

# The `autopdf` package<sup>\*</sup>

Karl Wette

October 30, 2012

## 1 Introduction

This package facilitates the conversion of various graphics formats to formats supported by pdf $\LaTeX$  (e.g. PDF). It has the following features:

- It uses Ghostscript<sup>1</sup> and GraphicsMagick<sup>2</sup> to perform graphics conversions, and therefore can convert any graphics formats that are understood by GraphicsMagick. (If only conversion from EPS to PDF conversion is needed, only Ghostscript is required.) Hybrid  $\LaTeX$ /EPS graphics, as produced by e.g. Gnuplot, as also supported. `autopdf` always produces a separate file for each converted graphic.
- Graphics conversion is performed on the fly, i.e. as pdf $\LaTeX$  processes the document. For this to work, pdf $\LaTeX$  must be run in “shell escape” mode, so that calls to Ghostscript and GraphicsMagick can be executed. Aside from Ghostscript and GraphicsMagick, no other external programs or scripts are required.
- When converting EPS or  $\LaTeX$ /EPS graphics, a wrapper  $\LaTeX$  file is generated to encapsulate the EPS graphic. `autopdf` tries to transfer relevant properties of the parent document, such as the current font, to the wrapper  $\LaTeX$  file, so that any  $\LaTeX$  typesetting in the graphic has a similar look to the rest of the documents. Custom  $\LaTeX$  commands can also be easily transferred to the wrapper  $\LaTeX$  file, and special support is provided for `PSfrag`<sup>3</sup>.
- After conversion, MD5 checksums of each input graphic and any associated files are stored. When pdf $\LaTeX$  is next run, the checksums are used to determine whether any part of the graphic has changed, and therefore whether a re-conversion is needed.

There are also a wide variety of graphics conversion packages available on CTAN<sup>4</sup>, particularly for the conversion of EPS graphics to PDF. Depending on your needs, one of these packages may be better suited. Many of the features of

---

<sup>\*</sup>This document corresponds to `autopdf` v1.0, dated 2012/04/22.

<sup>1</sup><http://www.ghostscript.com/>

<sup>2</sup><http://www.graphicsmagick.org/>

<sup>3</sup><http://www.ctan.org/pkg/psfrag>

<sup>4</sup><http://www.ctan.org/>

autopdf were inspired by the features provided by the `epstopdf`<sup>5</sup> and `auto-pst-pdf`<sup>6</sup> packages, and the `fragmaster.pl`<sup>7</sup> script.

## 2 Usage

Include the package:

```
\usepackage[options...]{autopdf}
```

Available options are:

**from** Default file extension of input graphics files; used if no file extension is present in the file name given to `\includegraphics`. Defaults to `.eps`.

**to** Default file extension of output graphics files; determines what format graphics are converted to. Defaults to `.pdf`.

**logfile** File extension of the log file which records the checksums of converted graphics files. The full file name is created by prepending the name of the current document, i.e. `\jobname.logfile`. Defaults to `autopdf_log`.

**nologfile** Takes no arguments; prevents the log file being created.

**showcmds** Print the command lines calling Ghostscript/GraphicsMagick to the pdfL<sup>A</sup>T<sub>E</sub>X log file as they are executed. Values are `true` or `false` (default).

**cleanup** Remove intermediate files after a successful conversion. Values are `true` (default) or `false`.

**scale** Scale input (L<sup>A</sup>T<sub>E</sub>X/)`EPS` graphic as they are converted, using any arguments supplied to `\includegraphics`. Values are `true` (default) or `false`.

**margin** Add an additional margin to input (L<sup>A</sup>T<sub>E</sub>X/)`EPS` graphics. Defaults to `0pt`.

**resolution** Specifies the resolution of the output graphics, in dots per inch. Defaults to `600`.

**gscmd** Specifies the name of the Ghostscript command. Defaults to `gswin64c` (on Windows) or `gs` (Linux, Mac). Note that any spaces in this option must be replaced by `~`.

**gmidentifycmd** Specifies the name of the GraphicsMagick `identify` command. Defaults to `gm~identify`. Note that any spaces in this option must be replaced by `~`.

**gmconvertcmd** Specifies the name of the GraphicsMagick `convert` command. Defaults to `gm~convert`. Note that any spaces in this option must be replaced by `~`.

---

<sup>5</sup><http://www.ctan.org/pkg/epstopdf>

<sup>6</sup><http://www.ctan.org/pkg/auto-pst-pdf>

<sup>7</sup><http://ratnuu.blogspot.de/2007/02/using-psfrag-with-pdflatex-useful.html>

Apart from at package inclusion, options to `autopdf` may be modified with the `\autopdfoptions{options...}` command. This command respects  $\TeX$  “scope”, e.g. so a call to `\autopdfoptions` within a `\begin{figure}...\end{figure}` environment will only affect graphics included for that particular figure.

Once the package is included, `\includegraphics` may be used as normal to include graphics; `autopdf` will perform any conversions as needed on the fly. For this to work, pdf $\LaTeX$  must be run in “shell escape” mode, which requires adding on of the following options to the pdf $\LaTeX$  command line:

- Linux, Mac: `-shell-escape`.
- MiKTeX (Windows): `--enable-write18`.

To include custom  $\LaTeX$  commands in a ( $\LaTeX$ /)EPS graphic, use the `\autopdfinclude...\autopdfendinclude` command:

```
\autopdfinclude
\usepackage{amssymb}
\newcommand{\fdot}{\dot{f}}
\autopdfendinclude
```

All  $\LaTeX$  commands between `\autopdfinclude` and `\autopdfendinclude` will be included in both the parent document and the wrapper  $\LaTeX$  file.

To include `PSfrag` replacements in a ( $\LaTeX$ /)EPS graphic, do not use the `PSfrag` package; instead use the replacement `\autopdfpsfrag` command:

```
\autopdfpsfrag[options...]{fdot}{Frequency derivative $\fdot$}
```

Available options to `\autopdfpsfrag` are:

**texpos** The  $\LaTeX$  text reference point. Defaults to B1.

**epspos** The Postscript text reference point. Defaults to B1.

**scale** Scaling factor. Defaults to 1.

**angle** Extra text rotation, in degrees. Defaults to 0.

**add** If included, add the replacement text to any existing replacement text, instead of replacing it (the default behaviour).

See the `PSfrag` manual for further details. Default values of the `\autopdfpsfrag` may be changed with the `\autopdfpsfoptions{options...}` command.

### 3 Implementation

```
1 <*package>
```

Required packages.

```
2 \RequirePackage{keyval}
3 \RequirePackage{ifthen}
4 \RequirePackage{ifpdf}
5 \RequirePackage{ifplatform}
6 \RequirePackage{graphicx}
```

Check that we’re running with pdf $\LaTeX$ , and that `PSfrag` hasn’t been included.

```

7 \AtBeginDocument{%
8   \ifthenelse{\NOT\boolean{pdf}}{%
9     \PackageError{autopdf}{%
10      This package is designed to work with pdfLaTeX. %
11      Use "pdflatex" instead of "latex" to compile this document%
12    }{}%
13  }{%
14  }%
15  \ifpackageloaded{psfrag}{%
16    \PackageError{autopdf}{%
17      This package is incompatible with the PSfrag package. %
18      Do not \string\usepackage{psfrag} in this document%
19    }{}%
20  }{%
21  }%
22 }

Global constants and variables.
23 \begingroup
24   \@makeother\%
25   \xdef\autopdf@pc{%
26 \endgroup
27 \def\autopdf@eol{^^J}
28 \newwrite\autopdf@write
29 \newlength\autopdf@width
30 \newlength\autopdf@height
31 \newcount\autopdf@width@dpi
32 \newcount\autopdf@height@dpi
33 \newtoks\autopdf@tex@toks

Utility functions, mostly for manipulating TeX token lists and text.
34 \def\autopdf@if#1#2{\ifthenelse{#1}{#2}{}}
35 \def\autopdf@ifelse#1#2#3{\ifthenelse{#1}{#2}{#3}}
36 \def\autopdf@cat#1#2#3{%
37   \toks@={#3}%
38   \edef\autopdf@cat#1#2#3{\the#1\the#2\the#3}%
39   \toks@={\the#1\the#2\the#3}%
40 }%
41 \autopdf@cat#1#2#3
42 }
43 \def\autopdf@ecat#1#2#3{%
44   \edef\autopdf@ecat#1#2#3{\the#1\the#2\the#3}%
45   \toks@={\the#1\the#2\the#3}%
46 }%
47 \autopdf@ecat#1#2#3
48 }
49 \def\autopdf@split#1#2#3#4{%
50   \tempcnta=#3%
51   \edef\autopdf@split#1#2#3#4{\the#1\the#2\the#3\the#4}%
52   \edef\autopdf@split#1#2#3#4{\the#1\the#2\the#3\the#4}%
53   \autopdf@if{\NOT\equal{#4}{}}{%
54     \expandafter\autopdf@split#1#2#3#4\@nil%
55   }{%
56     \edef#1{\autopdf@split#1#2#3#4}%
57     \edef#2{\autopdf@split#1#2#3#4}%

```

```

58 }%
59 \def\autopdf@Split#1#2\@nil{%
60   \autopdf@if{\@tempcnta>0}{%
61     \edef\autopdf@Split@a{\autopdf@Split@a#1}%
62     \edef\autopdf@Split@b{#2}%
63     \advance\@tempcnta\m@ne%
64   }%
65   \autopdf@if{\NOT\equal{#2}{}}{%
66     \expandafter\autopdf@Split#2\@nil%
67   }%
68 }
69 \def\autopdf@First#1#2#3{%
70   \edef\autopdf@First@a{%
71     \autopdf@Split#1\autopdf@First@a#2#3%
72   }
73 \def\autopdf@Last#1#2#3{%
74   \edef\autopdf@Last@a{%
75     \autopdf@Split\autopdf@Last@a#1#2#3%
76   }
77 \def\autopdf@BeforeDot#1.#2\@nil{#1}
78 \def\autopdf@AfterDot#1.#2\@nil{#2}

```

Functions which generate MD5 checksums, and read/write them to/from the log file.

```

79 \def\autopdf@CreateMDF#1#2#3#4{%
80   \autopdf@ifelse{\boolean{autopdf@scale}}{%
81     \def\autopdf@scale@str{true}%
82   }{%
83     \def\autopdf@scale@str{false}%
84   }%
85   \expandafter\edef\expandafter#1{%
86     \pdfmdfivesum{%
87       from=#2,%
88       to=#3,%
89       scale=\autopdf@scale@str,%
90       margin=\the\autopdf@margin,%
91       resolution=\the\autopdf@resolution,%
92       #4%
93     }%
94   }%
95 }
96 \def\autopdf@GetMDF#1{%
97   \expandafter\ifcsname autopdf@mdflist@#1\endcsname%
98   \expandafter\csname autopdf@mdflist@#1\endcsname%
99   \else%
100     \expandafter none%
101   \fi%
102 }
103 \def\autopdf@SetMDF#1#2{%
104   \expandafter\def\csname autopdf@mdflist@#1\endcsname{#2}%
105 }

```

Package options.

```

106 \newboolean{autopdf@showcmds}
107 \newboolean{autopdf@cleanup}

```

```

108 \newboolean{autopdf@scale}
109 \newlength\autopdf@margin
110 \newcount\autopdf@resolution
111 \define@key{autopdf}{from}{%
112   \edef\autopdf@from@default{.\expandafter\autopdf@AfterDot .#1\@nil}%
113 }
114 \define@key{autopdf}{to}{%
115   \edef\autopdf@to@default{.\expandafter\autopdf@AfterDot .#1\@nil}%
116 }
117 \define@key{autopdf}{logfile}{%
118   \edef\autopdf@log@file{\jobname.#1}%
119 }
120 \define@key{autopdf}{nologfile}[]{%
121   \edef\autopdf@log@file{}%
122 }
123 \define@key{autopdf}{showcmds}[true]{%
124   \setboolean{autopdf@showcmds}{#1}%
125 }
126 \define@key{autopdf}{cleanup}[true]{%
127   \setboolean{autopdf@cleanup}{#1}%
128 }
129 \define@key{autopdf}{scale}[true]{%
130   \setboolean{autopdf@scale}{#1}%
131 }
132 \define@key{autopdf}{margin}{%
133   \autopdf@margin=#1%
134 }
135 \define@key{autopdf}{resolution}{%
136   \autopdf@resolution=#1%
137 }
138 \define@key{autopdf}{gscmd}{%
139   \def\autopdf@GS{#1}%
140 }
141 \define@key{autopdf}{gmidentifycmd}{%
142   \def\autopdf@GMIDENTIFY{#1}%
143 }
144 \define@key{autopdf}{gmconvertcmd}{%
145   \def\autopdf@GMCONVERT{#1}%
146 }
147 \AtEndOfPackage{\let\@unprocessedoptions\relax}
148 \def\autopdf@SetOptions#1{%
149   \setkeys{autopdf}{#1}%
150 }
151 \autopdf@SetOptions{%
152   from=eps,to=pdf,%
153   logfile=autopdf_log,%
154   showcmds=false,%
155   cleanup=true,%
156   scale=true,%
157   margin=0pt,%
158   resolution=600,%
159   gmidentifycmd=gm~identify,%
160   gmconvertcmd=gm~convert%
161 }

```

```

162 \autopdf@IfElse{\boolean{windows}}{%
163   \autopdf@SetOptions{gscmd=gswin64c}%
164 }{%
165   \autopdf@SetOptions{gscmd=gs}%
166 }%
167 \edef\autopdf@a{%
168   \noexpand\autopdf@SetOptions{\@optionlist{\@currname.\@current}}%
169 }
170 \autopdf@a
171 \let\autopdfoptions\autopdf@SetOptions

```

Read the log file at the start of processing, and write to it at the end of the document.

```

172 \autopdf@If{\NOT\equal{\autopdf@log@file}{}}{%
173   \InputIfFileExists{\autopdf@log@file}{-}{-}%
174 }
175 \newtoks\autopdf@log@toks
176 \autopdf@log@toks={}
177 \def\autopdf@WriteLog#1{%
178   \autopdf@ECatToks{\global}{\autopdf@log@toks}{#1}%
179 }
180 \AtEndDocument{%
181   \autopdf@If{%
182     \(\NOT\equal{\autopdf@log@file}{})\AND%
183     \(\NOT\equal{\the\autopdf@log@toks}{})\%
184   }{%
185     \immediate\openout\autopdf@write\autopdf@log@file\relax%
186     \immediate\write\autopdf@write{\the\autopdf@log@toks}%
187     \immediate\closeout\autopdf@write%
188   }%
189 }

```

Function which executes external calls to graphics conversion programs.

```

190 \def\autopdf@Execute@diva{=====}
191 \def\autopdf@Execute@divb{ autopdf }
192 \def\autopdf@Execute@divc{-----}
193 \def\autopdf@Execute@divd{-----}
194 \def\autopdf@Execute#1{%
195   \begingroup%
196     \let\\relax%
197     \def~{\space}%
198     \def\AND{&&}%
199     \def\OR{||}%
200     \def\REDIRT0{>}%
201     \def\LEFT{(%}%
202     \def\RIGHT{)%}%
203     \autopdf@IfElse{\boolean{windows}}{%
204       \autopdf@IfElse{\boolean{autopdf@cleanup}}{%
205         \def\DELETE{del~/f~/q}%
206       }{%
207         \def\DELETE{echo}%
208       }%
209       \def\SILENT{1>nul~2>&1}%
210     }%
211     \autopdf@IfElse{\boolean{autopdf@cleanup}}{%

```

```

212     \def\DELETE{rm~-f}%
213   }{%
214     \def\DELETE{echo}%
215   }%
216   \def\SILENT{1>/dev/null~2>&1}%
217 }%
218 \autopdf@IfElse{\boolean{autopdf@showcmds}}{%
219   \immediate\write16{%
220     ^^J%
221     \autopdf@Execute@diva%
222     \autopdf@Execute@divb%
223     \autopdf@Execute@diva%
224   }%
225   \immediate\write16{#1}%
226   \immediate\write16{%
227     \autopdf@Execute@divc%
228     \autopdf@Execute@divd%
229     \autopdf@Execute@divc%
230   }%
231   \immediate\write18{#1}%
232   \immediate\write16{%
233     \autopdf@Execute@divc%
234     \autopdf@Execute@divb%
235     \autopdf@Execute@divc%
236     ^^J%
237   }%
238 }{%
239   \immediate\write18{\LEFT~#1~\RIGHT~\SILENT}%
240 }%
241 \endgroup%
242 }

```

The \autopdfinclude...\autopdfendinclude command.

```

243 \newcount\autopdf@Capture@list@count
244 \autopdf@Capture@list@count=\z@
245 \def\autopdf@Capture#1{%
246   \toks@={#1}%
247   \edef\autopdf@a{\the\toks@}%
248   \expandafter\edef\csname autopdf@Capture@list@%
249     \the\autopdf@Capture@list@count\endcsname{%
250     \expandafter\strip@prefix\meaning\autopdf@a%
251   }%
252   \advance\autopdf@Capture@list@count\@ne%
253   #1%
254 }
255 \long\def\autopdfinclude#1\autopdfendinclude{%
256   \autopdf@Capture{#1}%
257 }

```

The \autopdfpsfrag and \autopdfpsfoptions commands.

```

258 \newboolean{autopdf@PSfrag@add}
259 \define@key{autopdf@PSfrag}{texpos}{%
260   \edef\autopdf@PSfrag@texpos{#1}%
261 }
262 \define@key{autopdf@PSfrag}{epspos}{%

```



```

263 \edef\autopdf@PSfrag@epspos{#1}%
264 }
265 \define@key{autopdf@PSfrag}{scale}{%
266 \edef\autopdf@PSfrag@scale{#1}%
267 }
268 \define@key{autopdf@PSfrag}{angle}{%
269 \edef\autopdf@PSfrag@angle{#1}%
270 }
271 \define@key{autopdf@PSfrag}{add}[true]{%
272 \setboolean{autopdf@PSfrag@add}{#1}%
273 }
274 \def\autopdf@SetPSfragOptions#1{%
275 \setkeys{autopdf@PSfrag}{#1}%
276 }%
277 \autopdf@SetPSfragOptions{%
278 texpos=B1,epspos=B1,%
279 scale=1,angle=0,%
280 }
281 \def\autopdf@PSfrag{%
282 \@ifnextchar[{%
283 \autopdf@@PSfrag%
284 }{%
285 \autopdf@@PSfrag[]%
286 }%
287 }
288 \def\autopdf@@PSfrag[#1]#2#3{%
289 \begingroup%
290 \setkeys{autopdf@PSfrag}{#1}%
291 \def\autopdf@PSfrag@tag{#2}%
292 \def\autopdf@PSfrag@tex{#3}%
293 \xdef\autopdf@PSfrag@cmd{%
294 \string\psfrag\ifautopdf@PSfrag@add*\fi%
295 {\expandafter\strip@prefix\meaning\autopdf@PSfrag@tag}%
296 [\autopdf@PSfrag@texpos] [\autopdf@PSfrag@epspos]%
297 [\autopdf@PSfrag@scale] [\autopdf@PSfrag@angle]%
298 {\expandafter\strip@prefix\meaning\autopdf@PSfrag@tex}%
299 }%
300 \endgroup%
301 \expandafter\let\csname autopdf@Capture@list@%
302 \the\autopdf@Capture@list@count\endcsname\autopdf@PSfrag@cmd%
303 \advance\autopdf@Capture@list@count\@ne%
304 }
305 \let\autopdfpsfrag\autopdf@PSfrag
306 \let\autopdfpsfoptions\autopdf@SetPSfragOptions

Get the types of graphics files from their extensions, or use the defaults.
307 \def\autopdf@GetGrType#1#2{%
308 \@ifundefined{Gin@rule@#2}{%
309 \edef#1{\expandafter\autopdf@AfterDot #2\@nil}%
310 }{%
311 \def\autopdf@GetGrType@a{%
312 \edef#1{%
313 \expandafter\expandafter\expandafter%
314 \autopdf@@@GetGrType\csname Gin@rule@#2\endcsname}%
315 }%

```

```

316     }%
317     \autopdf@GetGrType@a%
318 }%
319 }
320 \def\autopdf@@GetGrType#1#2#3{%
321   #1%
322 }

Determine the size of a graphic, either from graphicx (for EPS files) or by running
the GraphicsMagick identify command (for other formats).
323 \def\autopdf@ReadGrSize#1#2{%
324   \begingroup%
325     \ifGin@bbox%
326     \else%
327       \autopdf@GetGrType\autopdf@type{#2}%
328       \ifundefined{Gread@\autopdf@type}{%
329         \autopdf@@ReadGrSize@Other{#1}{#2}%
330       }{%
331         \csname Gread@\autopdf@type\endcsname{#1#2}%
332       }%
333     \fi%
334     \Gin@viewport@code%
335     \Gin@nat@width=\Gin@urx bp%
336     \advance\Gin@nat@width-\Gin@llx bp%
337     \Gin@nat@height=\Gin@ury bp%
338     \advance\Gin@nat@height-\Gin@lly bp%
339     \Gin@req@sizes%
340     \autopdf@IfElse{\boolean{autopdf@scale}}{%
341       \global\autopdf@width=\Gin@req@width%
342       \global\autopdf@height=\Gin@req@height%
343     }{%
344       \global\autopdf@width=\Gin@nat@width%
345       \global\autopdf@height=\Gin@nat@height%
346     }%
347     \global\autopdf@width@dpi=\expandafter%
348       \autopdf@BeforeDot\the\autopdf@width\@nil%
349     \global\autopdf@height@dpi=\expandafter%
350       \autopdf@BeforeDot\the\autopdf@height\@nil%
351     \global\multiply\autopdf@width@dpi by \autopdf@resolution%
352     \global\multiply\autopdf@height@dpi by \autopdf@resolution%
353     \global\divide\autopdf@width@dpi by 72%
354     \global\divide\autopdf@height@dpi by 72%
355   \endgroup%
356 }
357 \def\autopdf@@ReadGrSize@Other#1#2{%
358   \autopdf@Execute{%
359     \autopdf@GMIDENTITY~%
360     -units~PixelsPerInch~%
361     -format~"%
362       \\\def\\width{\autopdf@pc[fx:w/image.resolution.x*72]}%
363       \\\def\\height{\autopdf@pc[fx:h/image.resolution.y*72]}%
364       "~%
365       #1#2~\REDIRTO~#1.size~%
366       \OR~\DELETE~#1.size~%
367   }%

```

```

368 \IfFileExists{#1.size}{%
369   \def\Gin@llx{0}\def\Gin@lly{0}%
370   \begingroup%
371     \input{#1.size}%
372     \edef\autopdf@a{%
373       \def\noexpand\Gin@urx{\width}%
374       \def\noexpand\Gin@ury{\height}%
375     }%
376     \expandafter%
377     \endgroup\autopdf@a%
378     \autopdf@Execute{\DELETE~#1.size}%
379 }{%
380   \PackageError{autopdf}{%
381     Could not determine size of "#1#2"%
382   }{%
383   }%
384 }

```

Replace the internal `graphicx` command `\Gininclude@graphics` with a new version, which performs any required graphics conversions before inclusion.

```

385 \def\autopdf@IncludeGraphics#1{%
386   \edef\autopdf@to{%
387     \begingroup%
388       \let\to\relax%
389       \expandafter%
390     \endgroup%
391     \autopdf@@IncludeGraphics#1\to\to\@nil%
392   }
393   \def\autopdf@@IncludeGraphics#1\to#2\to#3\@nil{%
394     \edef\autopdf@to{#2}%
395     \begingroup%
396       \let\input@path\Ginput@path%
397       \filename@parse{#1}%
398       \edef\autopdf@dir{\filename@area}%
399       \autopdf@if{\equal{\autopdf@dir}{}}{%
400         \edef\autopdf@dir{\@currdir}%
401       }%
402       \ifx\filename@ext\relax%
403         \edef\autopdf@from{\autopdf@from@default}%
404       \else%
405         \edef\autopdf@from{\Gin@sepdefault\filename@ext}%
406       \fi%
407       \Gin@getbase{\autopdf@from}%
408       \ifx\Gin@ext\relax%
409         \PackageError{autopdf}{%
410           File "#1\autopdf@from" could not be found%
411         }{%
412         }%
413       \else%
414         \edef\autopdf@base{\Gin@base}%
415         \autopdf@if{\equal{\autopdf@to}{}}{%
416           \edef\autopdf@to{\autopdf@to@default}%
417         }%
418         \let\autopdf@Gin@setfile\relax%
419         \@ifundefined{Gin@rule@\autopdf@to}{%
420           \@ifundefined{Gin@rule@*}{%

```

```

420     \PackageError{autopdf}{%
421       Graphics extension "\autopdf@to" is not supported%
422     }{}%
423   }{%
424     \def\autopdf@Gin@setfile{%
425       \expandafter\expandafter\expandafter\Gin@setfile%
426       \csname Gin@rule@*\endcsname{\autopdf@base\autopdf@to}%
427     }%
428   }%
429 }{%
430   \def\autopdf@Gin@setfile{%
431     \expandafter\expandafter\expandafter\Gin@setfile%
432     \csname Gin@rule@\autopdf@to\endcsname{%
433       \autopdf@base\autopdf@to%
434     }%
435   }%
436 }%
437 \autopdf@ConvertGraphics{\autopdf@from}{\autopdf@to}%
438 \IfFileExists{\autopdf@base\autopdf@to}{%
439 }{%
440   \PackageError{autopdf}{%
441     Could not convert %
442     "\autopdf@base\autopdf@mid" to "\autopdf@base#2". %
443     See "\autopdf@base.autopdf.log" for details%
444   }{}%
445 }%
446   \autopdf@Gin@setfile%
447 \fi%
448 \endgroup%
449 }
450 \let\Gininclude@graphics\autopdf@IncludeGraphics

```

Perform the graphics conversions. For (L<sup>A</sup>T<sub>E</sub>X)/EPS or PostScript graphics, generates the wrapper L<sup>A</sup>T<sub>E</sub>X file and converts to PostScript, then calls either Ghostscript (PS to PDF) or GraphicsMagick (other combinations) to convert to the final output format. For other graphics formats, call GraphicsMagick only.

```

451 \def\autopdf@ConvertGraphics#1#2{%
452   \autopdf@if{\NOT\equal{#1}{#2}}{%
453     \autopdf@WriteLog{%
454       \autopdf@pc in \autopdf@base#1\autopdf@eol%
455     }%
456     \autopdf@GetGrType\autopdf@from@type{#1}%
457     \autopdf@GetGrType\autopdf@to@type{#2}%
458     \autopdf@ifElse{\equal{\autopdf@from@type}{eps}}{%
459       \autopdf@ReadGrSize{\autopdf@base}{#1}%
460       \autopdf@EPSToPSTeX{#1}%
461       \autopdf@CreateMDF\autopdf@Graphics@mdfa{#1}{#2}{%
462         \pdfmdfivesum{\the\autopdf@tex@toks}%
463       }%
464     }{%
465       \autopdf@ifElse{\equal{\autopdf@from@type}{tex}}{%
466         \newboolean{autopdf@scale@old}%
467         \autopdf@ifElse{\boolean{autopdf@scale}}{%
468           \setboolean{autopdf@scale@old}{true}%

```

```

469     }{%
470         \setboolean{autopdf@scale@old}{false}%
471     }%
472     \setboolean{autopdf@scale}{false}%
473     \autopdf@ReadGrSize{\autopdf@base}{.eps}%
474     \autopdf@EPSToPSTeX{#1}%
475     \autopdf@CreateMDF\autopdf@Graphics@mdfa{#1}{#2}{%
476         \pdfmdfivesum file{\autopdf@base.tex}%
477         \pdfmdfivesum file{\autopdf@base.eps}%
478         \pdfmdfivesum{\the\autopdf@tex@toks}%
479     }%
480     \autopdf@IfElse{\boolean{autopdf@scale@old}}{%
481         \setboolean{autopdf@scale}{true}%
482     }{%
483         \setboolean{autopdf@scale}{false}%
484     }%
485 }{%
486     \autopdf@CreateMDF\autopdf@Graphics@mdfa{#1}{#2}{%
487         \pdfmdfivesum file{\autopdf@base#1}%
488     }%
489 }%
490 }%
491 \edef\autopdf@Graphics@mdfb{\autopdf@GetMDF{\autopdf@base#1}}%
492 \IfFileExists{\autopdf@base#2}{%
493 }{%
494     \edef\autopdf@Graphics@mdfb{rebuild}%
495 }%
496 \autopdf@If{\NOT\(%
497     \pdfstrcmp{\autopdf@Graphics@mdfa}{\autopdf@Graphics@mdfb}=0%
498 \)}{%
499     \autopdf@If{\NOT\(\pdfshellescape=1\)}{%
500         \PackageError{autopdf}{%
501             This package requires pdfLaTeX to %
502             be running in "shell escape" mode%
503         }{}%
504     }%
505     \autopdf@IfElse{%
506         \equal{\autopdf@from@type}{eps}%
507         \OR\equal{\autopdf@from@type}{tex}%
508     }{%
509         \autopdf@EPSToPS%
510         \IfFileExists{\autopdf@base.ps}{%
511             }{%
512                 \PackageError{autopdf}{%
513                     Could not convert %
514                     "\autopdf@base#1" to "\autopdf@base.ps". %
515                     See "\autopdf@base.autopdf.log" for details%
516                 }{}%
517             }%
518         \def\autopdf@mid{.ps}%
519     }{%
520         \autopdf@ReadGrSize{\autopdf@base}{#1}%
521         \def\autopdf@mid{#1}%
522     }%

```

```

523 \autopdf@if{\NOT\equal{\autopdf@mid}{#2}}{%
524 \autopdf@ifelse{\equal{\autopdf@mid}{.ps}\AND\equal{#2}{.pdf}}{%
525 \autopdf@PSToPDF%
526 }{%
527 \autopdf@Convert{\autopdf@mid}{#2}%
528 }%
529 }%
530 \iffileexists{\autopdf@base#2}{%
531 \autopdf@if{%
532 \NOT\(\equal{\autopdf@mid}{#1}\OR\equal{\autopdf@mid}{#2}\)%
533 }{%
534 \autopdf@Execute{\DELETE~\autopdf@base\autopdf@mid}%
535 }%
536 }{%
537 \PackageError{autopdf}{%
538 Could not convert %
539 "\autopdf@base\autopdf@mid" to "\autopdf@base#2". %
540 See "\autopdf@base.autopdf.log" for details%
541 }{}%
542 }%
543 }%
544 \autopdf@WriteLog{%
545 \string\autopdf@SetMDF{\autopdf@base#1}%
546 {\autopdf@Graphics@mdfa}\autopdf@eol%
547 \autopdf@pc out \autopdf@base#2\autopdf@eol%
548 }%
549 }%
550 }

```

Generates the wrapper L<sup>A</sup>T<sub>E</sub>X file for (L<sup>A</sup>T<sub>E</sub>X)/EPS or PostScript graphics.

```

551 \def\autopdf@EPSToPSTeX#1{%
552 \begingroup%
553 \global\autopdf@tex@toks={}%
554 \autopdf@ECatToks{\global}{\autopdf@tex@toks}{%
555 \string\documentclass{minimal}\autopdf@eol%
556 \string\usepackage[%
557 paperwidth=\the\autopdf@width,%
558 paperheight=\the\autopdf@height,%
559 margin=\the\autopdf@margin,%
560 offset=0pt,%
561 bindingoffset=0pt,%
562 noheadfoot,%
563 nomarginpar%
564 ]{geometry}\autopdf@eol%
565 \string\usepackage{graphicx}\autopdf@eol%
566 \string\usepackage{psfrag}\autopdf@eol%
567 \string\pagestyle{empty}\autopdf@eol%
568 \string\setlength{\string\parindent}{0pt}\autopdf@eol%
569 \string\setlength{\string\parskip}{0pt}\autopdf@eol%
570 }%
571 \def\autopdf@fonts{%
572 \tiny,\scriptsize,\footnotesize,\small,%
573 \normalsize,\large,\Large,\LARGE,\huge,\Huge%
574 }%
575 \@for\autopdf@a:=\autopdf@fonts\do{%

```

```

576     \begingroup%
577     \autopdf@a%
578     \autopdf@ECatToks{\global}{\autopdf@tex@toks}{%
579         \string\def\expandafter\string\autopdf@a{%
580             \string\fontencoding{\f@encoding}%
581             \string\fontfamily{\f@family}%
582             \string\fontseries{\f@series}%
583             \string\fontshape{\f@shape}%
584             \string\fontsize{\f@size}{\f@baselineskip}%
585             \string\selectfont%
586         }\autopdf@eol%
587     }%
588     \endgroup%
589 }%
590 \autopdf@ECatToks{\global}{\autopdf@tex@toks}{%
591     \string\normalsize\autopdf@eol%
592     \string\makeatletter\autopdf@eol%
593 }%
594 \count@=\z@%
595 \loop\ifnum\count@<\autopdf@Capture@list@count\relax%
596     \autopdf@ECatToks{\global}{\autopdf@tex@toks}{%
597         \string\def\string\autopdf@act{%
598             \csname autopdf@Capture@list@the\count@\endcsname%
599         }%
600         \string\autopdf@act\autopdf@eol%
601     }%
602     \advance\count@\@ne%
603 \repeat%
604 \autopdf@ECatToks{\global}{\autopdf@tex@toks}{%
605     \string\def\string\autopdf@act{\autopdf@eol%
606         \string\makeatother\autopdf@eol%
607         \string\begin{document}\autopdf@eol%
608     }%
609 \autopdf@IfElse{\equal{#1}{.tex}}{%
610     \autopdf@ECatToks{\global}{\autopdf@tex@toks}{%
611         \string\input{\autopdf@base#1}%
612     }%
613 }{%
614     \autopdf@ECatToks{\global}{\autopdf@tex@toks}{%
615         \string\includegraphics[%
616             width=0.99\string\textwidth,height=0.99\string\textheight%
617         ]{\autopdf@base#1}%
618     }%
619 }%
620 \autopdf@ECatToks{\global}{\autopdf@tex@toks}{%
621     \autopdf@pc\pdfmdfivesum file{\autopdf@base#1}\autopdf@eol%
622     \string\end{document}%
623 }%
624 \endgroup%
625 }

Calls latex and dvips to convert (LATEX)/EPS graphics to PostScript.
626 \def\autopdf@EPSToPS{%
627     \immediate\openout\autopdf@write \autopdf@base.autopdf.tex\relax%
628     \immediate\write\autopdf@write{\the\autopdf@tex@toks}%

```

```

629 \immediate\closeout\autopdf@write%
630 \autopdf@Execute{%
631   \LEFT~%
632     latex~%
633     -interaction=nonstopmode~%
634     -output-format=dvi~%
635     -aux-directory=\autopdf@dir~%
636     -output-directory=\autopdf@dir~%
637     \autopdf@base.autopdf.tex~%
638   \AND~%
639   dvips~%
640     -o~\autopdf@base.ps~%
641     \autopdf@base.autopdf.dvi~%
642   \AND~%
643     \DELETE~%
644     \autopdf@base.autopdf.tex~\autopdf@base.autopdf.aux~%
645     \autopdf@base.autopdf.log~\autopdf@base.autopdf.dvi~%
646   \RIGHT~%
647   \OR~%
648     \DELETE~\autopdf@base.ps%
649 }%
650 }

Calls Ghostscript to convert PostScript graphics to PDF.
651 \def\autopdf@PSToPDF{%
652   \autopdf@Execute{%
653     \autopdf@GS~%
654     -dSAFER~-dBATCH~-dNOPAUSE~-q~%
655     -sDEVICE=pdfwrite~-dCompatibilityLevel=1.4~%
656     -dAutoRotatePages="/None"~%
657     -sOutputFile=\autopdf@base.pdf~%
658     -c~.setpdfwrite~-f~\autopdf@base.ps~%
659   \OR~%
660     \DELETE~\autopdf@base.pdf~%
661 }%
662 }

Calls the GraphicsMagick convert command.
663 \def\autopdf@Convert#1#2{%
664   \autopdf@Execute{%
665     \LEFT~%
666     \autopdf@GMCONVERT~%
667     -units~PixelsPerInch~%
668     -density~\the\autopdf@resolution~%
669     \autopdf@base#1~%
670     -resize~\the\autopdf@width@dpi x\the\autopdf@height@dpi~%
671     \autopdf@base#2~%
672   \AND~%
673     identify~\autopdf@base#2~%
674   \RIGHT~%
675   \OR~%
676     \DELETE~\autopdf@base#2~%
677 }%
678 }
679 \end{package}

```