The **autopdf** package[*]

Karl Wette

December 10, 2013

1 Introduction

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

- It uses Ghostscript[1] and GraphicsMagick[2] 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 pdfLaTeX processes the document. For this to work, pdfLaTeX 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[3].
- After conversion, MD5 checksums of each input graphic and any associated files are stored. When pdfLaTeX 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[4], 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

---

[*] This document corresponds to autopdf v1.1, dated 2013/12/10.

[1] `http://www.ghostscript.com/`

[2] `http://www.graphicsmagick.org/`

[3] `http://www.ctan.org/pkg/psfrag`

[4] `http://www.ctan.org/`

autopdf were inspired by the features provided by the epstopdf[5] and auto-pst-pdf[6] packages, and the fragmaster.pl[7] 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 pdfLaTeX 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 (LaTeX/)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 (LaTeX/)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 `~`.

---

[5] `http://www.ctan.org/pkg/epstopdf`

[6] `http://www.ctan.org/pkg/auto-pst-pdf`

[7] `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, pdfLATEX must be run in "shell escape" mode, which requires adding on of the following options to the pdfLATEX 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 `Bl`.

epspos The Postscript text reference point. Defaults to `Bl`.

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

```
⟨*package⟩
```

Required packages.

```
\RequirePackage{keyval}
\RequirePackage{ifthen}
\RequirePackage{ifpdf}
\RequirePackage{ifplatform}
\RequirePackage{graphicx}
```

Check that we're running with pdfLATEX, and that PSfrag hasn't been included.

```
\AtBeginDocument{%
  \ifthenelse{\NOT\boolean{pdf}}{%
    \PackageError{autopdf}{%
      This package is designed to work with pdfLaTeX. %
      Use "pdflatex" instead of "latex" to compile this document%
    }{}%
  }{%
  }%
  \@ifpackageloaded{psfrag}{%
    \PackageError{autopdf}{%
      This package is incompatible with the PSfrag package. %
      Do not \string\usepackage{psfrag} in this document%
    }{}%
  }{%
  }%
}
```

Global constants and variables.

```
\begingroup
  \@makeother\%
  \xdef\autopdf@pc{%}
\endgroup
\def\autopdf@eol{^^J}
\newwrite\autopdf@write
\newlength\autopdf@width
\newlength\autopdf@height
\newcount\autopdf@width@dpi
\newcount\autopdf@height@dpi
\newtoks\autopdf@tex@toks
```

Utility functions, mostly for manipulating TEX token lists and text.

```
\def\autopdf@If#1#2{\ifthenelse{#1}{#2}{}}
\def\autopdf@IfElse#1#2#3{\ifthenelse{#1}{#2}{#3}}
\def\autopdf@CatToks#1#2#3{%
  \toks@={#3}%
  \edef\autopdf@CatToks@a{%
    #1#2={\the#2\the\toks@}%
  }%
  \autopdf@CatToks@a%
}
\def\autopdf@ECatToks#1#2#3{%
  \edef\autopdf@ECatToks@a{%
    #1#2={\the#2#3}%
  }%
  \autopdf@ECatToks@a%
}
\def\autopdf@Split#1#2#3#4{%
  \@tempcnta#3%
  \edef\autopdf@Split@a{}%
  \edef\autopdf@Split@b{#4}%
  \autopdf@If{\NOT\equal{#4}{}}{%
    \expandafter\autopdf@@Split#4\@nil%
  }%
  \edef#1{\autopdf@Split@a}%
  \edef#2{\autopdf@Split@b}%
}%
\def\autopdf@@Split#1#2\@nil{%
  \autopdf@If{\@tempcnta>0}{%
    \edef\autopdf@Split@a{\autopdf@Split@a#1}%
    \edef\autopdf@Split@b{#2}%
    \advance\@tempcnta\m@ne%
  }%
  \autopdf@If{\NOT\equal{#2}{}}{%
    \expandafter\autopdf@@Split#2\@nil%
  }%
}
\def\autopdf@First#1#2#3{%
  \edef\autopdf@First@a{}%
  \autopdf@Split#1\autopdf@First@a#2#3%
}
\def\autopdf@Last#1#2#3{%
  \edef\autopdf@Last@a{}%
  \autopdf@Split\autopdf@Last@a#1#2#3%
}
\def\autopdf@BeforeDot#1.#2\@nil{#1}
\def\autopdf@AfterDot#1.#2\@nil{#2}
```

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

```
\def\autopdf@CreateMDF#1#2#3#4{%
  \autopdf@IfElse{\boolean{autopdf@scale}}{%
    \def\autopdf@scale@str{true}%
  }{%
    \def\autopdf@scale@str{false}%
  }%
  \expandafter\edef\expandafter#1{%
    \pdfmdfivesum{%
      from=#2,%
      to=#3,%
      scale=\autopdf@scale@str,%
      margin=\the\autopdf@margin,%
      resolution=\the\autopdf@resolution,%
      #4%
    }%
  }%
}
\def\autopdf@GetMDF#1{%
  \expandafter\ifcsname autopdf@mdflist@#1\endcsname%
    \expandafter\csname autopdf@mdflist@#1\endcsname%
  \else%
    \expandafter none%
  \fi%
}
\def\autopdf@SetMDF#1#2{%
  \expandafter\def\csname autopdf@mdflist@#1\endcsname{#2}%
}
```

Package options.

```
\newboolean{autopdf@showcmds}
\newboolean{autopdf@cleanup}
```

```
\newboolean{autopdf@scale}
\newlength\autopdf@margin
\newcount\autopdf@resolution
\define@key{autopdf}{from}{%
  \edef\autopdf@from@default{.\expandafter\autopdf@AfterDot .#1\@nil}%
}
\define@key{autopdf}{to}{%
  \edef\autopdf@to@default{.\expandafter\autopdf@AfterDot .#1\@nil}%
}
\define@key{autopdf}{logfile}{%
  \edef\autopdf@log@file{\jobname.#1}%
}
\define@key{autopdf}{nologfile}[]{%
  \edef\autopdf@log@file{}%
}
\define@key{autopdf}{showcmds}[true]{%
  \setboolean{autopdf@showcmds}{#1}%
}
\define@key{autopdf}{cleanup}[true]{%
  \setboolean{autopdf@cleanup}{#1}%
}
\define@key{autopdf}{scale}[true]{%
  \setboolean{autopdf@scale}{#1}%
}
\define@key{autopdf}{margin}{%
  \autopdf@margin=#1%
}
\define@key{autopdf}{resolution}{%
  \autopdf@resolution=#1%
}
\define@key{autopdf}{gscmd}{%
  \def\autopdf@GS{#1}%
}
\define@key{autopdf}{gmidentifycmd}{%
  \def\autopdf@GMIDENTIFY{#1}%
}
\define@key{autopdf}{gmconvertcmd}{%
  \def\autopdf@GMCONVERT{#1}%
}
\AtEndOfPackage{\let\@unprocessedoptions\relax}
\def\autopdf@SetOptions#1{%
  \setkeys{autopdf}{#1}%
}
\autopdf@SetOptions{%
  from=eps,to=pdf,%
  logfile=autopdf_log,%
  showcmds=false,%
  cleanup=true,%
  scale=true,%
  margin=0pt,%
  resolution=600,%
  gmidentifycmd=gm~identify,%
  gmconvertcmd=gm~convert%
}
```

```
\autopdf@IfElse{\boolean{windows}}{%
  \autopdf@SetOptions{gscmd=gswin64c}%
}{%
  \autopdf@SetOptions{gscmd=gs}%
}%
\edef\autopdf@a{%
  \noexpand\autopdf@SetOptions{\@ptionlist{\@currname.\@currext}}%
}
\autopdf@a
\let\autopdfoptions\autopdf@SetOptions
```

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

```
\autopdf@If{\NOT\equal{\autopdf@log@file}{}}{%
  \InputIfFileExists{\autopdf@log@file}{}{}%
}
\newtoks\autopdf@log@toks
\autopdf@log@toks={}
\def\autopdf@WriteLog#1{%
  \autopdf@ECatToks{\global}{\autopdf@log@toks}{#1}%
}
\AtEndDocument{%
  \autopdf@If{%
    \(\NOT\equal{\autopdf@log@file}{}\)\AND%
    \(\NOT\equal{\the\autopdf@log@toks}{}\)%
  }{%
    \immediate\openout\autopdf@write\autopdf@log@file\relax%
    \immediate\write\autopdf@write{\the\autopdf@log@toks}%
    \immediate\closeout\autopdf@write%
  }%
}
```

Function which executes external calls to graphics conversion programs.

```
\def\autopdf@Execute@diva{================================}
\def\autopdf@Execute@divb{ autopdf }
\def\autopdf@Execute@divc{--------------------------------}
\def\autopdf@Execute@divd{---------}
\def\autopdf@Execute#1{%
  \begingroup%
    \let\\\relax%
    \def~{\space}%
    \def\AND{&&}%
    \def\OR{||}%
    \def\REDIRTO{>}%
    \def\LEFT{(}%
    \def\RIGHT{)}%
    \autopdf@IfElse{\boolean{windows}}{%
      \autopdf@IfElse{\boolean{autopdf@cleanup}}{%
        \def\DELETE{del~/f~/q}%
      }{%
        \def\DELETE{echo}%
      }%
      \def\SILENT{1>nul~2>&1}%
    }{%
      \autopdf@IfElse{\boolean{autopdf@cleanup}}{%
```

```
        \def\DELETE{rm~-f}%
      }{%
        \def\DELETE{echo}%
      }%
      \def\SILENT{1>/dev/null~2>&1}%
    }%
    \autopdf@IfElse{\boolean{autopdf@showcmds}}{%
      \immediate\write16{%
        ^^J%
        \autopdf@Execute@diva%
        \autopdf@Execute@divb%
        \autopdf@Execute@diva%
      }%
      \immediate\write16{#1}%
      \immediate\write16{%
        \autopdf@Execute@divc%
        \autopdf@Execute@divd%
        \autopdf@Execute@divc%
      }%
      \immediate\write18{#1}%
      \immediate\write16{%
        \autopdf@Execute@divc%
        \autopdf@Execute@divb%
        \autopdf@Execute@divc%
        ^^J%
      }%
    }{%
      \immediate\write18{\LEFT~#1~\RIGHT~\SILENT}%
    }%
  \endgroup%
}
```

The `\autopdfinclude...\autopdfendinclude` command.

```
\newcount\autopdf@Capture@list@count
\autopdf@Capture@list@count=\z@
\def\autopdf@Capture#1{%
  \toks@={#1}%
  \edef\autopdf@a{\the\toks@}%
  \expandafter\edef\csname autopdf@Capture@list@%
    \the\autopdf@Capture@list@count\endcsname{%
      \expandafter\strip@prefix\meaning\autopdf@a%
    }%
  \advance\autopdf@Capture@list@count\@ne%
  #1%
}
\long\def\autopdfinclude#1\autopdfendinclude{%
  \autopdf@Capture{#1}%
}
```

The `\autopdfpsfrag` and `\autopdfpsfoptions` commands.

```
\newboolean{autopdf@PSfrag@add}
\define@key{autopdf@PSfrag}{texpos}{%
  \edef\autopdf@PSfrag@texpos{#1}%
}
\define@key{autopdf@PSfrag}{epspos}{%
```

```
  \edef\autopdf@PSfrag@epspos{#1}%
}
\define@key{autopdf@PSfrag}{scale}{%
  \edef\autopdf@PSfrag@scale{#1}%
}
\define@key{autopdf@PSfrag}{angle}{%
  \edef\autopdf@PSfrag@angle{#1}%
}
\define@key{autopdf@PSfrag}{add}[true]{%
  \setboolean{autopdf@PSfrag@add}{#1}%
}
\def\autopdf@SetPSfragOptions#1{%
  \setkeys{autopdf@PSfrag}{#1}%
}%
\autopdf@SetPSfragOptions{%
  texpos=Bl,epspos=Bl,%
  scale=1,angle=0,%
}
\def\autopdf@PSfrag{%
  \@ifnextchar[{%
    \autopdf@@PSfrag%
  }{%
    \autopdf@@PSfrag[]%
  }%
}
\def\autopdf@@PSfrag[#1]#2#3{%
  \begingroup%
    \setkeys{autopdf@PSfrag}{#1}%
    \def\autopdf@PSfrag@tag{#2}%
    \def\autopdf@PSfrag@tex{#3}%
    \xdef\autopdf@PSfrag@cmd{%
      \string\psfrag\ifautopdf@PSfrag@add*\fi%
      {\expandafter\strip@prefix\meaning\autopdf@PSfrag@tag}%
      [\autopdf@PSfrag@texpos][\autopdf@PSfrag@epspos]%
      [\autopdf@PSfrag@scale][\autopdf@PSfrag@angle]%
      {\expandafter\strip@prefix\meaning\autopdf@PSfrag@tex}%
    }%
  \endgroup%
  \expandafter\let\csname autopdf@Capture@list@%
    \the\autopdf@Capture@list@count\endcsname\autopdf@PSfrag@cmd%
  \advance\autopdf@Capture@list@count\@ne%
}
\let\autopdfpsfrag\autopdf@PSfrag
\let\autopdfpsfoptions\autopdf@SetPSfragOptions
```

Get the types of graphics files from their extensions, or use the defaults.

```
\def\autopdf@GetGrType#1#2{%
  \@ifundefined{Gin@rule@#2}{%
    \edef#1{\expandafter\autopdf@AfterDot #2\@nil}%
  }{%
    \def\autopdf@GetGrType@a{%
      \edef#1{%
        \expandafter\expandafter\expandafter%
        \autopdf@@GetGrType\csname Gin@rule@#2\endcsname{}%
      }%
    }%
    \autopdf@GetGrType@a%
  }%
}
\def\autopdf@@GetGrType#1#2#3{%
  #1%
}
```

Determine the size of a graphic, either from graphicx (for EPS files) or by running the GraphicsMagick `identify` command (for other formats).

```
\def\autopdf@ReadGrSize#1#2{%
  \begingroup%
    \ifGin@bbox%
    \else%
      \autopdf@GetGrType\autopdf@type{#2}%
      \autopdf@Execute{%
        \autopdf@GMIDENTIFY~%
        -units~PixelsPerInch~%
        -format~"%
          \\\\def\\\\width{\autopdf@pc[fx:w/image.resolution.x*72]}%
          \\\\def\\\\height{\autopdf@pc[fx:h/image.resolution.y*72]}%
          "~%
        #1#2~\REDIRTO~#1.size~%
        \OR~\DELETE~#1.size~%
      }%
      \IfFileExists{#1.size}{%
        \def\Gin@llx{0}\def\Gin@lly{0}%
        \begingroup%
          \input{#1.size}%
          \edef\autopdf@a{%
            \def\noexpand\Gin@urx{\width}%
            \def\noexpand\Gin@ury{\height}%
          }%
          \expandafter%
        \endgroup\autopdf@a%
        \autopdf@Execute{\DELETE~#1.size}%
      }{%
        \PackageError{autopdf}{%
          Could not determine size of "#1#2"%
        }{}%
      }%
    \fi%
    \Gin@viewport@code%
    \Gin@nat@width=\Gin@urx bp%
    \advance\Gin@nat@width-\Gin@llx bp%
    \Gin@nat@height=\Gin@ury bp%
    \advance\Gin@nat@height-\Gin@lly bp%
    \Gin@req@sizes%
    \autopdf@IfElse{\boolean{autopdf@scale}}{%
      \global\autopdf@width=\Gin@req@width%
      \global\autopdf@height=\Gin@req@height%
    }{%
      \global\autopdf@width=\Gin@nat@width%
      \global\autopdf@height=\Gin@nat@height%
    }%
    \global\autopdf@width@dpi=\expandafter%
      \autopdf@BeforeDot\the\autopdf@width\@nil%
    \global\autopdf@height@dpi=\expandafter%
      \autopdf@BeforeDot\the\autopdf@height\@nil%
    \global\multiply\autopdf@width@dpi by \autopdf@resolution%
    \global\multiply\autopdf@height@dpi by \autopdf@resolution%
    \global\divide\autopdf@width@dpi by 72%
    \global\divide\autopdf@height@dpi by 72%
  \endgroup%
}
```

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

```
\def\autopdf@IncludeGraphics#1{%
  \edef\autopdf@to{}%
  \begingroup%
    \let\to\relax%
    \expandafter%
  \endgroup%
  \autopdf@@IncludeGraphics#1\to\to\@nil%
}
\def\autopdf@@IncludeGraphics#1\to#2\to#3\@nil{%
  \edef\autopdf@to{#2}%
  \begingroup%
    \let\input@path\Ginput@path%
    \filename@parse{#1}%
    \edef\autopdf@dir{\filename@area}%
    \autopdf@If{\equal{\autopdf@dir}{}}{%
      \edef\autopdf@dir{\@currdir}%
    }%
    \ifx\filename@ext\relax%
      \edef\autopdf@from{\autopdf@from@default}%
    \else%
      \edef\autopdf@from{\Gin@sepdefault\filename@ext}%
    \fi%
    \Gin@getbase{\autopdf@from}%
    \ifx\Gin@ext\relax%
      \PackageError{autopdf}{%
        File "#1\autopdf@from" could not be found%
      }{}%
    \else%
      \edef\autopdf@base{\Gin@base}%
      \autopdf@If{\equal{\autopdf@to}{}}{%
        \edef\autopdf@to{\autopdf@to@default}%
      }%
      \let\autopdf@Gin@setfile\relax%
      \@ifundefined{Gin@rule@\autopdf@to}{%
        \@ifundefined{Gin@rule@*}{%
          \PackageError{autopdf}{%
            Graphics extension "\autopdf@to" is not supported%
          }{}%
        }{%
          \def\autopdf@Gin@setfile{%
            \expandafter\expandafter\expandafter\Gin@setfile%
            \csname Gin@rule@*\endcsname{\autopdf@base\autopdf@to}%
```

```
     }%
    }%
   }{%
    \def\autopdf@Gin@setfile{%
     \expandafter\expandafter\expandafter\Gin@setfile%
      \csname Gin@rule@\autopdf@to\endcsname{%
       \autopdf@base\autopdf@to%
      }%
    }%
   }%
   \autopdf@ConvertGraphics{\autopdf@from}{\autopdf@to}%
   \IfFileExists{\autopdf@base\autopdf@to}{%
   }{%
    \PackageError{autopdf}{%
     Could not convert %
     "\autopdf@base\autopdf@mid" to "\autopdf@base#2". %
     See "\autopdf@base.autopdf.log" for details%
    }{}%
   }%
   \autopdf@Gin@setfile%
  \fi%
 \endgroup%
}
\let\Ginclude@graphics\autopdf@IncludeGraphics
```

Perform the graphics conversions. For (LaTeX/)EPS or PostScript graphics, generates the wrapper LaTeX file and coverts 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.

```
\def\autopdf@ConvertGraphics#1#2{%
 \autopdf@If{\NOT\equal{#1}{#2}}{%
  \autopdf@WriteLog{%
   \autopdf@pc in \autopdf@base#1\autopdf@eol%
  }%
  \autopdf@GetGrType\autopdf@from@type{#1}%
  \autopdf@GetGrType\autopdf@to@type{#2}%
  \autopdf@IfElse{\equal{\autopdf@from@type}{eps}}{%
   \autopdf@ReadGrSize{\autopdf@base}{#1}%
   \autopdf@EPSToPSTeX{#1}%
   \autopdf@CreateMDF\autopdf@Graphics@mdfa{#1}{#2}{%
    \pdfmdfivesum{\the\autopdf@tex@toks}%
   }%
  }{%
   \autopdf@IfElse{\equal{\autopdf@from@type}{tex}}{%
    \newboolean{autopdf@scale@old}%
    \autopdf@IfElse{\boolean{autopdf@scale}}{%
     \setboolean{autopdf@scale@old}{true}%
    }{%
     \setboolean{autopdf@scale@old}{false}%
    }%
    \setboolean{autopdf@scale}{false}%
    \autopdf@ReadGrSize{\autopdf@base}{.eps}%
    \autopdf@EPSToPSTeX{#1}%
    \autopdf@CreateMDF\autopdf@Graphics@mdfa{#1}{#2}{%
```

```
      \pdfmdfivesum file{\autopdf@base.tex}%
      \pdfmdfivesum file{\autopdf@base.eps}%
      \pdfmdfivesum{\the\autopdf@tex@toks}%
    }%
    \autopdf@IfElse{\boolean{autopdf@scale@old}}{%
      \setboolean{autopdf@scale}{true}%
    }{%
      \setboolean{autopdf@scale}{false}%
    }%
  }{%
    \autopdf@CreateMDF\autopdf@Graphics@mdfa{#1}{#2}{%
      \pdfmdfivesum file{\autopdf@base#1}%
    }%
  }%
}%
\edef\autopdf@Graphics@mdfb{\autopdf@GetMDF{\autopdf@base#1}}%
\IfFileExists{\autopdf@base#2}{%
}{%
  \edef\autopdf@Graphics@mdfb{rebuild}%
}%
\autopdf@If{\NOT\(%
  \pdfstrcmp{\autopdf@Graphics@mdfa}{\autopdf@Graphics@mdfb}=0%
\)}{%
  \autopdf@If{\NOT\(\pdfshellescape=1\)}{%
    \PackageError{autopdf}{%
      This package requires pdfLaTeX to %
      be running in "shell escape" mode%
    }{}%
  }%
  \autopdf@IfElse{%
    \equal{\autopdf@from@type}{eps}%
    \OR\equal{\autopdf@from@type}{tex}%
  }{%
    \autopdf@EPSToPS%
    \IfFileExists{\autopdf@base.ps}{%
    }{%
      \PackageError{autopdf}{%
        Could not convert %
        "\autopdf@base#1" to "\autopdf@base.ps". %
        See "\autopdf@base.autopdf.log" for details%
      }{}%
    }%
    \def\autopdf@mid{.ps}%
  }{%
    \autopdf@ReadGrSize{\autopdf@base}{#1}%
    \def\autopdf@mid{#1}%
  }%
  \autopdf@If{\NOT\equal{\autopdf@mid}{#2}}{%
    \autopdf@IfElse{\equal{\autopdf@mid}{.ps}\AND\equal{#2}{.pdf}}{%
      \autopdf@PSToPDF%
    }{%
      \autopdf@Convert{\autopdf@mid}{#2}%
    }%
  }%
```

```
      \IfFileExists{\autopdf@base#2}{%
        \autopdf@If{%
          \NOT\(\equal{\autopdf@mid}{#1}\OR\equal{\autopdf@mid}{#2}\)%
        }{%
          \autopdf@Execute{\DELETE~\autopdf@base\autopdf@mid}%
        }%
      }{%
        \PackageError{autopdf}{%
          Could not convert %
          "\autopdf@base\autopdf@mid" to "\autopdf@base#2". %
          See "\autopdf@base.autopdf.log" for details%
        }{}%
      }%
    }%
    \autopdf@WriteLog{%
      \string\autopdf@SetMDF{\autopdf@base#1}%
        {\autopdf@Graphics@mdfa}\autopdf@eol%
      \autopdf@pc out \autopdf@base#2\autopdf@eol%
    }%
  }%
}
```

Generates the wrapper LaTeX file for (LaTeX/)EPS or PostScript graphics.

```
\def\autopdf@EPSToPSTeX#1{%
  \begingroup%
    \global\autopdf@tex@toks={}%
    \autopdf@ECatToks{\global}{\autopdf@tex@toks}{%
      \string\documentclass{minimal}\autopdf@eol%
      \string\usepackage[%
        paperwidth=\the\autopdf@width,%
        paperheight=\the\autopdf@height,%
        margin=\the\autopdf@margin,%
        offset=0pt,%
        bindingoffset=0pt,%
        noheadfoot,%
        nomarginpar%
        ]{geometry}\autopdf@eol%
      \string\usepackage{graphicx}\autopdf@eol%
      \string\usepackage{psfrag}\autopdf@eol%
      \string\pagestyle{empty}\autopdf@eol%
      \string\setlength{\string\parindent}{0pt}\autopdf@eol%
      \string\setlength{\string\parskip}{0pt}\autopdf@eol%
    }%
    \def\autopdf@fonts{%
      \tiny,\scriptsize,\footnotesize,\small,%
      \normalsize,\large,\Large,\LARGE,\huge,\Huge%
    }%
    \@for\autopdf@a:=\autopdf@fonts\do{%
      \begingroup%
        \autopdf@a%
        \autopdf@ECatToks{\global}{\autopdf@tex@toks}{%
          \string\def\expandafter\string\autopdf@a{%
            \string\fontencoding{\f@encoding}%
            \string\fontfamily{\f@family}%
            \string\fontseries{\f@series}%
      \string\fontshape{\f@shape}%
      \string\fontsize{\f@size}{\f@baselineskip}%
      \string\selectfont%
     }\autopdf@eol%
    }%
   \endgroup%
  }%
  \autopdf@ECatToks{\global}{\autopdf@tex@toks}{%
   \string\normalsize\autopdf@eol%
   \string\makeatletter\autopdf@eol%
  }%
  \count@=\z@%
  \loop\ifnum\count@<\autopdf@Capture@list@count\relax%
   \autopdf@ECatToks{\global}{\autopdf@tex@toks}{%
    \string\def\string\autopdf@act{%
     \csname autopdf@Capture@list@\the\count@\endcsname%
    }%
    \string\autopdf@act\autopdf@eol%
   }%
   \advance\count@\@ne%
  \repeat%
  \autopdf@ECatToks{\global}{\autopdf@tex@toks}{%
   \string\def\string\autopdf@act{}\autopdf@eol%
   \string\makeatother\autopdf@eol%
   \string\begin{document}\autopdf@eol%
  }%
  \autopdf@IfElse{\equal{#1}{.tex}}{%
   \autopdf@ECatToks{\global}{\autopdf@tex@toks}{%
    \string\input{\autopdf@base#1}%
   }%
  }{%
   \autopdf@ECatToks{\global}{\autopdf@tex@toks}{%
    \string\includegraphics[%
    width=0.99\string\textwidth,height=0.99\string\textheight%
    ]{\autopdf@base#1}%
   }%
  }%
  \autopdf@ECatToks{\global}{\autopdf@tex@toks}{%
   \autopdf@pc\pdfmdfivesum file{\autopdf@base#1}\autopdf@eol%
   \string\end{document}%
  }%
 \endgroup%
}
```

Calls `latex` and `dvips` to convert (LaTeX/)EPS graphics to PostScript.

```
\def\autopdf@EPSToPS{%
 \immediate\openout\autopdf@write \autopdf@base.autopdf.tex\relax%
 \immediate\write\autopdf@write{\the\autopdf@tex@toks}%
 \immediate\closeout\autopdf@write%
 \autopdf@Execute{%
  \LEFT~%
   latex~%
    -interaction=nonstopmode~%
    -output-format=dvi~%
    -aux-directory=\autopdf@dir~%
```

```
        -output-directory=\autopdf@dir~%
        \autopdf@base.autopdf.tex~%
      \AND~%
      dvips~%
        -o~\autopdf@base.ps~%
        \autopdf@base.autopdf.dvi~%
      \AND~%
        \DELETE~%
        \autopdf@base.autopdf.tex~\autopdf@base.autopdf.aux~%
        \autopdf@base.autopdf.log~\autopdf@base.autopdf.dvi~%
    \RIGHT~%
    \OR~%
      \DELETE~\autopdf@base.ps%
  }%
}
```

Calls Ghostscript to convert PostScript graphics to PDF.

```
\def\autopdf@PSToPDF{%
  \autopdf@Execute{%
    \autopdf@GS~%
      -dSAFER~-dBATCH~-dNOPAUSE~-q~%
      -sDEVICE=pdfwrite~-dCompatibilityLevel=1.4~%
      -dAutoRotatePages="/None"~%
      -sOutputFile=\autopdf@base.pdf~%
      -c~.setpdfwrite~-f~\autopdf@base.ps~%
    \OR~%
      \DELETE~\autopdf@base.pdf~%
  }%
}
```

Calls the GraphicsMagick `convert` command.

```
\def\autopdf@Convert#1#2{%
  \autopdf@Execute{%
    \LEFT~%
      \autopdf@GMCONVERT~%
        -units~PixelsPerInch~%
        -density~\the\autopdf@resolution~%
        \autopdf@base#1~%
        -resize~\the\autopdf@width@dpi x\the\autopdf@height@dpi~%
        \autopdf@base#2~%
      \AND~%
        identify~\autopdf@base#2~%
    \RIGHT~%
    \OR~%
      \DELETE~\autopdf@base#2~%
  }%
}
⟨/package⟩
```