The epstopdf package

Heiko Oberdiek*
<heiko.oberdiek at googlemail.com>

2016/05/15 v2.6

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

1 Documentation **2**
1.1 Introduction . . . 2
1.2 Requirements . . . 2
1.3 Usage . . . 3
1.4 Options . . . 4
1.5 Configuration . . . 5
1.5.1 System configuration file epstopdf-sys.cfg . . . 5
1.5.2 User configuration file epstopdf.cfg . . . 5
1.5.3 Conversion program . . . 5
1.6 Other image formats . . . 6

2 Implementation **6**
2.1 Wrapper package . . . 6
2.1.1 Option handling . . . 8
2.2 Base package . . . 8
2.3 Preparations . . . 8
2.3.1 Reload check and identification . . . 8
2.3.2 Catcodes . . . 9
2.3.3 Load packages . . . 10
2.4 Checks . . . 10
2.5 Package loading . . . 11
2.6 Options . . . 11
2.6.1 Default setting . . . 11
2.7 Make and verbose . . . 11
2.8 Adding conversion support . . . 12
2.9 Declare graphics rule . . . 15

3 Test **16**
3.1 Preface for standard catcode check . . . 16
3.2 Catcode checks for loading . . . 16

4 Installation **17**
4.1 Download . . . 17
4.2 Bundle installation . . . 18
4.3 Package installation . . . 18
4.4 Refresh file name databases . . . 18
4.5 Some details for the interested . . . 18

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

5 Catalogue **19**

6 History **19**
[2001/01/06 v1.0] . . . 19
[2001/02/04 v1.1] . . . 19
[2006/02/20 v1.2] . . . 20
[2006/08/26 v1.3] . . . 20
[2007/04/26 v1.4] . . . 20
[2007/10/02 v1.5] . . . 20
[2007/11/11 v1.6] . . . 20
[2008/05/06 v1.7] . . . 20
[2009/03/01 v1.8] . . . 20
[2009/07/06 v1.9] . . . 20
[2009/07/07 v1.10] . . . 20
[2009/07/12 v2.0] . . . 20
[2009/07/15 v2.1] . . . 21
[2009/07/16 v2.2] . . . 21
[2009/09/24 v2.3] . . . 21
[2009/10/17 v2.4] . . . 21
[2016/05/15 v2.5] . . . 21
[2016/05/15 v2.6] . . . 21

7 Index **21**

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, teTeX, TeX Live support it. In Web2C based TeX distributions (teTeX, 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 (teTEX, 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

```
⟨*package⟩
```

2.1 Wrapper package

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

```
\begingroup\catcode61\catcode48\catcode32=10\relax%
 \catcode13=5 % ^^M
 \endlinechar=13 %
 \catcode35=6 % #
 \catcode39=12 % '
 \catcode44=12 % ,
 \catcode45=12 % -
 \catcode46=12 % .
 \catcode58=12 % :
 \catcode64=11 % @
 \catcode123=1 % {
 \catcode125=2 % }
 \expandafter\let\expandafter\x\csname ver@epstopdf.sty\endcsname
 \ifx\x\relax % plain-TeX, first loading
 \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}%
  [2016/05/15 v2.6 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}
\let\ETE@SavedAtEnd\ETE@AtEnd
\RequirePackage{epstopdf-base}[2016/05/15]
\let\ETE@AtEnd\ETE@SavedAtEnd
```

2.1.1 Option handling

```
\DeclareOption*{%
  \expandafter\epstopdfsetup\expandafter{\CurrentOption}%
}
\ProcessOptions*\relax
\ETE@AtEnd%
⟨/package⟩
```

2.2 Base package

```
⟨*base⟩
```

2.3 Preparations

2.3.1 Relead check and identification

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

```
\begingroup\catcode61\catcode48\catcode32=10\relax%
  \catcode13=5 % ^^M
  \endlinechar=13 %
  \catcode35=6 % #
  \catcode39=12 % '
  \catcode44=12 % ,
  \catcode45=12 % -
  \catcode46=12 % .
  \catcode58=12 % :
  \catcode64=11 % @
  \catcode123=1 % {
  \catcode125=2 % }
  \expandafter\let\expandafter\x\csname ver@epstopdf-base.sty\endcsname
  \ifx\x\relax % plain-TeX, first loading
  \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-base}{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-base.sty\endcsname
\ProvidesPackage{epstopdf-base}%
  [2016/05/15 v2.6 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

```
\RequirePackage{infwarerr}[2007/09/09]
\RequirePackage{grfext}\relax
\RequirePackage{kvoptions}[2007/10/02]
\RequirePackage{pdftexcmds}[2007/11/11]
```

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\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 Package loading

2.6 Options

```
\SetupKeyvalOptions{family=ETE,prefix=ETE@}
\DeclareBoolOption{update}
\DeclareBoolOption{verbose}
\newif\ifETE@prepend
\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}}
```

2.6.1 Default setting

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

2.7 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
```

2.8 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%
}
```

2.9 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%
}
\ifETE@prepend
 \expandafter\PrependGraphicsExtensions
\else
 \expandafter\AppendGraphicsExtensions
\fi
{.eps}
\let\ETE@prepend\@undefined
\DeclareVoidOption{prepend}{%
 \PrependGraphicsExtensions{.eps}%
}
\let\ETE@append\@undefined
\DeclareVoidOption{append}{%
 \AppendGraphicsExtensions{.eps}%
}
\InputIfFileExists{epstopdf-sys.cfg}{}{}
```

```
\InputIfFileExists{epstopdf.cfg}{}{}
\ETE@AtEnd%
⟨/base⟩
```

3 Test

3.1 Preface for standard catcode check

```
⟨*test1⟩
\input miniltx.tex\relax
\def\Gin@driver{pdftex.def}
\input graphicx.sty\relax
\resetatcatcode
⟨/test1⟩
```

3.2 Catcode checks for loading

```
⟨*test1⟩
\catcode`\{=1 %
\catcode`\}=2 %
\catcode`\#=6 %
\catcode`\@=11 %
\expandafter\ifx\csname count@\endcsname\relax
  \countdef\count@=255 %
\fi
\expandafter\ifx\csname @gobble\endcsname\relax
  \long\def\@gobble#1{}%
\fi
\expandafter\ifx\csname @firstofone\endcsname\relax
  \long\def\@firstofone#1{#1}%
\fi
\expandafter\ifx\csname loop\endcsname\relax
  \expandafter\@firstofone
\else
  \expandafter\@gobble
\fi
{%
  \def\loop#1\repeat{%
    \def\body{#1}%
    \iterate
  }%
  \def\iterate{%
    \body
      \let\next\iterate
    \else
      \let\next\relax
    \fi
    \next
  }%
  \let\repeat=\fi
}%
\def\RestoreCatcodes{}
\count@=0 %
\loop
  \edef\RestoreCatcodes{%
    \RestoreCatcodes
    \catcode\the\count@=\the\catcode\count@\relax
  }%
\ifnum\count@<255 %
  \advance\count@ 1 %
\repeat

\def\RangeCatcodeInvalid#1#2{%
```

```
  \count@=#1\relax
  \loop
    \catcode\count@=15 %
  \ifnum\count@<#2\relax
    \advance\count@ 1 %
  \repeat
}
\def\RangeCatcodeCheck#1#2#3{%
  \count@=#1\relax
  \loop
    \ifnum#3=\catcode\count@
    \else
      \errmessage{%
        Character \the\count@\space
        with wrong catcode \the\catcode\count@\space
        instead of \number#3%
      }%
    \fi
  \ifnum\count@<#2\relax
    \advance\count@ 1 %
  \repeat
}
\def\space{ }
\expandafter\ifx\csname LoadCommand\endcsname\relax
  \def\LoadCommand{\input epstopdf.sty\relax}%
\fi
\def\Test{%
  \RangeCatcodeInvalid{0}{47}%
  \RangeCatcodeInvalid{58}{64}%
  \RangeCatcodeInvalid{91}{96}%
  \RangeCatcodeInvalid{123}{255}%
  \catcode`\@=12 %
  \catcode`\\=0 %
  \catcode`\%=14 %
  \LoadCommand
  \RangeCatcodeCheck{0}{36}{15}%
  \RangeCatcodeCheck{37}{37}{14}%
  \RangeCatcodeCheck{38}{47}{15}%
  \RangeCatcodeCheck{48}{57}{12}%
  \RangeCatcodeCheck{58}{63}{15}%
  \RangeCatcodeCheck{64}{64}{12}%
  \RangeCatcodeCheck{65}{90}{11}%
  \RangeCatcodeCheck{91}{91}{15}%
  \RangeCatcodeCheck{92}{92}{0}%
  \RangeCatcodeCheck{93}{96}{15}%
  \RangeCatcodeCheck{97}{122}{11}%
  \RangeCatcodeCheck{123}{255}{15}%
  \RestoreCatcodes
}
\Test
\csname @@end\endcsname
\end
```

⟨/test1⟩

4 Installation

4.1 Download

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

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

[1] http://ctan.org/pkg/epstopdf

CTAN:macros/latex/contrib/oberdiek/epstopdf.pdf Documentation.

Bundle. All the packages of the bundle 'oberdiek' 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/oberdiek.tds.zip

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

4.2 Bundle installation

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

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

Script installation. Check the directory TDS:scripts/oberdiek/ for scripts that need further installation steps. Package attachfile2 comes with the Perl script pdfatfi.pl that should be installed in such a way that it can be called as pdfatfi. Example (linux):

```
chmod +x scripts/oberdiek/pdfatfi.pl
cp scripts/oberdiek/pdfatfi.pl /usr/local/bin/
```

4.3 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
```

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/oberdiek/epstopdf.sty
epstopdf-base.sty        → tex/latex/oberdiek/epstopdf-base.sty
epstopdf.pdf             → doc/latex/oberdiek/epstopdf.pdf
test/epstopdf-test1.tex → doc/latex/oberdiek/test/epstopdf-test1.tex
epstopdf.dtx             → source/latex/oberdiek/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.

4.4 Refresh file name databases

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

4.5 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
```

5 Catalogue

The following XML file can be used as source for the TeX Catalogue. The elements caption and description are imported from the original XML file from the Catalogue. The name of the XML file in the Catalogue is epstopdf-pkg.xml.

```
⟨*catalogue⟩
<?xml version='1.0' encoding='us-ascii'?>
<!DOCTYPE entry SYSTEM 'catalogue.dtd'>
<entry datestamp='$Date$' modifier='$Author$' id='epstopdf-pkg'>
  <name>epstopdf-pkg</name>
  <caption>Call epstopdf "on the fly"</caption>
  <authorref id='auth:oberdiek'/>
  <copyright owner='Heiko Oberdiek' year='2001,2006-2010'/>
  <license type='lppl1.3'/>
  <version number='2.6'/>
  <description>
    The package adds support for EPS files in the
    <xref refid='graphicx'>graphicx</xref> package when running under
    PDFTeX.  If an EPS graphic is detected, the package spawns a
    process to convert the EPS to PDF, using the script
    <xref refid='epstopdf'>epstopdf</xref>.  This of course requires
    that shell escape is enabled for the PDFTeX run.
    <p/>
    The package is part of the <xref refid='oberdiek'>oberdiek</xref>
    bundle.
  </description>
  <documentation details='Package documentation'
      href='ctan:/macros/latex/contrib/oberdiek/epstopdf.pdf'/>
  <ctan file='true' path='/macros/latex/contrib/oberdiek/epstopdf.dtx'/>
  <miktex location='oberdiek'/>
  <texlive location='oberdiek'/>
  <install path='/macros/latex/contrib/oberdiek/oberdiek.tds.zip'/>
</entry>
⟨/catalogue⟩
```

6 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

7 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.

Symbols

\# 562
\% 638
\@ 563, 636
\@PackageError 355, 505
\@PackageInfo 459, 512
\@PackageInfoNoLine 314, 469
\@PackageWarningNoLine 237, 250, 264, 301
\@cdr 440
\@ehc 363, 507
\@empty 411, 420, 440, 443, 475, 481, 525, 526
\@firstofone 298, 312, 320, 571, 574
\@firstoftwo 386
\@gobble 317, 568, 576
\@ifundefined 510
\@namedef 517
\@nil . . 326, 328, 330, 333, 341, 344, 440
\@secondoftwo 388
\@spaces 338, 348
\@undefined 58, 176, 541, 545
\\ 331, 345, 504, 637
\{ 560
\} 561

A

\advance 601, 609, 624
\aftergroup 29, 147
\AppendGraphicsExtensions . . 538, 547

B

\body 580, 584

C

\catcode 2, 3, 5, 6, 7, 8, 9, 10, 11, 12, 13, 33, 34, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 69, 70, 72, 73, 74, 78, 79, 80, 81, 82, 83, 84, 87, 88, 90, 91, 92, 93, 97, 99, 120, 121, 123, 124, 125, 126, 127, 128, 129, 130, 131, 151, 152, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 187, 188, 190, 191, 192, 196,

197, 198, 199, 200, 201, 202, 205, 206, 208, 209, 210, 211, 215, 217, 560, 561, 562, 563, 598, 607, 615, 619, 636, 637, 638
\CommandLine 394, 462, 467
\count@ 565, 594, 598, 600, 601, 605, 607, 608, 609, 613, 615, 618, 619, 623, 624
\countdef 565
\csname 14, 21, 50, 66, 76, 132, 139, 168, 184, 194, 236, 260, 293, 352, 368, 564, 567, 570, 573, 628, 655
\CurrentOption 114

D
\DeclareBoolOption . 271, 272, 278, 280
\DeclareComplementaryOption . . . 281
\DeclareOption 113
\DeclareStringOption . . . 276, 277, 279
\DeclareVoidOption . 274, 275, 542, 546

E
\empty 17, 18, 135, 136
\end 656
\endcsname 14, 21, 50, 66, 76, 132, 139, 168, 184, 194, 236, 260, 293, 352, 368, 564, 567, 570, 573, 628, 655
\endinput 29, 109, 147, 230
\endlinechar 4, 35, 71, 77, 89, 122, 153, 189, 195, 207
\epstopdfcall 384, 517
\epstopdfDeclareGraphicsRule . 503, 520
\epstopdfsetup 4, 114, 240, 254, 282, 283
\errmessage 617
\ETE@@Date 333, 336
\ETE@@Time 339, 341
\ETE@append 545
\ETE@AtEnd 95, 96, 109, 110, 112, 117, 213, 214, 230, 241, 255, 551
\ETE@Date 326, 330
\ETE@DefCommandLine 393, 457
\ETE@DefX 396, 409
\ETE@epstopdf 521, 523
\ETE@FileInfo . 307, 323, 460, 461, 470
\ETE@GenerateName . . . 444, 450, 500
\ETE@Gin@setfile 403, 499
\ETE@IfFileExists 366, 371, 422
\ETE@InsideSetfiletrue 408, 415
\ETE@Make 294, 309, 466
\ETE@OrgGin@setfile 353, 356, 405, 412, 424, 487
\ETE@outdir 442
\ETE@prepend 541
\ETE@prependfalse 275
\ETE@prependtrue 274
\ETE@program@epstopdf 524
\ETE@SavedAtEnd 110, 112
\ETE@Size 328, 344
\ETE@Skip 375, 423
\ETE@suffix . . . 420, 481, 484, 501, 526
\ETE@WarnModDate . . . 296, 300, 305

F
\filename@base 451, 478
\filename@parse 447

G
\Gin@base 391, 422, 424, 434, 437, 444, 447, 477, 483, 484
\Gin@driver 245, 248, 555
\Gin@eext 433, 434
\Gin@ext 391, 432, 436, 437
\Gin@setfile 353, 499
\GraphicsRead 417, 422, 490
\GraphicsType 416, 488
\grffile@IfFileExists 371, 372

I
\if 394, 400
\ifcase 525
\ifETE@disable 404
\ifETE@InsideSetfile 383, 385
\ifETE@prefersuffix 418
\ifETE@prepend 273, 535
\ifETE@update 295, 310
\ifETE@verbose 458, 468
\IfFileExists 369
\ifnum 262, 311, 600, 608, 615, 623
\ifx 15, 18, 21, 50, 58, 61, 133, 136, 139, 168, 176, 179, 236, 245, 248, 260, 293, 331, 345, 352, 368, 411, 420, 432, 443, 475, 481, 504, 525, 526, 564, 567, 570, 573, 628
\immediate 23, 52, 141, 170
\includegraphics 463
\input 554, 556, 629
\InputIfFileExists 549, 550
\iterate 581, 583, 585

L
\LoadCommand 629, 639
\loop 579, 595, 606, 614

M
\MessageBreak 302, 337, 347, 357, 360, 362, 460, 461, 462

N
\newcommand . . 240, 254, 282, 384, 503
\newif 273, 383
\next 427, 429, 585, 587, 589
\number 620

O
\OutputDirectory 442, 443, 451, 475, 478, 525
\OutputFile 444, 449, 461, 466, 470, 492, 531

P
\PackageInfo 26, 144
\pdf@filemoddate 311, 326
\pdf@filesize 328
\pdf@shellescape 262
\pdf@strcmp 311
\pdf@system 467

\pdffilemoddate 302
\PrependGraphicsExtensions . . 536, 543
\ProcessOptions 116
\ProvidesPackage 19, 67, 137, 185

R

\RangeCatcodeCheck 612, 640, 641, 642, 643, 644, 645, 646, 647, 648, 649, 650, 651
\RangeCatcodeInvalid 604, 632, 633, 634, 635
\repeat 579, 591, 602, 610, 625
\RequirePackage 111, 231, 232, 233, 234, 361
\resetatcatcode 557
\RestoreCatcodes . . 593, 596, 597, 652

S

\setkeys 282
\SetupKeyvalOptions 270
\SourceExt 289, 433, 436, 439, 440
\SourceFile 434, 437, 460, 466
\space 302, 338, 348, 356, 524, 531, 618, 619, 627

T

\Test 631, 654
\the 77, 78, 79, 80, 81, 82, 83, 84, 97, 195, 196, 197, 198, 199, 200, 201, 202, 215, 598, 618, 619
\TMP@EnsureCode 94, 101, 102, 103, 104, 105, 106, 107, 108, 212, 219, 220, 221, 222, 223, 224, 225, 226, 227, 228, 229

W

\write 23, 52, 141, 170

X

\x 14, 15, 18, 22, 26, 28, 51, 56, 66, 75, 87, 132, 133, 136, 140, 144, 146, 169, 174, 184, 193, 205, 244, 245, 247, 248, 375, 398, 411, 448, 455, 474, 495