The epstopdf package

Heiko Oberdiek*

2020-01-24 v2.11

Abstract

This packages adds support of handling eps images to package graphics or graphicx with option pdftex. If an eps image is detected, epstopdf is automatically called to convert it to pdf format.

Contents

*Please report any issues at https://github.com/ho-tex/epstopdf/issues

1 Documentation

1.1 Introduction

LaTeX provides its graphics bundle to include graphics files. Both packages graphics or graphicx may be used. the latter one loads the first and adds options in key value style for `\includegraphics`.

Usually the drivers do not support all kind of graphics files. Other image types must be converted, before they become usuable. In case of driver dvips, the graphics rule may contain a conversion rule. Then all that package graphics must know is the bounding box, the command is passed to dvips that calls it and embeds the converted image.

However, pdfTeX has its driver for PDF output already build in. It's graphics inclusion commands (`\pdfximage`) does not allow the execution of external commands. Therefore commands in the last argument of `\DeclareGraphicsRule` were of no use. But external programs can be called within pdfTeX. This feature is called "shell escape" or "write 18" and must usually enabled explicitely because of security reasons. Now, this package epstopdf hooks into package graphics' code to catch that argument with the external command and executes it to convert the graphics file to a supported format and passes the control of graphics inclusion back to package graphics.

1.2 Requirements

- The feature `\write18` must be enabled. This allows the running of external programs during TeX's compile run. Keep in mind that this is a security

risk. The feature is an addition to TEX. MikTEX and TEX Live support it. In Web2C based TEX distributions (TEX Live) it can be enabled in the configuration file `texmf.cnf`:

```
shell_escape = 1
```

Because of the security risk, it is better to do it on the command line only:

`--shell-escape` (TEX Live)
`--enable-write18` (MiKTEX)

Example:

```
pdflatex -shell-escape test.tex
```

- The program epstopdf for the conversion from EPS to PDF. However, other programs can be used and configured by `\DeclareGraphicsRule`. Example:

```
\epstopdfDeclareGraphicsRule{.eps}{pdf}{.pdf}{%
  ps2pdf -dEPSCrop #1 \OutputFile
}
```

1.3 Usage

The package is loaded after graphic{s,x}, e.g.:

```
\usepackage[pdftex]{graphicx}
\usepackage{epstopdf}
```

Now images with file name extension `.eps` are detected and supported using `\includegraphics`.

If the graphics file name is explicitly specified with extension `.eps` the new rule for EPS files is called and the conversion performed. If option update is in force then the conversion step is dropped if the target file already exists and is not older then the EPS file.

The situation is more complicate if the graphics file is given without file name extension. Then the graphics package must search for a supported image file. The possible extensions are stored in the graphics extension list, that can be set by `\DeclareGraphicsExtensions`. The algorithm:

function search(⟨*filebase*⟩)
  foreach ⟨*ext*⟩ in ⟨*graphics extensions*⟩
    foreach ⟨*dir*⟩ in ⟨*current directory*⟩, ⟨`\graphicspath`⟩
      ⟨*file*⟩ := ⟨*dir*⟩ + ⟨*filebase*⟩ + ⟨*ext*⟩
      if exist ⟨*file*⟩
        return found
  return not found

Package epstopdf puts `.eps` at the end of the graphics extension search list. This is the behaviour of option append that is enabled by default. That means, the conversion is called last unless a supported file type cannot be found earlier. This avoids unnecessary conversion steps that slow down the LATEX run. If you want to use option update and your pdfTEX supports it, then an outdated PDF file also would be found earlier unless suffix is used that is the default since version 2.0.

With an empty option suffix and option prepend there is a risk that an original PDF file is overwritten:

If the original image file is the PDF file and there is also a generated EPS file, then the original PDF file can be regenerated (depending on the option settings)

and the original PDF file gets lost. Therefore option suffix is introduced in version 1.9 to create a separate name space for generated output files.

Note: Usually the conversion program needs the exact location of the image file. Usually the current directory works. Also if the image file is found using `\graphicspath`, the location is known. However, if the image is somewhere in a directory of environment variable `TEXINPUTS`, then the package does not know the exact location and the conversion program will not find the image file unless it implements a search using `TEXINPUTS` (program kpsewhich may be of help in this task).

1.4 Options

Options can be given as package options or later using:

`\epstopdfsetup` {⟨*key value list*⟩}

LaTeX expands the option list before passing the option list to the package's option handling code. This can fail for option suffix if it contains some of the macros described below. Use `\epstopdfsetup` after the package is loaded. Or load package kvoptions-patch before. This package is also loaded by option patch of package kvoptions. LaTeX's option code is redefined to respect key value options and let the values untouched.

update: The conversion program is only called, if the target file does not exist or is older than the source image file.

append: Puts the extension `.eps` at the end of the graphics extension list (default).

prepend: Puts the extension `.eps` at the begin of the graphics extension list.

outdir: The converted file may put in an other output directory. The value of outdir must include the directory separator. Example for the current directory:

```
\epstopdfsetup{outdir=./}
```

For other directories ensure, that they can be found. See `\graphicspath` or `TEXINPUTS`.

suffix: This option takes a string that is put between the file name base and the extension of the output file. Rationale: It can happen, that a PDF file is the original file and the EPS file the generated file. If now the package thinks, that the PDF file is the generated file, it will 'regnerate' it. But in reality the original file is lost. Therefore I recommend to use this option always to generate a separate name space for generated files. Proposed value is `-generated` or `.generated`. The suffix `.generated` will also work here without the need for package grffile).

Example:

```
\epstopdfsetup{suffix=-generated}
```
Then `foo.eps` is converted to `foo-generated.pdf`.

`\SourceExt` can be used inside the suffix string. It's will be replaced by the extension of the image source file without the leading dot, for instance:

```
\epstopdfsetup{suffix=-\SourceExt-converted-to}
foo.eps ⇒ foo-eps-converted-to.pdf
```

See also the next option prefersuffix that modifies the behaviour of option suffix in some cases.

Default for suffix is '`-\SourceExt-converted-to`'.

prefersuffix: If a suffix is set by option suffix, then there can be two image file names that could be taken into account for inclusion: A image file name with the suffix string inside its name and a image file name without; e.g. for `foo.eps` the names could be:

`foo-suffix.pdf`, `foo.pdf`

If option perfersuffix is turned on, the file `foo-suffix.pdf` and its generation is preferred over using `foo.pdf`. Otherwise `foo.pdf` is included without generating `foo-suffix.pdf`. The default of option prefersuffix is `true`.

program@epstopdf: The name for the conversion program from EPS to PDF, default is '`epstopdf`'.

verbose: It prints some information about the image in the `.log` file (default).

1.5 Configuration

1.5.1 System configuration file `epstopdf-sys.cfg`

If `epstopdf-sys` exists it is loaded at the end of the package epstopdf-base and before the user configuration file. It's intended for TeX distributors. Thus they could add additional conversion rules (e.g., .gif -¿ .png) or set options.

1.5.2 User configuration file `epstopdf.cfg`

A configuration file `epstopdf.cfg` is loaded at the end of the package if it exists. It can be used for changing the default option setting. Example:

```
\epstopdfsetup{verbose=false}
```

1.5.3 Conversion program

You can use `\DeclareGraphicsRule` in a similar way as the route via dvips to specify the conversion command line. The conversion argument starts with a back tick, followed by the conversion command including parameters.

The whole conversion argument should also be wrapped inside `\epstopdfcall`. This reduces the problem with packages (e.g. pst-pdf) that use the conversion argument and expands it. Macros `\SourceFile`, `\OutputFile`, and `\SourceExt` are not defined outside epstopdf-base's `\Gin@setfile` and error messages because of undefined command names are the result. If `\epstopdfcall` detects that it is called outside epstopdf-base's `\Gin@setfile` then it replaces the conversion argument by package graphics's default, usually the image file.

The following macros are available inside:

`\OutputFile:` : output file name (with known path and extension)

`\SourceFile:` : source file name (with known path and extension), usually the same as `#1`,

`\SourceExt:` : source extension without leading dot.

Conversion from EPS to PDF. Other programs than epstopdf can be used to convert from EPS to PDF. Example that uses Ghostscript:

```
\DeclareGraphicsRule{.eps}{pdf}{.pdf}{%
 \epstopdfcall{‘ps2pdf -dEPSCrop #1 \noexpand\OutputFile}%
}
```

`\DeclareGraphicsRule` expands the argument, therefore `\noexpand` is necessary. As convenience package epstopdf-base defines `\epstopdfDeclareGraphicsRule`. Then the conversion argument is not expanded, `\epstopdfcall` and the back tick are added:

```
\epstopdfDeclareGraphicsRule{.eps}{pdf}{.pdf}{%
  ps2pdf -dEPSCrop #1 \OutputFile
}
```

Also `\OutputFile` respects the setting of option outdir.

1.6 Other image formats

The support that package epstopdf implements is not limited to EPS files. Other image conversions can be declared. The following example shows it for GIF images under Unix with ImageMagick's convert:

```
\epstopdfDeclareGraphicsRule{.gif}{png}{.png}{%
  convert #1 \OutputFile
}
```

The file extension `.gif` can be added to the extension list that package graphics searches if the file extension is not given in `\includegraphics`. The list can be set by `\GraphicsExtensions`.

```
\AppendGraphicsExtensions{.gif}
```
or
```
\PrependGraphicsExtensions{.gif}
```

2 Implementation

1 ⟨*package⟩

2.1 Wrapper package

Reload check, especially if the package is not used with LaTeX.

```
2 \begingroup\catcode61\catcode48\catcode32=10\relax%
3   \catcode13=5 % ^^M
4   \endlinechar=13 %
5   \catcode35=6 % #
6   \catcode39=12 % ’
7   \catcode44=12 % ,
8   \catcode45=12 % -
9   \catcode46=12 % .
10  \catcode58=12 % :
11  \catcode64=11 % @
12  \catcode123=1 % {
13  \catcode125=2 % }
14  \expandafter\let\expandafter\x\csname ver@epstopdf.sty\endcsname
15  \ifx\x\relax % plain-TeX, first loading
16  \else
```

```
    \def\empty{}%
    \ifx\x\empty % LaTeX, first loading,
      % variable is initialized, but \ProvidesPackage not yet seen
    \else
      \expandafter\ifx\csname PackageInfo\endcsname\relax
        \def\x#1#2{%
          \immediate\write-1{Package #1 Info: #2.}%
        }%
      \else
        \def\x#1#2{\PackageInfo{#1}{#2, stopped}}%
      \fi
      \x{epstopdf}{The package is already loaded}%
      \aftergroup\endinput
    \fi
  \fi
\endgroup%
```

Package identification:

```
\begingroup\catcode61\catcode48\catcode32=10\relax%
  \catcode13=5 % ^^M
  \endlinechar=13 %
  \catcode35=6 % #
  \catcode39=12 % '
  \catcode40=12 % (
  \catcode41=12 % )
  \catcode44=12 % ,
  \catcode45=12 % -
  \catcode46=12 % .
  \catcode47=12 % /
  \catcode58=12 % :
  \catcode64=11 % @
  \catcode91=12 % [
  \catcode93=12 % ]
  \catcode123=1 % {
  \catcode125=2 % }
  \expandafter\ifx\csname ProvidesPackage\endcsname\relax
    \def\x#1#2#3[#4]{\endgroup
      \immediate\write-1{Package: #3 #4}%
      \xdef#1{#4}%
    }%
  \else
    \def\x#1#2[#3]{\endgroup
      #2[{#3}]%
      \ifx#1\@undefined
        \xdef#1{#3}%
      \fi
      \ifx#1\relax
        \xdef#1{#3}%
      \fi
    }%
  \fi
\expandafter\x\csname ver@epstopdf.sty\endcsname
\ProvidesPackage{epstopdf}%
  [2020-01-24 v2.11 Conversion with epstopdf on the fly (HO)]%
```

Larger catcode set because of configuration files needed.

```
\begingroup\catcode61\catcode48\catcode32=10\relax%
  \catcode13=5 % ^^M
  \endlinechar=13 %
```

```
  \catcode123=1 % {
  \catcode125=2 % }
  \catcode64=11 % @
  \def\x{\endgroup
    \expandafter\edef\csname ETE@AtEnd\endcsname{%
      \endlinechar=\the\endlinechar\relax
      \catcode13=\the\catcode13\relax
      \catcode32=\the\catcode32\relax
      \catcode35=\the\catcode35\relax
      \catcode61=\the\catcode61\relax
      \catcode64=\the\catcode64\relax
      \catcode123=\the\catcode123\relax
      \catcode125=\the\catcode125\relax
    }%
  }%
\x\catcode61\catcode48\catcode32=10\relax%
\catcode13=5 % ^^M
\endlinechar=13 %
\catcode35=6 % #
\catcode64=11 % @
\catcode123=1 % {
\catcode125=2 % }
\def\TMP@EnsureCode#1#2{%
  \edef\ETE@AtEnd{%
    \ETE@AtEnd
    \catcode#1=\the\catcode#1\relax
  }%
  \catcode#1=#2\relax
}
\TMP@EnsureCode{39}{12}% '
\TMP@EnsureCode{40}{12}% (
\TMP@EnsureCode{41}{12}% )
\TMP@EnsureCode{42}{12}% *
\TMP@EnsureCode{45}{12}% -
\TMP@EnsureCode{47}{12}% /
\TMP@EnsureCode{91}{12}% [
\TMP@EnsureCode{93}{12}% ]
\edef\ETE@AtEnd{\ETE@AtEnd\noexpand\endinput}
```

Ensure packages loaded by the full epstopdf, for compatibility.

```
\let\ETE@SavedAtEnd\ETE@AtEnd
\RequirePackage{infwarerr}[2007/09/09]
\RequirePackage{grfext}\relax
\RequirePackage{kvoptions}[2007/10/02]
\RequirePackage{pdftexcmds}[2007/11/11]
\RequirePackage{epstopdf-base}[2019/11/27]
\let\ETE@AtEnd\ETE@SavedAtEnd
```

2.1.1 Option handling

```
\DeclareOption*{%
  \expandafter\epstopdfsetup\expandafter{\CurrentOption}%
}
\ProcessOptions*\relax
```

```
\ETE@AtEnd%
```

⟨/package⟩

2.2 Base package

123 ⟨*base⟩

2.3 Preparations

2.3.1 Relead check and identification

Reload check, especially if the package is not used with LaTeX.

```
124 \begingroup\catcode61\catcode48\catcode32=10\relax%
125   \catcode13=5 % ^^M
126   \endlinechar=13 %
127   \catcode35=6 % #
128   \catcode39=12 % '
129   \catcode44=12 % ,
130   \catcode45=12 % -
131   \catcode46=12 % .
132   \catcode58=12 % :
133   \catcode64=11 % @
134   \catcode123=1 % {
135   \catcode125=2 % }
136   \expandafter\let\expandafter\x\csname ver@epstopdf-base.sty\endcsname
137   \ifx\x\relax % plain-TeX, first loading
138   \else
139     \def\empty{}%
140     \ifx\x\empty % LaTeX, first loading,
141       % variable is initialized, but \ProvidesPackage not yet seen
142     \else
143       \expandafter\ifx\csname PackageInfo\endcsname\relax
144         \def\x#1#2{%
145           \immediate\write-1{Package #1 Info: #2.}%
146         }%
147       \else
148         \def\x#1#2{\PackageInfo{#1}{#2, stopped}}%
149       \fi
150       \x{epstopdf-base}{The package is already loaded}%
151       \aftergroup\endinput
152     \fi
153   \fi
154 \endgroup%
```

Package identification:

```
155 \begingroup\catcode61\catcode48\catcode32=10\relax%
156   \catcode13=5 % ^^M
157   \endlinechar=13 %
158   \catcode35=6 % #
159   \catcode39=12 % '
160   \catcode40=12 % (
161   \catcode41=12 % )
162   \catcode44=12 % ,
163   \catcode45=12 % -
164   \catcode46=12 % .
165   \catcode47=12 % /
166   \catcode58=12 % :
167   \catcode64=11 % @
168   \catcode91=12 % [
169   \catcode93=12 % ]
170   \catcode123=1 % {
171   \catcode125=2 % }
172   \expandafter\ifx\csname ProvidesPackage\endcsname\relax
173     \def\x#1#2#3[#4]{\endgroup
```

```
      \immediate\write-1{Package: #3 #4}%
      \xdef#1{#4}%
    }%
  \else
    \def\x#1#2[#3]{\endgroup
      #2[{#3}]%
      \ifx#1\@undefined
        \xdef#1{#3}%
      \fi
      \ifx#1\relax
        \xdef#1{#3}%
      \fi
    }%
  \fi
\expandafter\x\csname ver@epstopdf-base.sty\endcsname
\ProvidesPackage{epstopdf-base}%
  [2020-01-24 v2.11 Base part for package epstopdf]%
```

2.3.2 Catcodes

```
\begingroup\catcode61\catcode48\catcode32=10\relax%
  \catcode13=5 % ^^M
  \endlinechar=13 %
  \catcode123=1 % {
  \catcode125=2 % }
  \catcode64=11 % @
  \def\x{\endgroup
    \expandafter\edef\csname ETE@AtEnd\endcsname{%
      \endlinechar=\the\endlinechar\relax
      \catcode13=\the\catcode13\relax
      \catcode32=\the\catcode32\relax
      \catcode35=\the\catcode35\relax
      \catcode61=\the\catcode61\relax
      \catcode64=\the\catcode64\relax
      \catcode123=\the\catcode123\relax
      \catcode125=\the\catcode125\relax
    }%
  }%
\x\catcode61\catcode48\catcode32=10\relax%
\catcode13=5 % ^^M
\endlinechar=13 %
\catcode35=6 % #
\catcode64=11 % @
\catcode123=1 % {
\catcode125=2 % }
\def\TMP@EnsureCode#1#2{%
  \edef\ETE@AtEnd{%
    \ETE@AtEnd
    \catcode#1=\the\catcode#1\relax
  }%
  \catcode#1=#2\relax
}
\TMP@EnsureCode{33}{12}% !
\TMP@EnsureCode{39}{12}% '
\TMP@EnsureCode{42}{12}% *
\TMP@EnsureCode{44}{12}% ,
\TMP@EnsureCode{45}{12}% -
\TMP@EnsureCode{46}{12}% .
```

```
\TMP@EnsureCode{47}{12}% /
\TMP@EnsureCode{58}{12}% :
\TMP@EnsureCode{60}{12}% <
\TMP@EnsureCode{62}{12}% >
\TMP@EnsureCode{96}{12}% `
\edef\ETE@AtEnd{\ETE@AtEnd\noexpand\endinput}
```

2.3.3 Load packages

This package is split into epstopdf-base and epstopdf The base version is called directly by the graphics package. Unfortunately it still includes many contributed packages which breaks the layering of the core LaTeX graphics release.

Compatibility concerns mean refactoring the package to only have options in epstopdf is tricky, so from release 2.8, if no options have been passed in to epstopdf-base, the package loading and option setting are skipped and the default settings are set directly. In the case of LuaTeX, the Lua portion of pdftexcmds is used to provide the file handling functionality.

```
\ifx\@curroptions\@empty
  \ifx\@PackageInfo\@undefined
    \let\@PackageInfo\PackageInfo
    \let\@PackageWarningNoLine\PackageWarningNoLine
    \def\@PackageInfoNoLine#1#2{\PackageInfo{#1}{#2\@gobble}}
    \let\@PackageError\PackageError
  \fi
  \ifx\pdf@strcmp\@undefined
    \ifx\directlua\@undefined
      \def\pdf@strcmp{\pdfstrcmp}
      \def\pdf@filemoddate{\pdffilemoddate}
      \def\pdf@filesize{\pdffilesize}
      \def\pdf@system#{\immediate\write18 }
    \else
      \directlua{require('pdftexcmds')}
      \long\def\pdf@strcmp#1#2{\directlua{%
        oberdiek.pdftexcmds.strcmp('\luaescapestring{#1}',
                                   '\luaescapestring{#2}')}}%
      \def\pdf@filemoddate#1{\directlua{%
        oberdiek.pdftexcmds.filemoddate('\luaescapestring{#1}')}}
      \def\pdf@filesize#1{\directlua{%
        oberdiek.pdftexcmds.filesize('\luaescapestring{#1}')}}
      \def\pdf@system#1{\directlua{%
        oberdiek.pdftexcmds.system('\luaescapestring{#1}')}}
    \fi
  \fi
\else
  \RequirePackage{infwarerr}[2007/09/09]
  \RequirePackage{grfext}\relax
  \RequirePackage{kvoptions}[2007/10/02]
  \RequirePackage{pdftexcmds}[2007/11/11]
\fi
```

2.4 Checks

Check, whether package graphics is loaded (also graphicx loads graphics). Because miniltx.tex does not know `\@ifpackageloaded` we test for `\Gin@setfile` instead.

```
\begingroup\expandafter\expandafter\expandafter\endgroup
\expandafter\ifx\csname Gin@setfile\endcsname\relax
  \@PackageWarningNoLine{epstopdf}{%
```

```
    No graphics package \string‘graphic{s,x}\string’ loaded%
  }%
  \newcommand*{\epstopdfsetup}[1]{}%
  \expandafter\ETE@AtEnd
\fi%
```

Check, whether pdftex.def is loaded. `\ver@pdftex.def` is not available with miniltx.tex, thus we test for `\Gin@driver`.

```
\begingroup
  \def\x{luatex.def}%
  \ifx\Gin@driver\x
  \else
  \def\x{pdftex.def}%
  \ifx\Gin@driver\x
  \else
    \@PackageWarningNoLine{epstopdf}{%
      Drivers other than ‘pdftex’ and ‘luatex’ are not supported%
    }%
    \endgroup
    \newcommand*{\epstopdfsetup}[1]{}%
    \expandafter\expandafter\expandafter\ETE@AtEnd
  \fi%
  \fi%
\endgroup
```

Check, whether the shell escape feature is enabled.

```
\begingroup
  \expandafter\ifx\csname pdf@shellescape\endcsname\relax
  \else
    \ifnum\pdf@shellescape>0 %
    \else
      \@PackageWarningNoLine{epstopdf}{%
        Shell escape feature is not enabled%
      }%
    \fi
  \fi
\endgroup
```

2.5 Options

As noted above, if no options have been passed in (typically if called directly from `pdftex.def`) then the kvoptions handling is not loaded and the defaults are set directly.

```
\newif\ifETE@prepend
\ifx\SetupKeyvalOptions\@undefined
  \def\ETE@let#1#2{%
    \expandafter\let\csname ifETE@#1\expandafter\endcsname
    \csname if#2\endcsname}
  \ETE@let{verbose}{true}
  \ETE@let{disable}{false}
  \ETE@let{update}{true}
  \ETE@let{prepend}{false}
  \ETE@let{prefersuffix}{true}
  \def\ETE@outdir{}
  \def\ETE@suffix{-\SourceExt-converted-to}
  \def\ETE@program@epstopdf{\epstopdf@sys@cmd}
  \newcommand*{\epstopdfsetup}[1]{}%
\else
```

```
  \SetupKeyvalOptions{family=ETE,prefix=ETE@}
  \DeclareBoolOption{update}
  \DeclareBoolOption{verbose}
  \DeclareVoidOption{prepend}{\ETE@prependtrue}
  \DeclareVoidOption{append}{\ETE@prependfalse}
  \DeclareStringOption{outdir}
  \DeclareStringOption{suffix}
  \DeclareBoolOption{prefersuffix}
  \DeclareStringOption{program@epstopdf}
```

Options disable and enable are for testing only. Therefore they are not documented on purpose.

```
  \DeclareBoolOption{disable}
  \DeclareComplementaryOption{enable}{disable}
  \newcommand*{\epstopdfsetup}{\setkeys{ETE}}
```

2.5.1 Default setting

```
  \epstopdfsetup{%
    verbose,%
    enable,%
    append,%
    update,%
    prefersuffix,%
    suffix=-\SourceExt-converted-to,%
    program@epstopdf=epstopdf%
  }
\fi
```

2.6 Make and verbose

```
\begingroup\expandafter\expandafter\expandafter\endgroup
\expandafter\ifx\csname pdf@filemoddate\endcsname\relax
  \def\ETE@Make#1#2{%
    \ifETE@update
      \ETE@WarnModDate
    \fi
    \@firstofone
  }%
  \def\ETE@WarnModDate{%
    \@PackageWarningNoLine{epstopdf}{%
      \string\pdffilemoddate\space is not available,\MessageBreak
      option 'update' will be ignored%
    }%
    \global\let\ETE@WarnModDate\relax
  }%
  \def\ETE@FileInfo#1#2{#1 file: <#2>}%
\else
  \def\ETE@Make#1#2{%
    \ifETE@update
      \ifnum\pdf@strcmp{\pdf@filemoddate{#1}}{\pdf@filemoddate{#2}}>0 %
        \expandafter\expandafter\expandafter\@firstofone
      \else
        \@PackageInfoNoLine{epstopdf}{%
          Output file is already uptodate%
        }%
        \expandafter\expandafter\expandafter\@gobble
      \fi
```

```
    \else
      \expandafter\@firstofone
    \fi
  }%
  \def\ETE@FileInfo#1#2{%
    #1 file: <#2>%
    \expandafter\expandafter\expandafter
    \ETE@Date\pdf@filemoddate{#2}\@nil
    \expandafter\expandafter\expandafter
    \ETE@Size\pdf@filesize{#2}\@nil
  }%
  \def\ETE@Date#1\@nil{%
    \ifx\\#1\\%
    \else
      \ETE@@Date#1\@nil
    \fi
  }%
  \def\ETE@@Date#1:#2#3#4#5#6#7#8#9{%
    \MessageBreak
    \@spaces\space\space\space date: #2#3#4#5-#6#7-#8#9 %
    \ETE@@Time
  }%
  \def\ETE@@Time#1#2#3#4#5#6#7\@nil{%
    #1#2:#3#4:#5#6%
  }%
  \def\ETE@Size#1\@nil{%
    \ifx\\#1\\%
    \else
      \MessageBreak
      \@spaces\space\space\space size: #1 bytes%
    \fi
  }%
\fi
```

2.7 Adding conversion support

Patch `\Gin@setfile` to execute `#3`, if it contains a command.

```
\expandafter\ifx\csname ETE@OrgGin@setfile\endcsname\relax
  \let\ETE@OrgGin@setfile\Gin@setfile
\else
  \@PackageError{epstopdf}{%
    Command \string\ETE@OrgGin@setfile\space
    already defined.\MessageBreak
  }{%
    Probably some package has included the code of this package%
    \MessageBreak
    instead of using \string\RequirePackage{epstopdf}.%
    \MessageBreak
    \@ehc
  }%
\fi
\def\ETE@IfFileExists{%
  \begingroup\expandafter\expandafter\expandafter\endgroup
  \expandafter\ifx\csname grffile@IfFileExists\endcsname\relax
    \expandafter\IfFileExists
  \else
    \global\let\ETE@IfFileExists\grffile@IfFileExists
    \expandafter\grffile@IfFileExists
  \fi
```

```
}
\def\ETE@Skip#1#2\x\fi\fi{%
  \fi
  \fi
  \endgroup
  \fi
  \fi
  #1%
}
\newif\ifETE@InsideSetfile
\newcommand*{\epstopdfcall}[1]{%
  \ifETE@InsideSetfile
    \expandafter\@firstoftwo
  \else
    \expandafter\@secondoftwo
  \fi
  {`#1}%
  {\Gin@base\Gin@ext}%
}
\def\ETE@DefCommandLine#1{%
  \edef\CommandLine{\expandafter\fi\if`#1}%
}
\def\ETE@DefX#1{%
  \expandafter\expandafter\expandafter\def
  \expandafter\expandafter\expandafter\x
  \expandafter\expandafter\expandafter{%
    \expandafter\fi\if`#1\relax\else
  }%
}
\def\ETE@Gin@setfile#1#2#3{%
  \ifETE@disable
    \ETE@OrgGin@setfile{#1}{#2}{#3}%
  \else
    \begingroup
      \ETE@InsideSetfiletrue
      \ETE@DefX{#3}%
    \expandafter\endgroup
    \ifx\x\@empty
      \ETE@OrgGin@setfile{#1}{#2}{#3}%
    \else
      \begingroup
        \ETE@InsideSetfiletrue
        \def\GraphicsType{#1}%
        \def\GraphicsRead{#2}%
        \ifETE@prefersuffix
        \else
          \ifx\ETE@suffix\@empty
          \else
            \ETE@IfFileExists{\Gin@base\GraphicsRead}{%
              \ETE@Skip{%
                \ETE@OrgGin@setfile{#1}{#2}{\Gin@base#2}%
              }%
            }{%
              \let\next\relax
            }%
            \next
          \fi
        \fi
```

```
    \ifx\Gin@ext\relax
      \let\SourceExt\Gin@eext
      \def\SourceFile{\Gin@base\Gin@eext}%
    \else
      \let\SourceExt\Gin@ext
      \def\SourceFile{\Gin@base\Gin@ext}%
    \fi
    \edef\SourceExt{% remove dot
      \expandafter\@cdr\SourceExt\@empty\@nil
    }%
    \let\OutputDirectory\ETE@outdir
    \ifx\OutputDirectory\@empty
      \edef\OutputFile{\ETE@GenerateName{\Gin@base}{#2}}%
    \else
      \begingroup
        \filename@parse{\Gin@base#2}%
        \edef\x{\endgroup
          \def\noexpand\OutputFile{%
            \ETE@GenerateName{%
              \OutputDirectory\filename@base
            }{#2}%
          }%
        }%
      \x
    \fi
    \ETE@DefCommandLine{#3}%
    \ifETE@verbose
      \@PackageInfo{epstopdf}{%
        \ETE@FileInfo{Source}\SourceFile\MessageBreak
        \ETE@FileInfo{Output}\OutputFile\MessageBreak
        Command: <\CommandLine>\MessageBreak
        \string\includegraphics
      }%
    \fi
    \ETE@Make\SourceFile\OutputFile{%
      \pdf@system{\CommandLine}%
      \ifETE@verbose
        \@PackageInfoNoLine{epstopdf}{%
          \ETE@FileInfo{Result}\OutputFile
        }%
      \fi
    }%
    \edef\x{\endgroup
      \ifx\OutputDirectory\@empty
      \else
        \def\noexpand\Gin@base{%
          \OutputDirectory\noexpand\filename@base
        }%
      \fi
      \ifx\ETE@suffix\@empty
      \else
        \edef\noexpand\Gin@base{%
          \noexpand\Gin@base\ETE@suffix
        }%
      \fi
      \noexpand\ETE@OrgGin@setfile{%
        \GraphicsType
      }{%
```

```
            \GraphicsRead
          }{%
            \OutputFile
          }%
        }%
      \x
    \fi
  \fi
}
\let\Gin@setfile\ETE@Gin@setfile
\def\ETE@GenerateName#1#2{%
  #1\ETE@suffix#2%
}
```

2.8 Declare graphics rule

```
\newcommand*{\epstopdfDeclareGraphicsRule}[4]{%
  \ifx\\#4\\%
    \@PackageError{epstopdf-base}{%
      Conversion command is missing%
    }\@ehc
  \else
    \begingroup
      \@ifundefined{Gin@rule@#1}{%
      }{%
        \@PackageInfo{epstopdf-base}{%
          Redefining graphics rule for `#1'%
        }%
      }%
    \endgroup
    \@namedef{Gin@rule@#1}##1{{#2}{#3}{\epstopdfcall{#4}}}%
  \fi
}
```

`\DeclareGraphicsRule` for .eps

```
\epstopdfDeclareGraphicsRule{.eps}{pdf}{.pdf}{%
  \ETE@epstopdf{#1}%
}
\def\ETE@epstopdf#1{%
  \ETE@program@epstopdf\space
  \ifcase\ifx\OutputDirectory\@empty
           \ifx\ETE@suffix\@empty
             1%
           \fi
         \fi
         0 %
    --outfile=\OutputFile\space
  \fi
  #1%
}
\ifx\AppendGraphicsExtensions\@undefined\else
\ifETE@prepend
  \expandafter\PrependGraphicsExtensions
\else
  \expandafter\AppendGraphicsExtensions
\fi
{.eps}
\fi
```

```
\let\ETE@prepend\@undefined
\ifx\SetupKeyvalOptions\@undefined
\else
\DeclareVoidOption{prepend}{%
  \PrependGraphicsExtensions{.eps}%
}
\let\ETE@append\@undefined
\DeclareVoidOption{append}{%
  \AppendGraphicsExtensions{.eps}%
}
\fi
\InputIfFileExists{epstopdf-sys.cfg}{}{}
\InputIfFileExists{epstopdf.cfg}{}{}
```

Use `epstpdf` if the cfg files have not set a default.

```
\expandafter\ifx\csname epstopdf@sys@cmd\endcsname\relax
  \def\epstopdf@sys@cmd{epstopdf}
\fi
\ETE@AtEnd%
```

⟨/base⟩

3 Installation

3.1 Download

Package. This package is available on CTAN[1]:

CTAN:macros/latex/contrib/epstopdf-pkg/epstopdf.dtx The source file.

CTAN:macros/latex/contrib/epstopdf-pkg/epstopdf.pdf Documentation.

Bundle. All the packages of the bundle 'epstopdf' are also available in a TDS compliant ZIP archive. There the packages are already unpacked and the documentation files are generated. The files and directories obey the TDS standard.

CTAN:install/macros/latex/contrib/epstopdf-pkg.tds.zip

TDS refers to the standard "A Directory Structure for TEX Files" (CTAN:pkg/tds). Directories with `texmf` in their name are usually organized this way.

3.2 Bundle installation

Unpacking. Unpack the `epstopdf.tds.zip` in the TDS tree (also known as `texmf` tree) of your choice. Example (linux):

```
unzip epstopdf.tds.zip -d ~/texmf
```

3.3 Package installation

Unpacking. The `.dtx` file is a self-extracting docstrip archive. The files are extracted by running the `.dtx` through plain TEX:

```
tex epstopdf.dtx
```

[1] CTAN:pkg/epstopdf-pkg

TDS. Now the different files must be moved into the different directories in your installation TDS tree (also known as `texmf` tree):

```
epstopdf.sty      → tex/latex/epstopdf/epstopdf.sty
epstopdf-base.sty → tex/latex/epstopdf/epstopdf-base.sty
epstopdf.pdf      → doc/latex/epstopdf/epstopdf.pdf
epstopdf.dtx      → source/latex/epstopdf/epstopdf.dtx
```

If you have a `docstrip.cfg` that configures and enables docstrip's TDS installing feature, then some files can already be in the right place, see the documentation of docstrip.

3.4 Refresh file name databases

If your TEX distribution (TEX Live, MiKTEX, ...) relies on file name databases, you must refresh these. For example, TEX Live users run `texhash` or `mktexlsr`.

3.5 Some details for the interested

Unpacking with LATEX. The `.dtx` chooses its action depending on the format:

plain TEX: Run docstrip and extract the files.

LATEX: Generate the documentation.

If you insist on using LATEX for docstrip (really, docstrip does not need LATEX), then inform the autodetect routine about your intention:

```
latex \let\install=y\input{epstopdf.dtx}
```

Do not forget to quote the argument according to the demands of your shell.

Generating the documentation. You can use both the `.dtx` or the `.drv` to generate the documentation. The process can be configured by the configuration file `ltxdoc.cfg`. For instance, put this line into this file, if you want to have A4 as paper format:

```
\PassOptionsToClass{a4paper}{article}
```

An example follows how to generate the documentation with pdfLATEX:

```
pdflatex epstopdf.dtx
makeindex -s gind.ist epstopdf.idx
pdflatex epstopdf.dtx
makeindex -s gind.ist epstopdf.idx
pdflatex epstopdf.dtx
```

4 History

[2001/01/06 v1.0]

- First public version, published in the pdfTEX mailing list.

[2001/02/04 v1.1]

- Minor documentation update.
- CTAN.

[2006/02/20 v1.2]

- DTX framework.
- Compatibility for `miniltx.tex`.

[2006/08/26 v1.3]

- Check for `\write18` if available and print a warning if the feature is not enabled.

[2007/04/26 v1.4]

- Documentation rewritten and extended.

[2007/10/02 v1.5]

- New option update: If the converted file exists, it will be only converted if it is out of date.
- Updating the extension list is delegated to package grfext. Fine tuning is done by the new options append, prepend.
- New option outdir for changing the output directory.
- New option verbose.
- `\SourceFile` and `\OutputFile` introduced.
- Configuration file support added.

[2007/11/11 v1.6]

- Use of package pdftexcmds for LuaTEX support.

[2008/05/06 v1.7]

- Warning messages uses "loaded" instead of "found".

[2009/03/01 v1.8]

- Warning message for missing `pdftex.def` changed.

[2009/07/06 v1.9]

- Option suffix added.

[2009/07/07 v1.10]

- `\SourceExt` added.
- If option suffix is set, the inclusion of an image without the suffix namespace is preferred over generating the the image within the suffix namespace.

[2009/07/12 v2.0]

- New default settings.
- Package is split into epstopdf that only takes package options and loads epstopdf-base that does the work.
- `\epstopdfDeclareGraphicsRule` and `\epstopdfcall` added.
- `epstopdf-sys.cfg` is loaded before `epstopdf.cfg` if `epstopdf-sys.cfg` exists.

[2009/07/15 v2.1]

- Default setting: verbose is now turned on as the documentation for v2.0 said.
- Documentation fixes.

[2009/07/16 v2.2]

- Fixed redefined `\Gin@setfile`.
- Documentation extended for package options.

[2009/09/24 v2.3]

- Bug fix for the case that both option suffix and outdir are used.

[2009/10/17 v2.4]

- The name of the program 'epstopdf' can be configured via the new option program@epstopdf.

[2016/05/15 v2.5]

- Wording of warning message fixed (Karl Berry).
- `\ETE@Gin@setfile` added (Karl Berry).

[2016/05/15 v2.6]

- luaTeX compatibility

[2019/11/24 v2.7]

- New epstopdf repository.

[2019/11/27 v2.8]

- The base package configured not to require additional packages if called with no arguments.
- Code re-arranged to avoid:
  `\end occurred when \ifx on line 165 was incomplete` warning in dvi mode.

[2019-11-30 v2.9]

- Read the `epstopdf.cfg` file even in the base code is being used.

[2020-01-24 v2.11]

- use ’ not " for Lua strings, as the latter not made safe.

[2020-01-24 v2.11]

- Ensure that `\epstopdf@sys@cmd` is defined even if no `cfg` file is found.

5 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; plain numbers refer to the code lines where the entry is used.