef\svg@tempa{%
2202
          \endgroup%
          \def\noexpand\svgx@cnv@informat{\svg@tempa}%
2203
2204
        ጉ%
2205
      \svg@tempa%
```

If the first argument (\svgx@format) was empty, \svgx@cnv@informat is set to the a file type, which is generated anyway.

```
2206 \ifx\svgx@cnv@informat\@empty%
2207 \renewcommand*\svgx@cnv@informat{pdf}%
2208 \ifxetex\else\ifpdf\else%
2209 \renewcommand*\svgx@cnv@informat{ps}%
2210 \fi\fi\%
2211 \fi\%
2212 }
```

\svgx@magick@cmd \svgx@gs@cmd

Depending on option convert, one of these two macros is actually used by \svgx@cnv@cmd. For invoking the conversion process, the required platform-dependent executable is set, if nothing was set by a package option.

```
2213 \ifx\svgx@magick@exe\@empty
2214
     \ifwindows
2215
        \renewcommand*\svgx@magick@exe{magick}
2216
      \else
        \renewcommand*\svgx@magick@exe{convert}
2217
2218 \fi
2219 \fi
2220 \newcommand*\svgx@magick@cmd[3]{%
2221
      \svgx@magick@exe\space%
      \svgx@useformatkey{svgx@cnv@dpi}{#3}{-density }%
2223
      \svgx@useformatkey{svgx@magick@set}{#3}{}%
2224
      "#1.#2"\space%
2225
      \svgx@useformatkey{svgx@magick@opr}{#3}{}%
2226
      "#1.#3"%
2227 }
```

```
2228 \ifx\svgx@gs@exe\@empty
             2229
                    \ifwindows
             2230
                      \renewcommand*\svgx@gs@exe{gswin64c}
             2231
                      \renewcommand*\svgx@gs@exe{gs}
             2232
             2233
                   \fi
             2234 \fi
             2235 \newcommand*\svgx@gs@cmd[3]{%
                   \verb|\svgx@gs@exe\space-dSAFER -dBATCH -dNOPAUSE\space||
             2236
                    \label{lem:condition} $$\sup_{0\to\infty}\sup_{0\to\infty}\sup_{0\to\infty}\mathbb{F}_{3}{-\sup_{0\to\infty}\mathbb{F}_{3}}
             2237
                    \svgx@useformatkey{svgx@cnv@dpi}{#3}{-r}%
             2238
                    \svgx@useformatkey{svgx@gs@opt}{#3}{}%
             2239
             2240
                    -sOutputFile="#1.#3"\space"#1.#2"%
             2241 }
             If the file doesn't exist
\svgx@move
             2242 \newcommand*\svgx@move[3] {%
             2243
                   \begingroup%
                      \IfFileExists{"#1".#2}{%}
             2244
                        \svg@shell@move{#1.#2}{#3#1.#2}%
             2245
             2246
                      }{%
             2247
                        \edef\svg@tempa{#2}%
                        \@svg@tempswafalse%
             2248
             2249
                        \expandafter\svgx@ifinlist\expandafter{\svg@tempa}{\svgx@cnv@format}{%
             2250
                           \@svg@tempswatrue%
             2251
                          \def\svg@tempb{conversion}%
             2252
                        }{%
             2253
                           \expandafter\svgx@ifinlist\expandafter{\svg@tempa}{pdf,ps,eps}{%
             2254
                             \@svg@tempswatrue%
                             \def\svg@tempb{extraction}%
             2255
                          }{}%
             2256
                        }%
             2257
                        \if@svg@tempswa%
             2258
             2259
                          \edef\svg@tempb{%
             2260
                            The graphic file \svg@tempb\space failed\MessageBreak%
                             for '#1.#2'\MessageBreak%
             2261
             2262
                             Troubleshooting: Please check the log file how\MessageBreak%
             2263
                             the invocation of the extraction took place and\MessageBreak%
             2264
                             try to execute it yourself in the terminal%
                          }%
             2265
             2266
                        \else%
                          \def\svg@tempb{%
             2267
                            The extraction to format '#2' failed\MessageBreak%
             2268
             2269
                             for '#1.#2'\MessageBreak%
             2270
                             Only file types 'pdf,ps,eps' are supported for\MessageBreak%
                             key 'exportformat'%
             2271
             2272
                          }%
             2273
                        \fi%
             2274
                        \PackageWarning{svg-extract}{\svg@tempb}%
                      }%
             2275
                    \endgroup%
             2276
             2277 }
```

\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.

```
2278 \newcommand*\svgx@ifinlist[2]{%
2279
      \begingroup%
        \def\svg@tempa##1,#1,##2\@nil{%
2280
2281
          \IfArgIsEmpty{##2}{%
2282
             \aftergroup\@secondoftwo%
2283
2284
             \aftergroup\@firstoftwo\%
          }%
2285
```

```
2286 }%
2287 \expandafter\svg@tempa\expandafter,#2,#1,\@nil%
2288 \endgroup%
2289 }
```

\svgx@onlywindows Do only some stuff, if Windows was detected.

```
2290 \newcommand*\svgx@onlywindows[1]{}
2291 \AfterPackage*{ifplatform}{\renewcommand*\svgx@onlywindows[1]{\ifwindows#1\fi}}
```

\svgx@ifkeyandval

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

```
2292 \newcommand*\svgx@ifkeyandval[3]{%
2293 \def\svg@tempa#1=##2=##3\@nil{\Ifstr{##3}{=}{#2}{#3}}%
2294 \svg@tempa#1==\@nil%
2295 }
```

\svgx@cnv@get@dpi

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.

```
2296 \newcommand*\svgx@cnv@get@dpi[1]{%
2297 \begingroup%
2298 \def\svg@tempa##1dpi##2x##3dpi##4\@nil{%
2299 \edef\svg@tempa{##1}%
```

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

```
2300 \@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.

```
2301
          \Ifnumber{##1}{%
2302
             \IfArgIsEmpty{##3}{\@svg@tempswatrue}{%
2303
               \Ifnumber{##3}{\edef\svg@tempa{##1x##3}}{}%
2304
            }%
2305
          }{}%
2306
          \if@svg@tempswa\else%
            \expandafter\svg@tempb\svg@tempa xx\@nil%
2307
2308
          \fi%
2309
        ጉ%
```

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

```
\def\svg@tempb##1x##2x##3\@ni1{%
2310
2311
         \frak{1}{x}{x}{x}
2312
           \@svg@tempswatrue%
2313
           \IfArgIsEmpty{##1}{\@svg@tempswafalse}{%
2314
             \Ifnumber{##1}{}{\@svg@tempswafalse}%
2315
           }%
2316
           \IfArgIsEmpty{##2}{\@svg@tempswafalse}{%
             2317
           ጉ%
2318
           \if@svg@tempswa%
2319
             \edef\svg@tempa{##1x##2}%
2320
2321
           \fi%
2322
         }{}%
2323
       }%
        \IfArgIsEmpty{#1}{%
2324
2325
         \let\svg@tempa\@empty%
```

```
2326
        }{%
2327
          \lowercase{\svg@tempa#1dpi#1xdpi\@nil}%
2328
          \if@svg@tempswa\else%
             \let\svg@tempa\relax%
2329
          \fi%
2330
2331
        }%
2332
        \edef\svg@tempb{%
2333
          \endgroup%
          \ifx\svg@tempa\relax%
2334
             \let\noexpand\svg@tempa\noexpand\relax%
2335
2336
             \def\noexpand\svg@tempa{\svg@tempa}%
2337
2338
           \fi%
2339
        }%
2340
      \svg@tempb%
2341 }
```

\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.

```
2342 \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.

```
2343 \svgx@ifkeyandval{#1}{%

2344 \svg@ifvalueisrelax{##2}{%

2345 \expandafter\let\csname #2@##1\endcsname\relax%

2346 }{%

2347 \@namedef{#2@##1}{##2}%

2348 }%
```

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

```
2349 }{%
2350 \svg@ifvalueisrelax{##1}{%
2351 \expandafter\let\csname #2\endcsname\relax%
2352 }{%
2353 \@namedef{#2}{##1}%
2354 }%
2355 }%
2356 }
```

The command \svgx@useformatkey checks, if a format specific key was definded 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.

```
2357 \newcommand*\svgx@useformatkey[3]{%
                                       \scr@ifundefinedorrelax{#1@#2}{%
2358
                                                       \scr@ifundefinedorrelax{#1}{}{%
2359
                                                                  \verb|\expandafter if x \csname #1 \ends name \end{|} empty \else % $$ $ (empty \else % \end{|} empty \end{|} % $$ $ (empty \else % \end{|} empty \end{|} empty \end{|} % $$ $ (empty \else % \end{|} empty \end{|} empty \end{|} % $$ $ (empty \else % \end{|} empty \end{|} empty \end{|} % $$ $ (empty \else % \end{|} empty \end{|} empty \end{|} % $$ $ (empty \else % \end{|} empty \end{|} empty \end{|} % $$ $ (empty \else % \end{|} empty \end{|} empty \end{|} % $$ $ (empty \else % \end{|} empty \end{|} empty \end{|} % $$ $ (empty \else % \end{|} empty \end{|} empty \end{|} % $$ $ (empty \else % \end{|} empty \end{|} empty \end{|} % $$ $ (empty \else % \end{|} empty \end{|} empty \end{|} % $$ $ (empty \else % \end{|} empty \end{|} empty \end{|} % $$ $ (empty \else % \end{|} empty \end{|} empty \end{|} % $$ $ (empty \else % \end{|} empty \end{|} empty \end{|} empty \end{|} % $$ $ (empty \end{|} empty \end{|} 
2360
2361
                                                                                #3\0 nameuse{#1}\space{
2362
                                                                  \fi%
2363
                                                       \scr@ifundefinedorrelax{#1@#2+}{}{%
2364
                                                                    \expandafter\ifx\csname #10#2+\endcsname\@empty\else%
2365
2366
                                                                                #3\0 = {#10#2+}\space
2367
                                                                  \fi%
                                                    }%
2368
                                     }{%
2369
```

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

```
2370 \expandafter\ifx\csname #10#2\endcsname\@empty\else%

2371 #3\@nameuse{#10#2}\space%

2372 \fi%

2373 }%

2374 }
```

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

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 grahic 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.

```
2375 \newif\if@svgx@standalone
2376 \newcommand*\svgxsetbox[2][]{%
2377
      \@svgx@standalonetrue%
2378
      \svgx@setbox{#1}{#2}%
2379
      \scr@ifundefinedorrelax{AtEndPreamble}{%
        \let\svg@tempa\@firstofone%
2380
      }{%
2381
2382
        \def\svg@tempa{\AtEndPreamble}%
2383
      }%
      \svg@tempa{\AtBeginDocument{\svgx@setbox{#1}{#2}}}%
2384
2385 }
2386 \newcommand*\svgx@setbox[2]{%
      \sbox\svg@box{\svg@@input[{#1},draft=false]{#2}}%
2387
      \svgxsetpapersize%
2388
2389 }
```

\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.

```
2390 \newcommand*\svgxsetpapersize{%
2391 \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.

```
2392
      \scr@ifundefinedorrelax{stockwidth}{}{%
2393
        \setlength\stockwidth{\paperwidth}%
2394
      }%
      \scr@ifundefinedorrelax{mediawidth}{}{%
2395
2396
        \setlength\mediawidth{\paperwidth}%
      }%
2397
2398
      \setlength\textwidth{\paperwidth}%
2399
      \setlength\paperheight{\the\dimexpr\ht\svg@box+\dp\svg@box\relax}%
2400
      \scr@ifundefinedorrelax{stockheight}{}{%
2401
        \setlength\stockheight{\paperheight}%
2402
2403
      \scr@ifundefinedorrelax{mediaheight}{}{%
2404
        \setlength\mediaheight{\paperheight}%
2405
2406
      \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.

```
2407
                       \hoffset=-1in%
                 2408
                       \oddsidemargin=\z0%
                       \evensidemargin=\z0%
                 2409
                 2410
                       \voffset=-1in%
                       \topmargin=\z0%
                 2411
                 2412
                       \headheight=\z0%
                 2413
                       \headsep=\z0%
                 2414
                       \topskip=\z0%
                 2415
                       \footskip=\z0%
                 2416
                       \marginparsep=\z0%
                 2417
                       \marginparwidth=\z0%
                 2418
                       \marginparpush=\z0%
                 2419 }
                 2420 \@onlypreamble\svgxsetpapersize
                 With \svgxoutputbox the created box is displayed.
\svgxoutputbox
\if@svgx@beamer
                 2421 \newif\if@svgx@beamer
                 2422 \@ifclassloaded{beamer}{\@svgx@beamertrue}{}%
                 2423 \newcommand*\svgxoutputbox{%
                 2424
                      \begingroup%
                 2425
                         \setlength\parindent{\z0}%
                 2426
                          \setlength\parskip{\z0}%
                          \setlength\parfillskip{\z0}%
                 2427
                         \if@svgx@beamer%
                 2428
                            \setbeamertemplate{navigation symbols}{}%
                 2429
                            \begin{frame}[plain]%
                 2430
                            \usebox\svg@box%
                 2431
                            \end{frame}%
                 2432
                         \else%
                 2433
                 2434
                            \usebox\svg@box%
                 2435
                          \fi%
                 2436
                          \endgraf%
                 2437
                       \endgroup%
                 2438 }
```

D. Processing Options

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

```
2439 (*base)
2440 \FamilyExecuteOptions{SVG}{%
2441 inkscape=true,inkscapepath=basesubdir,
2442 inkscapelatex=true,inkscapearea=drawing,distort=false,%
2443 usexcolor=true,usetransparent=true%
2444 }
2445 (/base)
2446 (*extract)
2447 \FamilyExecuteOptions{SVG}{%
2448 extract=true,extractpath=basesubdir,%
2449 extractruns=2,extractname=namenumbered,extractdistort=false,%
2450 convert=magick,convert=false,%
2451 gsdevice={png=png16m},gsdevice={jpeg=jpeg},gsdevice={jpg=jpeg},%
2452
      gsdevice={tif=tiff48nc},gsdevice={tiff=tiff48nc},%
2453
      gsdevice={eps=eps2write},gsdevice={ps=ps2write}%
2454 }
2455 (/extract)
2456 \FamilyProcessOptions{SVG}
```

E. Macros for file access

Finally, platform dependend macros for creating directories as well as moving and deleting files are provided, if --shell-escape is enabled. Only then package **ifplatform** is only used in order to do not raise a warning.

```
2457 \ifnum\pdf@shellescape=\@ne\relax\else%
                    2458
                          \expandafter\endinput%
                    2459 \fi
                    2460 \RequirePackage{ifplatform}[2010/10/22]
 \svg@shell@mkdir
                    The platform dependent commands for file access.
\svg@shell@@mkdir
                    2461 \ifwindows
    \svg@shell@mv
                          \newcommand*\svg@shell@@mkdir[1]{if not exist "#1" mkdir "#1"}
                    2462
   \svg@shell@@mv
                    2463
                          \newcommand*\svg@shell@@mv{move}
    \svg@shell@rm
                          \newcommand*\svg@shell@@rm{del}
                    2464
   \svg@shell@@rm
                    2465 \ensuremath{\setminus} \texttt{else}
                    2466
                          \newcommand*\svg@shell@@mkdir[1]{mkdir -p "#1"}
                    2467
                          \newcommand*\svg@shell@@mv{mv}
                          \newcommand*\svg@shell@@rm{rm}
                    2468
                    2469 \fi
                    A directory should only be created, if it isn't the current working directory.
                    2470 \newcommand*\svg@shell@mkdir[1]{%
                          \begingroup%
                    2472
                             \svg@quotes@remove[{#1}]{\svg@tempa}%
                    2473
                             \@svg@tempswatrue%
                            \Ifstr{\svg@tempa}{}{\@svg@tempswafalse}{%
                    2474
                            \Ifstr{\svg@tempa}{./}{\@svg@tempswafalse}{%
                    2475
                    2476
                            }}%
                            \if@svg@tempswa%
                    2477
                              \ShellEscape{\svg@shell@@mkdir{\svg@tempa}}%
                    2478
                    2479
                    2480
                          \endgroup%
                    2481 }
                    Commands for moving and deleting files.
                    2482 \newcommand*\svg@shell@move[2]{%
                          \ShellEscape{\svg@shell@@mv\space"#1"\space"#2"}%
                    2483
                    2484 }
                    2485 \newcommand*\svg@shell@rm[1]{%
                          \ShellEscape{\svg@shell@@rm\space"#1"}%
                    2487 }
                    At the very end, the catcodes are restored.
```

File II: svg.dtx Date: 2020/01/13 v2.02e

2488 \svg@catcodecodes@restore

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@cnv@run <u>1344</u>
$\mathtt{apptex} \; (\mathtt{opt.}) \; \ldots \ldots \qquad \qquad 5, \underline{309}$	\if@svgx@out@sec <u>1516</u>
	\if@svgx@preamble@write <u>1802</u>
С	\if@svgx@run <u>1140</u>
$\mathtt{clean}\;(\mathtt{opt.})\;\ldots\ldots\;$	\if@svgx@standalone 2375
clear (opt.) <u>1717</u>	\includeinkscape $\dots 6-7, \underline{764}, \underline{1760}$
convert (opt.)	$\texttt{angle} \; (\texttt{param.}) \; \ldots \ldots \; \qquad \qquad$
convertdensity (opt.) 1447	$\mathtt{apptex} \; (\mathtt{param.}) \dots \dots 6, \underline{805}$
convertdpi (opt.) $\dots \dots \dots$	clean (param.)
convertformat (opt.)	convert (param.) $\dots \dots 7, \underline{1760}$
counters:	$\texttt{convertdpi} \; (\text{param.}) \; \dots \qquad 7, \; \underline{1760}$
svg@param@lastpage	convertformat (param.) $\dots $ $7, 1760$
svgx@out@count <u>1516</u>	distort (param.) 6, <u>805</u>
svgx@runs <u>1240</u>	draft (param.)
D	exclude (param.)
distort (opt.)	extract (param.) 7, <u>1760</u>
draft (opt.)	extractangle (param.) $\cdots 7, \underline{1760}$
dvipsopt (opt.)	extractapptex (param.) $$
dvipbopt (opt.)	extractdistort (param.) \cdots $7, \underline{1760}$
E	extractformat (param.) $$
end (opt.)	extractheight (param.) $$
eps (opt.) <u>1172</u>	extractpreamble (param.) $$
exclude (opt.) 9, <u>1738</u>	extractpretex (param.) 7, <u>1760</u>
ext (opt.) <u>214</u>	extractruns (param.)
extension (opt.) $\underline{214}$	extractwidth (param.)
extract (opt.)	gsdevice (param.) 7, <u>1760</u>
extractapptex (opt.)	gsopt (param.)
extractdistort (opt.) $8, 1585$	height (param.) 6 , 805
extractformat (opt.) $8, \underline{1172}$	inkscapeformat (param.) $6, 805$
extractheight (opt.)	inkscapelatex (param.) 6 , 805
extractkeepaspectratio (opt.)	lastpage (param.)
extractpath (opt.)	latexopt (param.) $\dots \dots \dots$
extractpostex (opt.)	magickoperator (param.) $\cdots 7, 1760$
extractpreamble (opt.) $\dots \dots 8, \overline{1213}$	magicksetting (param.) $7, \underline{1760}$
extractpreambleend (opt.)	origin (param.) 6, <u>805</u>
extractpretex (opt.)	pretex (param.)
extractruns (opt.)	scale (param.) 6, 805 width (param.) 6, 805
extractscale (opt.)	\includesvg
extractwidth (opt.)	angle (param.)
G	apptex (param.)
gsdevice (opt.)	clean (param.) 7, <u>1760</u>
gsexe (opt.)	convert (param.) 7, <u>1760</u>
gsopt (opt.)	convertdpi (param.)
. ,	convertformat (param.) $\dots 7, \underline{1760}$
Н	distort (param.) 6, <u>716</u>
$\texttt{height} \; (\text{opt.}) \; \dots \dots \dots \dots \qquad 5, \underline{252}$	draft (param.)
	exclude (param.) 7, <u>1760</u>
\	extract (param.)
\if@svg@draft	extractangle (param.)
\if@svg@file@found	extractapptex (param.)
\if@svg@param@distort	extractformat (param.) 7, <u>1760</u>
\if@svg@quotes@found	extractheight (param.) 7, <u>1760</u>
\if@svg@tempswa 12	extractpreamble (param.) $7, \frac{1760}{1}$
\if@svg@use@transparent 30	extractpretex (param.) 7, 1760
\if@svg@use@xcolor 30	extractruns (param.) $\dots 7, \overline{1760}$
$\verb \if@svgx@beamer \underline{2421}$	extractscale (param.)
$\verb \if@svgx@classfound \underline{2152}$	extractwidth (param.)

gadowi co () // [/6])	
gsdevice (param.)	ext <u>214</u>
gsopt (param.)	extension $\underline{214}$
$\texttt{height} \; (\texttt{param.}) \dots \dots \qquad \qquad$	extract 8, <u>1140</u>
inkscape (param.) $6, \frac{716}{}$	extractapptex
inkscapearea (param.) $6, 716$	extractdistort 8, <u>1585</u>
inkscapedpi (param.) $\dots \dots \dots$	extractformat
inkscapeformat (param.) 6, 716	extractheight
inkscapelatex (param.) 6 , $\frac{20}{716}$	
inkscapeopt (param.) 6 , 716	extractkeepaspectratio 1585
lastpage (param.)	extractname 8, <u>1516</u>
	extractpath \ldots δ , $\underline{1516}$
latexopt (param.) 7, <u>1760</u>	extractpostex <u>1681</u>
magickoperator (param.) $7, \underline{1760}$	extractpreamble $\ldots 8, \underline{1213}$
magicksetting (param.) 7, <u>1760</u>	extractpreambleend $$
origin (param.)	extractpretex
$\mathtt{pretex}\;(\mathtt{param.})\dots\dots 6,\underline{716}$	extractruns 9, 1240
scale (param.)	extractscale
svgextension (param.) $6, \frac{716}{}$	extractwidth 8, <u>1585</u>
width (param.)	gsdevice
inkscape (opt.)	gsexe
inkscapearea (opt.) 5, <u>161</u>	,
inkscapedensity (opt.) $\underline{175}$	gsopt
inkscapedpi (opt.)	height 5, <u>252</u>
inkscapeexe (opt.)	inkscape $\dots 4, \underline{54}$
inkscapeformat (opt.)	inkscapearea $5, \underline{161}$
inkscapelatex (opt.)	inkscapedensity $\dots $ 175
inkscapename (opt.)	inkscapedpi $5, \underline{175}$
inkscapeopt (opt.)	inkscapeexe $5, \underline{192}$
inkscapepath (opt.)	inkscapeformat
111110 cap op a on (op a.)	inkscapelatex
К	inkscapename
keepaspectratio (opt.)	inkscapeopt $\dots \dots \dots$
100 (opt.)	inkscapepath $\dots \qquad 4, \frac{221}{221}$
1	keepaspectratio
lastpage (opt.)	lastpage 6, 332
	latex
latex (opt.)	1atex <u>157</u>
1 o t o morro () 0 1257	1-1
latexexe (opt.)	latexexe
latexext (opt.) 9, <u>1257</u>	latexext $9, \overline{1257}$
	latexext 9, $\frac{1257}{1257}$ latexopt 9, $\frac{1257}{1257}$
$\begin{array}{cccccccccccccccccccccccccccccccccccc$	$\begin{array}{cccccccccccccccccccccccccccccccccccc$
latexext (opt.) 9, 1257 latexopt (opt.) 9, 1257 M	$\begin{array}{cccccccccccccccccccccccccccccccccccc$
M M M Magickexe (opt.)	$\begin{array}{cccccccccccccccccccccccccccccccccccc$
M M M Magickexe (opt.) 11, 1472 Magickoperator (opt.) 11, 1472 Magicko	$\begin{array}{cccccccccccccccccccccccccccccccccccc$
M M M Magickexe (opt.)	$\begin{array}{cccccccccccccccccccccccccccccccccccc$
Magickexe (opt.) 9, 1257	$\begin{array}{cccccccccccccccccccccccccccccccccccc$
N	latexext 9, 1257 latexopt 9, 1257 magickexe 11, 1472 magickoperator 11, 1472 magicksetting 11, 1472 name 1516 notransparent 3, 30
N N N N N N N N N N	$\begin{array}{cccccccccccccccccccccccccccccccccccc$
N N N N N N N N N N	$\begin{array}{cccccccccccccccccccccccccccccccccccc$
N N N N N N N N N N	$\begin{array}{cccccccccccccccccccccccccccccccccccc$
N	latexext 9, 1257 latexopt 9, 1257 magickexe 11, 1472 magickoperator 11, 1472 magicksetting 11, 1472 name 1516 notransparent 3, 30 off 7, 134, 1168 on 7, 134, 1168 path 1516 pdf 1172
N	latexext 9, 1257 latexopt 9, 1257 magickexe 11, 1472 magickoperator 11, 1472 magicksetting 11, 1472 name 1516 notransparent 3, 30 off 7, 134, 1168 on 7, 134, 1168 path 1516 pdf 1172 pdflatex 1257
N	latexext 9, 1257 latexopt 9, 1257 magickexe 11, 1472 magickoperator 11, 1472 magicksetting 11, 1472 name 1516 notransparent 3, 30 off 7, 134, 1168 on 7, 134, 1168 path 1516 pdf 1172 pdflatex 1257 pdftoepsopt 9, 1295
N	latexext 9, 1257 latexopt 9, 1257 magickexe 11, 1472 magickoperator 11, 1472 magicksetting 11, 1472 name 1516 notransparent 3, 30 off 7, 134, 1168 on 7, 134, 1168 path 1516 pdf 1172 pdflatex 1257 pdftoepsopt 9, 1295 pdftops 1295
N	latexext 9, 1257 latexopt 9, 1257 magickexe 11, 1472 magickoperator 11, 1472 magicksetting 11, 1472 name 1516 notransparent 3, 30 off 7, 134, 1168 on 7, 134, 1168 path 1516 pdf 1172 pdflatex 1257 pdftoepsopt 9, 1295 pdftopsopt 9, 1295 pdftopsopt 9, 1295
Latexext (opt.)	latexext 9, 1257 latexopt 9, 1257 magickexe 11, 1472 magickoperator 11, 1472 magicksetting 11, 1472 name 1516 notransparent 3, 30 off 7, 134, 1168 on 7, 134, 1168 path 1516 pdf 1172 pdflatex 1257 pdftoepsopt 9, 1295 pdftopsopt 9, 1295 pdftopsopt 9, 1295 pdftopsopt 9, 1295 png 1420
latexext (opt.) 9, 1257 latexopt (opt.) 9, 1257 M magickexe (opt.) 11, 1472 magickoperator (opt.) 11, 1472 magicksetting (opt.) 11, 1472 N name (opt.) 1516 notransparent (opt.) 3, 30 noxcolor (opt.) 3, 30 off (opt.) 7, 134, 1168 on (opt.) 7, 134, 1168 options: apptex 5, 309 clean 9, 1717	latexext 9, 1257 latexopt 9, 1257 magickexe 11, 1472 magickoperator 11, 1472 magicksetting 11, 1472 name 1516 notransparent 3, 30 off 7, 134, 1168 on 7, 134, 1168 path 1516 pdf 1172 pdflatex 1257 pdftopsopt 9, 1295 png 1420 postex 309
Latexext (opt.)	latexext 9, 1257 latexopt 9, 1257 magickexe 11, 1472 magickoperator 11, 1472 magicksetting 11, 1472 name 1516 notransparent 3, 30 off 7, 134, 1168 on 7, 134, 1168 path 1516 pdf 1172 pdflatex 1257 pdftopsopt 9, 1295 pdftopsopt 9, 1295 pdftopsopt 9, 1295 pdftopsopt 9, 1295 png 1420 postex 309 preamble 1213
latexext (opt.) 9, 1257 latexopt (opt.) 9, 1257 M magickexe (opt.) 11, 1472 magickoperator (opt.) 11, 1472 magicksetting (opt.) 11, 1472 N name (opt.) 1516 notransparent (opt.) 3, 30 noxcolor (opt.) 3, 30 off (opt.) 7, 134, 1168 on (opt.) 7, 134, 1168 options: apptex 5, 309 clean 9, 1717 clear 1717 convert 10, 1344	latexext 9, 1257 latexopt 9, 1257 magickexe 11, 1472 magickoperator 11, 1472 magicksetting 11, 1472 name 1516 notransparent 3, 30 off 7, 134, 1168 on 7, 134, 1168 path 1516 pdf 1172 pdflatex 1257 pdftopsopt 9, 1295 pdftopsopt 9, 1295 pdftopsopt 9, 1295 png 1420 postex 309 preamble 1213 pretex 5, 309
Name (opt.) 7, 134, 1168 on (opt.) 7, 134, 1168 options: apptex 5, 309 clean 9, 1717 convert 10, 1344 convertdensity 1472 latexopt (opt.) 17, 17257 latexopt (opt.) 17, 17257 latexopt (opt.) 16, 10577 latexopt (opt.) 17, 10577 latexopt (opt.) 17, 10577 latexopt (opt.) 17, 10577 latexopt (opt.) 17, 10577 latexopt (opt.) 16, 10577 latexopt (opt.) 16, 10577 latexopt (opt.) 16, 10577 latexopt (opt.) 16, 10577 latexopt (opt.) 17, 10577 latexopt (opt.) 16, 10577 latexopt (opt.) 17, 10577 latexopt	latexext 9, 1257 latexopt 9, 1257 magickexe 11, 1472 magickoperator 11, 1472 magicksetting 11, 1472 name 1516 notransparent 3, 30 off 7, 134, 1168 on 7, 134, 1168 path 1516 pdf 1172 pdflatex 1257 pdftopsopt 9, 1295 pdftopsopt 9, 1295 pdftopsopt 9, 1295 postex 309 preamble 1213 pretex 5, 309 pstoepsopt 9, 1295
latexext (opt.) 9, 1257 latexopt (opt.) 9, 1257 M magickexe (opt.) 11, 1472 magickoperator (opt.) 11, 1472 magicksetting (opt.) 11, 1472 N name (opt.) 1516 notransparent (opt.) 3, 30 noxcolor (opt.) 3, 30 off (opt.) 7, 134, 1168 on (opt.) 7, 134, 1168 options: apptex 5, 309 clean 9, 1717 clear 1717 convert 10, 1344	latexext 9, 1257 latexopt 9, 1257 magickexe 11, 1472 magickoperator 11, 1472 magicksetting 11, 1472 name 1516 notransparent 3, 30 off 7, 134, 1168 on 7, 134, 1168 path 1516 pdf 1172 pdflatex 1257 pdftopsopt 9, 1295 pdftopsopt 9, 1295 pdftopsopt 1420 postex 309 preamble 1213 pretex 5, 309 pstoepsopt 9, 1295 pstopdfopt 9, 1295 pstopdfopt 9, 1295
Name (opt.) 7, 134, 1168 on (opt.) 7, 134, 1168 options: apptex 5, 309 clean 9, 1717 convert 10, 1344 convertdensity 1472 latexopt (opt.) 17, 17257 latexopt (opt.) 17, 17257 latexopt (opt.) 16, 10577 latexopt (opt.) 17, 10577 latexopt (opt.) 17, 10577 latexopt (opt.) 17, 10577 latexopt (opt.) 17, 10577 latexopt (opt.) 16, 10577 latexopt (opt.) 16, 10577 latexopt (opt.) 16, 10577 latexopt (opt.) 16, 10577 latexopt (opt.) 17, 10577 latexopt (opt.) 16, 10577 latexopt (opt.) 17, 10577 latexopt	latexext 9, 1257 latexopt 9, 1257 magickexe 11, 1472 magickoperator 11, 1472 magicksetting 11, 1472 name 1516 notransparent 3, 30 off 7, 134, 1168 on 7, 134, 1168 path 1516 pdf 1172 pdflatex 1257 pdftopsopt 9, 1295 pdftopsopt 9, 1295 pdftopsopt 9, 1295 postex 309 preamble 1213 pretex 5, 309 pstoepsopt 9, 1295
Name (opt.) 7, 134, 1168 on (opt.) 7, 134, 1168 options: apptex 5, 309 clean 9, 1717 convert 10, 1344 convertdpi 10, 1447	latexext 9, 1257 latexopt 9, 1257 magickexe 11, 1472 magickoperator 11, 1472 magicksetting 11, 1472 name 1516 notransparent 3, 30 off 7, 134, 1168 on 7, 134, 1168 path 1516 pdf 1172 pdflatex 1257 pdftopsopt 9, 1295 pdftopsopt 9, 1295 pdftopsopt 1420 postex 309 preamble 1213 pretex 5, 309 pstoepsopt 9, 1295 pstopdfopt 9, 1295 pstopdfopt 9, 1295
Name (opt.) 7, 134, 1168 on (opt.) 7, 134, 1168 options: apptex 5, 309 clean 9, 1717 convert deni convertdensity convertdensity 10, 1447 convert opt.) 10, 1420	latexext 9, 1257 latexopt 9, 1257 magickexe 11, 1472 magickoperator 11, 1472 magicksetting 11, 1472 name 1516 notransparent 3, 30 off 7, 134, 1168 on 7, 134, 1168 path 1516 pdf 1172 pdflatex 1257 pdftoepsopt 9, 1295 pdftops 1295 pdftopsopt 9, 1295 png 1420 postex 309 preamble 1213 pretex 5, 309 pstoepsopt 9, 1295 pstopdfopt 9, 1295 pstopdfopt 9, 1295 pstopdfopt 9, 1295 pstopdfopt 9, 1295 pstoplopt 9, 1295
latexext (opt.) 9, 1257 M magickexe (opt.) 11, 1472 magickoperator (opt.) 11, 1472 N name (opt.) 1516 notransparent (opt.) 3, 30 O Off (opt.) 7, 134, 1168 on (opt.) 7, 134, 1168 on (opt.) 7, 134, 1168 options: apptex 5, 309 clean 9, 1717 clear 1717 convert 10, 1344 convertdensity 1447 convertdormat 10, 1447 convertformat 10, 1420 distort 5, 252	latexext 9, 1257 latexopt 9, 1257 magickexe 11, 1472 magickoperator 11, 1472 magicksetting 11, 1472 name 1516 notransparent 3, 30 off 7, 134, 1168 on 7, 134, 1168 path 1516 pdf 1172 pdflatex 1257 pdftopsopt 9, 1295 pdftopsopt 9, 1295 pdftopsopt 1420 postex 309 preamble 1213 pretex 5, 309 pstoepsopt 9, 1295 pstopdfopt 9, 1295 pstopdfopt 9, 1295 pstogestension 5, 252 svgextension 5, 214
Latexext (opt.) 9, 1257	latexext 9, 1257 latexopt 9, 1257 magickexe 11, 1472 magicksetting 11, 1472 name 1516 notransparent 3, 30 noxcolor 3, 30 off 7, 134, 1168 on 7, 134, 1168 path 1516 pdf 1172 pdflatex 1257 pdftoepsopt 9, 1295 pdftops 1295 pdftopsopt 9, 1295 png 1420 postex 309 preamble 1213 pretex 5, 309 pstoepsopt 9, 1295 pstopdfopt 9, 1295 scale 5, 252 svgextension 5, 214 svgpath 202
Name (opt.) 7, 134, 1168	latexext 9, 1257 latexopt 9, 1257 magickexe 11, 1472 magickoperator 11, 1472 magicksetting 11, 1472 name 1516 notransparent 3, 30 off 7, 134, 1168 on 7, 134, 1168 path 1516 pdf 1172 pdflatex 1257 pdftoepsopt 9, 1295 pdftops 1295 pdftopsopt 9, 1295 png 1420 postex 309 preamble 1213 pretex 5, 309 pstoepsopt 9, 1295 pstopdfopt 9, 1295 scale 5, 252 svgextension 5, 214 svgpath 202 tex 157
Name (opt.) 7, 134, 1168 on (opt.) 7, 134, 1168 options: apptex 5, 309 clean 9, 1717 clear 1717 convert 10, 1344 convertdensity 10, 1447 convertdensity 10, 1447 convertdensity 10, 1447 convert 10, 1420 distort 5, 252 draft 6, 346 dvipsopt 9, 1295 end 1213 distort 1213	latexext 9, 1257 latexopt 9, 1257 magickexe 11, 1472 magickoperator 11, 1472 magicksetting 11, 1472 name 1516 notransparent 3, 30 off 7, 134, 1168 on 7, 134, 1168 path 1516 pdf 1172 pdflatex 1257 pdftopsopt 9, 1295 pdftops 1295 pdftopsopt 9, 1295 pdftopsopt 9, 1295 preamble 1213 pretex 5, 309 pstoepsopt 9, 1295 pstopdfopt 9, 1295 scale 5, 252 svgextension 5, 214 svgpath 202 tex 157 usetransparent 3, 30

Р	magicksetting-\includesvg 7 , 1760
parameters:	origin-\includeinkscape 6 , 805
$angle-includeinkscape \dots 6, 805$	origin-\includesvg $6, \underline{748}$
angle-\includesvg $\dots \dots \dots$	pretex-\includeinkscape 6 , 805
apptex-\includeinkscape $6, 805$	pretex-\includesvg $\ldots \ldots 6, \overline{716}$
apptex-\includesvg 6, 716	scale-\includeinkscape 6, 805
clean - \includeinkscape 7, 1760	scale-\includesvg $6, 716$
·	
	- · ·
$convert-\c$	width-\includeinkscape $6, 805$
convert - \includesvg 7, 1760	width-\includesvg $6, \frac{716}{200}$
convertdpi-\includeinkscape . 7 , 1760	path (opt.) 1516
convertdpi-\includesvg 7 , 1760	pdf (opt.)
convertformat – \includeinkscape 7 , 1760	pdflatex (opt.) 1257
convertformat-\includesvg 7 , 1760	pdftoepsopt (opt.)
$\verb distort-\include inkscape 6, \underline{805} $	pdftops (opt.)
$\texttt{distort-} \setminus \texttt{includesvg} \ \dots \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ $	pdftopsopt (opt.)
$draft-\includeinkscape \dots 6, 805$	png (opt.)
$draft-\includesvg$	postex (opt.) 309
exclude - \includeinkscape 7, 1760	preamble (opt.) 1213
exclude $-$ includes y	pretex (opt.)
extract – \includeinkscape $7, \frac{2760}{1760}$	pstoepsopt (opt.)
extract \includesvg \ldots \tau, \frac{1760}{1760}	
<u> </u>	pstopdfopt (opt.)
extractangle -\includeinkscape 7, 1760	S
extractangle - \includesvg $\frac{7}{1760}$	_
extractapptex-\includeinkscape $7, \underline{1760}$	scale (opt.)
extractapptex-\includesvg 7 , 1760	\setsvg <u>699</u>
$\verb"extractdistort-\\ \verb"includeinkscape" .$	\svg@append@input@path 409
$ \gamma, \underline{1760} $	\svg@box $\underline{12}$
extractdistort-\includesvg 7 , 1760	\svg@deactivate@dq $\dots 349$
extractformat – \includeinkscape 7 , 1760	\svg@deprecated@key $\underline{22}$
extractformat-\includesvg 7 , 1760	$\verb \svg@deprecated@param \underline{825}$
extractheight - \includeinkscape 7 , 1760	\svg@dummy@key <u>1113</u>
extractheight-\includesvg 7, 1760	$\verb \svg@extension@parse \underline{564}$
extractpreamble - \includeinkscape	\svg@extension@@parse 564
$ \begin{array}{ccccccccccccccccccccccccccccccccc$	\svg@extract <u>1802</u>
extractpreamble - \includesvg . 7, 1760	\svg@file@base 487
extractpretex-\includeinkscape 7, 1760	\svg@file@ext 214
extractpretex \includesinkscape 7, \frac{1760}{1760}	\svg@file@missing 600
extractruns - \includeinkscape 7, 1760	\svg@file@name
extractruns – \includesvg $7, \frac{1760}{1760}$	\svg@file@path 487
extractscale - \includeinkscape 7, 1760	\svg@file@suffix 487
extractscale - \includesvg $7, \underline{1760}$	\svg@filename@parse 527
extractwidth-\includeinkscape 7 , 1760	\svg@get@lastpage <u>930</u>
extractwidth-\includesvg 7 , 1760	$\verb \svg@get@path \dots \dots \underline{487}$
gsdevice-\includeinkscape 7 , 1760	\svg@iffilenewer $\underline{646}$
gsdevice-\includesvg	\svg@ifvalueisrelax $\underline{476}$
$\mathtt{gsopt-} \\ includeinkscape \ldots 7, \underline{1760}$	\svg@includegraphics@file $\underline{1095}$
$gsopt-\includesvg$	$\svg@includegraphics@patched 1095$
height-\includeinkscape $6, 805$	\svg@includegraphics@saved 1063
height-\includesvg $\dots \dots \dots$	\svg@ink@area <u>161</u>
inkscape-\includesvg $6, \frac{716}{716}$	\svg@ink@cmd 922
inkscapearea-\includesvg $6, \frac{716}{716}$	\svg@ink@dpi 175
inkscapedpi-\includesvg $6, \frac{716}{716}$	\svg@ink@exe 192
inkscapeformat-\includeinkscape $6, 805$	\svg@ink@format 136
inkscapeformat \includesinkscape θ , $\frac{600}{500}$	\svg@ink@latex
	_
inkscapelatex-\includeinkscape $6, \underline{805}$	\svg@ink@mode $\underline{54}$
inkscapelatex-\includesvg 6, 716	\svg@ink@opt <u>192</u>
inkscapeopt -\includesvg $6, \frac{716}{2007}$	\svg@ink@run
lastpage-\includeinkscape 6, 805	\svg@input <u>981</u>
$lastpage-\c includes vg \dots 6, \underline{745}$	\svg@input@path 701
$\texttt{latexopt-} \\ \texttt{\lambda} \\ \texttt{includeinkscape} \dots \textcolor{red}{7}, \underline{1760} \\ \\ \texttt{\lambda} \\ \$	\svg@@input $\underline{981}$
$\texttt{latexopt-} \setminus \texttt{includesvg} \ \dots \ 7, \ \underline{1760}$	\svg@local@param@def $\dots \dots \underline{678}$
${\tt magickoperator-\includeinkscape}$.	$\verb \svg@local@param@set \underline{678}$
$\dots \dots $	$\verb \svg@local@param@use \underline{678}$
magickoperator – \includesvg 7 , 1760	\svg@normalize@path 450
magicksetting-\includeinkscape $7, \overline{1760}$	\svg@normalize@@path 450

$\verb \svg@out@base \underline{221} $	$\verb \svgx@ifkeyandval $
\svg@out@name <u>221</u>	\svgx@latex@exe <u>1257</u>
\svg@out@path <u>221</u>	\svgx@latex@ext <u>1257</u>
$\verb \svg@param@apptex \underline{309}$	$\verb \svgx@latex@opt \underline{1257}$
<pre>svg@param@lastpage (counter) 332</pre>	\svgx@magick@cmd <u>2213</u>
\svg@param@pretex <u>309</u>	\svgx@magick@exe <u>1472</u>
$\verb \svg@param@scale \underline{252} $	\svgx@magick@opr <u>1472</u>
$\verb \svg@param@width $	$\verb \svgx@magick@set \underline{1472}$
\svg@patches <u>1063</u>	\svgx@move <u>2242</u>
$\svg@pictur@patched \dots 1072$	\svgx@onlywindows $\underline{2290}$
\svg@picture@saved <u>1063</u>	$svgx@out@count (counter) \dots 1516$
\svg@quotes@check <u>385</u>	\svgx@out@name <u>1516</u>
\svg@quotes@@check 385	\svgx@out@path <u>1516</u>
\svg@quotes@remove 360	\svgx@out@sec <u>2129</u>
\svg@quotes@@remove 360	\svgx@param@apptex <u>1681</u>
\svg@remove@leadingchar 393	\svgx @param@distort 1585
\svg@sanitize@dq 354	$\verb \svgx@param@pretex \underline{1681}$
\svg@set@input@path 409	\svgx@param@scale <u>1585</u>
\svg@shell@mkdir <u>2461</u>	\svgx @param@width 1585
\svg@shell@@mkdir <u>2461</u>	\svgx@pdftoeps@exe 1295
\svg@shell@mv 2461	\svgx@pdftoeps@opt 1295
\svg@shell@@mv	\svgx@pdftops@exe <u>1295</u>
\svg@shell@rm <u>2461</u>	\svgx@pdftops@opt 1295
\svg@shell@@rm 2461	\svgx@preamble <u>1213</u>
\svg@tempa <u>12</u>	\svgx@pstoeps@exe <u>1295</u>
\svg@tempb	\svgx@pstoeps@opt <u>1295</u>
\svg@wrn@scale 964	\svgx@pstopdf@exe <u>1295</u>
sygextension (opt.)	\svgx@pstopdf@opt <u>1295</u>
\syghidepreambleend 9, <u>1782</u>	\svgx@read@line <u>1802</u>
\svghidepreamblestart 9, <u>1782</u> \svgpath 4, <u>701</u>	\svgx@read@preamble@from 2157
-	\svgx@read@preamble@skip 2157
svgpath (opt.) 202 \svgsetup 3, 7, 699	\svgx@read@preamble@till 2157
\svgx@clean	svgx@runs (counter)
\svgx@cnv@cmd	\svgx@setbox
\svgx@cnv@dpi 1447	\svgx@setformatkey
\svgx@cnv@format 1420	\svgx@stream@out 1802
\svgx@cnv@get@dpi 2296	\svgx@useformatkey 2342
\svgx@cnv@get@informat 2194	\svgxoutputbox
\svgx@cnv@informat 2194	\svgxsetbox
\svgx@documentclass 2152	\svgxsetpapersize
\svgx@dvips@exe <u>1295</u>	\bvg\test\test\test\test\test\test\test\tes
\svgx@dvips@opt 1295	Т
\svgx@endpreamble <u>1213</u>	tex (opt.)
\svgx@format <u>1172</u>	
\svgx@get@out@sec <u>2129</u>	U
\svgx@gs@cmd <u>2213</u>	usetransparent (opt.)
$\verb \svgx@gs@device \underline{1494}$	usexcolor (opt.)
\svgx@gs@exe <u>1494</u>	
\svgx@gs@opt <u>1494</u>	W
\svgx@ifinlist <u>2278</u>	$\mathtt{width}\;(\mathrm{opt.})\;\ldots\ldots\;$
Change History	
v1.0	huge number of packages which deal
General	with this topic and the large variety
initial version by Philip Ilten 2	of implementing this functionality;
v r	naming exported graphics after their
v2.00	consecutive numbering can't be
General	ensured for all variants of subfigures,

support of subfigures stopped due to the

 ${\tt clean} \; ({\tt opt.}) {:} \; \; changes, \; file \; list \; possible \quad \underline{1717}$

convert (opt.): changed/extended ..

Implementation

convertdpi (opt.): new <u>1447</u>	pdftopsopt (opt.): new <u>1295</u>
convertformat (opt.): new $\underline{1420}$	png (opt.): deprecated <u>1420</u>
$\mathtt{draft} \; (\mathrm{opt.}) \colon \; new \; \ldots \; \underline{346}$	postex (opt.): deprecated 309
dvipsopt (opt.): new <u>1295</u>	preamble (opt.): deprecated 1213
end (opt.): deprecated <u>1213</u>	pstoepsopt (opt.): new <u>1295</u>
eps (opt.): deprecated <u>1172</u>	pstopdfopt (opt.): new <u>1295</u>
extract (opt.): new <u>1140</u>	scale (opt.): new
extractapptex (opt.): new 1681	\setsvg: deprecated 699
extractformat (opt.): new 1172	\svghidepreambleend: new 1782
extractheight (opt.): new 1585	\svghidepreamblestart: new 1782
extractname (opt.): new 1516	\svgpath: new
extractpath (opt.): new 1516	svgpath (opt.): deprecated 202
extractpreamble (opt.): new 1213	\svgsetup: new 699
extractpreambleend (opt.): new 1213	usetransparent (opt.): new 30
extractpretex (opt.): new 1681	usexcolor (opt.): new
extractruns (opt.): new 1240	ubercolor (opt.). new
extractscale (opt.): new 1585	v2.00a
extractwidth (opt.): new 1585	Implementation
	\svgxsetpapersize: Bug fix for
	checking stock- and mediasizes <u>2390</u>
	0
· · · · · · · · · · · · · · · · · ·	v2.00b
height (opt.): new	Implementation
\includeinkscape: new 764	latex (opt.): new, alternative key for
\includesvg:	inkscapelatex 157
changes, especially to optional	tex (opt.): new, alternative key for
parameters	$inkscapelatex \dots 157$
angle (param.): new	<u></u>
draft (param.): new	v2.01
height (param.): new	General
inkscape (param.): new	new option svgextension to change the
inkscapearea (param.): new 716	format of files exported by <i>Inkscape</i>
inkscapedpi (param.): new 716	from svg to a custom one 2
inkscapeformat (param.): new 716	usage of $\langle tex filename \rangle$ within
inkscapelatex (param.): new 716	Inkscape graphics locates files in all
-	
inkscapeopt (param.): new 716	declared searched folders 2
inkscapeopt (param.): new 716 lastpage (param.): new 745	declared searched folders
inkscapeopt (param.): new 716 lastpage (param.): new 745 origin (param.): new 748	declared searched folders
inkscapeopt (param.): new 716 lastpage (param.): new 745 origin (param.): new 748 scale (param.): new 716	declared searched folders
inkscapeopt (param.): new 716 lastpage (param.): new 745 origin (param.): new 748 scale (param.): new 716 inkscape (opt.): changed/extended 54	declared searched folders
inkscapeopt (param.): new 716 lastpage (param.): new 745 origin (param.): new 748 scale (param.): new 716 inkscape (opt.): changed/extended 54 inkscapearea (opt.): new 161	declared searched folders
inkscapeopt (param.): new 716 lastpage (param.): new 745 origin (param.): new 748 scale (param.): new 716 inkscape (opt.): changed/extended 54 inkscapearea (opt.): new 161 inkscapedpi (opt.): new 175	declared searched folders
inkscapeopt (param.): new 716 lastpage (param.): new 745 origin (param.): new 748 scale (param.): new 716 inkscape (opt.): changed/extended 54 inkscapearea (opt.): new 161 inkscapedpi (opt.): new 175 inkscapeexe (opt.): new 192	declared searched folders
inkscapeopt (param.): new 716 lastpage (param.): new 745 origin (param.): new 748 scale (param.): new 716 inkscape (opt.): changed/extended 54 inkscapearea (opt.): new 161 inkscapedpi (opt.): new 175 inkscapeexe (opt.): new 192 inkscapeformat (opt.): new 136	declared searched folders
inkscapeopt (param.): new 716 lastpage (param.): new 745 origin (param.): new 748 scale (param.): new 716 inkscape (opt.): changed/extended 54 inkscapearea (opt.): new 161 inkscapedpi (opt.): new 175 inkscapeexe (opt.): new 192 inkscapelatex (opt.): new 136 inkscapelatex (opt.): new 157	declared searched folders
inkscapeopt (param.): new 716 lastpage (param.): new 745 origin (param.): new 716 inkscape (opt.): changed/extended 54 inkscapearea (opt.): new 161 inkscapedpi (opt.): new 175 inkscapeexe (opt.): new 192 inkscapeformat (opt.): new 136 inkscapelatex (opt.): new 157 inkscapename (opt.): new 221	declared searched folders
inkscapeopt (param.): new 716 lastpage (param.): new 745 origin (param.): new 716 inkscape (opt.): changed/extended 54 inkscapearea (opt.): new 161 inkscapedpi (opt.): new 175 inkscapeexe (opt.): new 192 inkscapeformat (opt.): new 157 inkscapename (opt.): new 221 inkscapeopt (opt.): new 192	declared searched folders
inkscapeopt (param.): new 716 lastpage (param.): new 745 origin (param.): new 748 scale (param.): new 716 inkscape (opt.): changed/extended 54 inkscapearea (opt.): new 161 inkscapedpi (opt.): new 192 inkscapeformat (opt.): new 136 inkscapelatex (opt.): new 157 inkscapeopt (opt.): new 221 inkscapepath (opt.): new 192 inkscapepath (opt.): new 221	declared searched folders
inkscapeopt (param.): new 716 lastpage (param.): new 745 origin (param.): new 748 scale (param.): new 716 inkscape (opt.): changed/extended 54 inkscapearea (opt.): new 161 inkscapedpi (opt.): new 192 inkscapeformat (opt.): new 136 inkscapelatex (opt.): new 157 inkscapename (opt.): new 221 inkscapepath (opt.): new 221 lastpage (opt.): new 332	declared searched folders
inkscapeopt (param.): new 716 lastpage (param.): new 745 origin (param.): new 748 scale (param.): new 716 inkscape (opt.): changed/extended 54 inkscapearea (opt.): new 161 inkscapedpi (opt.): new 192 inkscapeformat (opt.): new 136 inkscapelatex (opt.): new 157 inkscapeopt (opt.): new 221 inkscapepath (opt.): new 192 inkscapepath (opt.): new 221	declared searched folders
inkscapeopt (param.): new 716 lastpage (param.): new 745 origin (param.): new 748 scale (param.): new 716 inkscape (opt.): changed/extended 54 inkscapearea (opt.): new 161 inkscapedpi (opt.): new 192 inkscapeformat (opt.): new 136 inkscapelatex (opt.): new 157 inkscapename (opt.): new 221 inkscapepath (opt.): new 221 lastpage (opt.): new 332	declared searched folders
inkscapeopt (param.): new 716 lastpage (param.): new 745 origin (param.): new 748 scale (param.): new 716 inkscape (opt.): changed/extended 54 inkscapearea (opt.): new 161 inkscapedpi (opt.): new 192 inkscapeformat (opt.): new 136 inkscapelatex (opt.): new 157 inkscapename (opt.): new 221 inkscapepath (opt.): new 221 lastpage (opt.): new 332 latexexe (opt.): new 1257	declared searched folders
inkscapeopt (param.): new 716 lastpage (param.): new 745 origin (param.): new 748 scale (param.): new 716 inkscape (opt.): changed/extended 54 inkscapearea (opt.): new 161 inkscapedpi (opt.): new 192 inkscapeexe (opt.): new 136 inkscapelatex (opt.): new 157 inkscapename (opt.): new 221 inkscapepath (opt.): new 221 lastpage (opt.): new 332 latexexe (opt.): new 1257 latexext (opt.): new 1257	declared searched folders
inkscapeopt (param.): new 716 lastpage (param.): new 745 origin (param.): new 748 scale (param.): new 716 inkscape (opt.): changed/extended 54 inkscapearea (opt.): new 161 inkscapedpi (opt.): new 192 inkscapeexe (opt.): new 136 inkscapelatex (opt.): new 157 inkscapename (opt.): new 221 inkscapepath (opt.): new 221 lastpage (opt.): new 332 latexexe (opt.): new 1257 latexext (opt.): new 1257 latexopt (opt.): new 1257 latexopt (opt.): new 1257	declared searched folders
inkscapeopt (param.): new 716 lastpage (param.): new 745 origin (param.): new 748 scale (param.): new 716 inkscape (opt.): changed/extended 54 inkscapearea (opt.): new 161 inkscapedpi (opt.): new 192 inkscapeexe (opt.): new 136 inkscapelatex (opt.): new 157 inkscapename (opt.): new 221 inkscapepath (opt.): new 221 lastpage (opt.): new 332 latexexe (opt.): new 1257 latexopt (opt.): new 1257 latexopt (opt.): new 1257 magickexe (opt.): new 1472	declared searched folders
inkscapeopt (param.): new 716 lastpage (param.): new 745 origin (param.): new 748 scale (param.): new 716 inkscape (opt.): changed/extended 54 inkscapearea (opt.): new 161 inkscapedpi (opt.): new 192 inkscapeexe (opt.): new 136 inkscapelatex (opt.): new 157 inkscapename (opt.): new 221 inkscapeopt (opt.): new 192 inkscapepath (opt.): new 332 latexexe (opt.): new 1257 latexext (opt.): new 1257 latexopt (opt.): new 1257 latexopt (opt.): new 1257 magickexe (opt.): new 1472 magickperator (opt.): new 1472 magicksetting (opt.): new 1472	declared searched folders
inkscapeopt (param.): new 716 lastpage (param.): new 745 origin (param.): new 748 scale (param.): new 716 inkscape (opt.): changed/extended 54 inkscapearea (opt.): new 161 inkscapedpi (opt.): new 192 inkscapeexe (opt.): new 136 inkscapelatex (opt.): new 157 inkscapename (opt.): new 221 inkscapeopt (opt.): new 192 inkscapepath (opt.): new 332 latexexe (opt.): new 1257 latexext (opt.): new 1257 latexopt (opt.): new 1257 latexopt (opt.): new 1257 magickexe (opt.): new 1472 magicksetting (opt.): new 1472 name (opt.): new 1472	declared searched folders
inkscapeopt (param.): new 716 lastpage (param.): new 745 origin (param.): new 748 scale (param.): new 716 inkscape (opt.): changed/extended 54 inkscapearea (opt.): new 161 inkscapedpi (opt.): new 192 inkscapeexe (opt.): new 136 inkscapelatex (opt.): new 157 inkscapename (opt.): new 221 inkscapeopt (opt.): new 192 inkscapepath (opt.): new 1257 latexexe (opt.): new 1257 latexext (opt.): new 1257 latexopt (opt.): new 1257 magickexe (opt.): new 1472 magicksetting (opt.): new 1472 name (opt.): new 1472 name (opt.): deprecated 1516	declared searched folders
inkscapeopt (param.): new 716 lastpage (param.): new 745 origin (param.): new 748 scale (param.): new 716 inkscape (opt.): changed/extended 54 inkscapearea (opt.): new 161 inkscapedpi (opt.): new 192 inkscapeexe (opt.): new 136 inkscapelatex (opt.): new 157 inkscapename (opt.): new 221 inkscapeopt (opt.): new 192 inkscapepath (opt.): new 1257 latexexe (opt.): new 1257 latexext (opt.): new 1257 latexopt (opt.): new 1257 magickexe (opt.): new 1257 magickexe (opt.): new 1472 magicksetting (opt.): new 1472 magicksetting (opt.): new 1472 name (opt.): 1516 support of subfig removed 1516	declared searched folders
inkscapeopt (param.): new 716 lastpage (param.): new 745 origin (param.): new 748 scale (param.): new 716 inkscape (opt.): changed/extended 54 inkscapedpi (opt.): new 161 inkscapedpi (opt.): new 192 inkscapeexe (opt.): new 136 inkscapelatex (opt.): new 157 inkscapename (opt.): new 221 inkscapeopt (opt.): new 192 inkscapepath (opt.): new 1257 lastpage (opt.): new 1257 latexex (opt.): new 1257 latexopt (opt.): new 1257 magickexe (opt.): new 1257 magickexe (opt.): new 1472 magickexe (opt.): new 1472 magicksetting (opt.): new 1472 name (opt.): 1256 support of subfig removed 1516 notransparent (opt.): new 30	declared searched folders
inkscapeopt (param.): new 716 lastpage (param.): new 745 origin (param.): new 748 scale (param.): new 716 inkscape (opt.): changed/extended 54 inkscapedpi (opt.): new 161 inkscapedpi (opt.): new 192 inkscapeexe (opt.): new 136 inkscapelatex (opt.): new 157 inkscapename (opt.): new 221 inkscapeopt (opt.): new 192 inkscapepath (opt.): new 1257 latexexe (opt.): new 1257 latexext (opt.): new 1257 latexopt (opt.): new 1257 magickexe (opt.): new 1257 magickexe (opt.): new 1472 magickexe (opt.): new 1472 magicksetting (opt.): new 1472 name (opt.): 1516 support of subfig removed 1516 notransparent (opt.): new 30 noxcolor (opt.): new 30	declared searched folders
inkscapeopt (param.): new 716 lastpage (param.): new 745 origin (param.): new 748 scale (param.): new 716 inkscape (opt.): changed/extended 54 inkscapearea (opt.): new 161 inkscapedpi (opt.): new 192 inkscapeexe (opt.): new 136 inkscapelatex (opt.): new 157 inkscapelatex (opt.): new 221 inkscapeopt (opt.): new 192 inkscapeopt (opt.): new 1257 lastpage (opt.): new 1257 latexexe (opt.): new 1257 latexopt (opt.): new 1257 latexopt (opt.): new 1257 magickexe (opt.): new 1257 magickexe (opt.): new 1472 magicksetting (opt.): new 1472 name (opt.): 1472 name (opt.): 0 deprecated 1516 support of subfig removed 1516 noxcolor (opt.): new 30 off (opt.): new 134, 1168	declared searched folders
inkscapeopt (param.): new 716 lastpage (param.): new 745 origin (param.): new 748 scale (param.): new 716 inkscape (opt.): changed/extended 54 inkscapearea (opt.): new 161 inkscapedpi (opt.): new 192 inkscapeexe (opt.): new 136 inkscapelatex (opt.): new 157 inkscapename (opt.): new 221 inkscapeopt (opt.): new 192 inkscapepath (opt.): new 221 lastpage (opt.): new 1257 latexexe (opt.): new 1257 latexopt (opt.): new 1257 latexopt (opt.): new 1257 magickexe (opt.): new 1257 magickexe (opt.): new 1472 magicksetting (opt.): new 1472 name (opt.): 160 deprecated 1516 support of subfig removed 1516 notransparent (opt.): new 30 not(opt.): new 134, 1168 on (opt.): new 134, 1168	declared searched folders
inkscapeopt (param.): new 716 lastpage (param.): new 745 origin (param.): new 748 scale (param.): new 716 inkscape (opt.): changed/extended 54 inkscapearea (opt.): new 161 inkscapedpi (opt.): new 192 inkscapeexe (opt.): new 136 inkscapelatex (opt.): new 157 inkscapename (opt.): new 221 inkscapeopt (opt.): new 192 inkscapepath (opt.): new 221 lastpage (opt.): new 1257 latexexe (opt.): new 1257 latexopt (opt.): new 1257 latexopt (opt.): new 1257 magickexe (opt.): new 1472 magicksetting (opt.): new 1472 name (opt.): deprecated 1516 support of subfig removed 1516 notransparent (opt.): new 30 noxcolor (opt.): new 134, 1168 on (opt.): new 134, 1168 on (opt.): deprecated 1516	declared searched folders
inkscapeopt (param.): new 716 lastpage (param.): new 745 origin (param.): new 748 scale (param.): new 716 inkscape (opt.): changed/extended 54 inkscapearea (opt.): new 161 inkscapedpi (opt.): new 175 inkscapeexe (opt.): new 192 inkscapeformat (opt.): new 157 inkscapelatex (opt.): new 221 inkscapeopt (opt.): new 192 inkscapeopt (opt.): new 221 lastpage (opt.): new 1257 latexexe (opt.): new 1257 latexext (opt.): new 1257 latexopt (opt.): new 1257 magickexe (opt.): new 1472 magicksetting (opt.): new 1472 magicksetting (opt.): new 1472 name (opt.): 30 noxcolor (opt.): new 30 noxcolor (opt.): new 30 noxcolor (opt.): new 134, 1168 on (opt.): new 134, 1168 path (opt.): deprecated 1516 pdf (opt.):	declared searched folders
inkscapeopt (param.): new 716 lastpage (param.): new 745 origin (param.): new 748 scale (param.): new 716 inkscape (opt.): changed/extended 54 inkscapearea (opt.): new 161 inkscapedpi (opt.): new 192 inkscapeexe (opt.): new 192 inkscapeformat (opt.): new 157 inkscapelatex (opt.): new 192 inkscapeopt (opt.): new 221 inkscapeopt (opt.): new 192 inkscapepath (opt.): new 221 lastpage (opt.): new 1257 latexexe (opt.): new 1257 latexopt (opt.): new 1257 latexopt (opt.): new 1257 magickexe (opt.): new 1472 magicksetting (opt.): new 1472 magicksetting (opt.): new 1516 support of subfig removed 1516 noxcolor (opt.): new 30 noxcolor (opt.): new 134, 1168 on (opt.): new 134, 1168 on (opt.): deprecated 1516	declared searched folders
inkscapeopt (param.): new 716 lastpage (param.): new 745 origin (param.): new 748 scale (param.): new 716 inkscape (opt.): changed/extended 54 inkscapearea (opt.): new 161 inkscapedpi (opt.): new 175 inkscapeexe (opt.): new 192 inkscapeformat (opt.): new 157 inkscapelatex (opt.): new 221 inkscapeopt (opt.): new 192 inkscapeopt (opt.): new 221 lastpage (opt.): new 1257 latexexe (opt.): new 1257 latexext (opt.): new 1257 latexopt (opt.): new 1257 magickexe (opt.): new 1472 magicksetting (opt.): new 1472 magicksetting (opt.): new 1472 name (opt.): 30 noxcolor (opt.): new 30 noxcolor (opt.): new 30 noxcolor (opt.): new 134, 1168 on (opt.): new 134, 1168 path (opt.): deprecated 1516 pdf (opt.):	declared searched folders

\includeinkscape:	usage of \svg@remove@leadingchar 214
usage of \svg@extension@parse $\underline{764}$	\svgpath: parse argument for enclosing
extractdistort (param.): new <u>1760</u>	braces and provide if necessary $\dots 701$
\includesvg:	\svgx@setbox: new <u>2375</u>
switched to \svg@filename@parse 713	\svgxsetbox: late execution of
angle (param.): validation of	\svgxsetpapersize $\underline{2375}$
argument	
distort (param.): new	v2.02a
extractangle (param.): new <u>1760</u>	General
extractdistort (param.): new <u>1760</u>	fix bug for package polyglossia which
inkscape (opt.): usage of	fakes babel poorly
\svg@sanitize@dq $\dots 54$	Implementation
inkscapepath (opt.): usage of	\svg@deactivate@dq: bug fix for
$\verb \svg@sanitize@dq$	polyglossia <u>349</u>
keepaspectratio (opt.): new \dots 252	2.024
\svg@append@input@path: avoid	v2.02b
duplicates in \input@path 409	General
\svg@deactivate@dq: new 349	fix bug for package tikzscale which
\svg@extension@parse: new 564	changes \includegraphics globally 2
\svg@extension@@parse: new <u>564</u>	Implementation
\svg@file@missing: notify svg file	\svg@patches: fix bug for package tikzscale: store original definitions of
when missing exported files \dots 600	\picture and \includegraphics
\svg@filename@parse:	right after loading package svg . 1063
usage of \svg@extension@parse 527	right after loading package svg . 1005
usage of \svg@remove@leadingchar 527	v2.02c
usage of \svg@sanitize@dq <u>527</u>	General
\svg@local@param@set: reasonable	fix bugs with current kernel (2019/10/01)
value for key distort <u>678</u>	regarding file name parsing 2
\svg@normalize@path: usage of	regarding me name pareing ******
\svg@deactivate@dq $\dots $ 450	v2.02d
\svg@quotes@remove:	General
calling \svg@quotes@check 360	fix bugs with current kernel
usage of $\svg@sanitize@dq$ 360	(2019/10/01) regarding file name
\svg@remove@leadingchar: new 393	parsing, see https://github.com/
\svg@sanitize@dq: new 354	mrpiggi/svg/issues/16 2
\svg@set@input@path: usage of	Implementation
\svg@deactivate@dq $\dots $ $\underline{409}$	\svg@iffilenewer: use \filemoddate
<pre>svgextension (opt.):</pre>	with XeLATEX, see https://github.
usage of \svgQquotes@remove $\underline{214}$	com/mrpiggi/svg/issues/12 $\underline{646}$