The **epstopdf** package

Heiko Oberdiek
<oberdiek@uni-freiburg.de>

2009/10/17 v2.4

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

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:

$\boxed{\texttt{\textbackslash epstopdfsetup} \{\langle \textit{key value list}\rangle\}}$

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` $\Rightarrow$ `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
}
```

5

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
 3   \catcode44 12 % ,
 4   \catcode45 12 % -
 5   \catcode46 12 % .
 6   \catcode58 12 % :
 7   \catcode64 11 % @
 8   \catcode123 1 % {
 9   \catcode125 2 % }
10   \expandafter\let\expandafter\x\csname ver@epstopdf.sty\endcsname
11   \ifx\x\relax % plain-TeX, first loading
12   \else
13     \def\empty{}%
14     \ifx\x\empty % LaTeX, first loading,
15       % variable is initialized, but \ProvidesPackage not yet seen
16     \else
17       \catcode35 6 % #
18       \expandafter\ifx\csname PackageInfo\endcsname\relax
19         \def\x#1#2{%
20           \immediate\write-1{Package #1 Info: #2.}%
21         }%
22       \else
23         \def\x#1#2{\PackageInfo{#1}{#2, stopped}}%
24       \fi
25       \x{epstopdf}{The package is already loaded}%
26       \aftergroup\endinput
27     \fi
28   \fi
29 \endgroup
```

Package identification:

```
30 \begingroup
31   \catcode35 6 % #
32   \catcode40 12 % (
33   \catcode41 12 % )
34   \catcode44 12 % ,
35   \catcode45 12 % -
36   \catcode46 12 % .
37   \catcode47 12 % /
38   \catcode58 12 % :
39   \catcode64 11 % @
40   \catcode91 12 % [
41   \catcode93 12 % ]
42   \catcode123 1 % {
43   \catcode125 2 % }
44   \expandafter\ifx\csname ProvidesPackage\endcsname\relax
45     \def\x#1#2#3[#4]{\endgroup
46       \immediate\write-1{Package: #3 #4}%
```

```
47       \xdef#1{#4}%
48     }%
49   \else
50     \def\x#1#2[#3]{\endgroup
51       #2[{#3}]%
52       \ifx#1\@undefined
53         \xdef#1{#3}%
54       \fi
55       \ifx#1\relax
56         \xdef#1{#3}%
57       \fi
58     }%
59   \fi
60 \expandafter\x\csname ver@epstopdf.sty\endcsname
61 \ProvidesPackage{epstopdf}%
62   [2009/10/17 v2.4 Conversion with epstopdf on the fly (HO)]

63 \begingroup
64   \catcode123 1 % {
65   \catcode125 2 % }
66   \def\x{\endgroup
67     \expandafter\edef\csname ETE@AtEnd\endcsname{%
68       \catcode35 \the\catcode35\relax
69       \catcode64 \the\catcode64\relax
70       \catcode123 \the\catcode123\relax
71       \catcode125 \the\catcode125\relax
72     }%
73   }%
74 \x
75 \catcode35 6 % #
76 \catcode64 11 % @
77 \catcode123 1 % {
78 \catcode125 2 % }
79 \def\TMP@EnsureCode#1#2{%
80   \edef\ETE@AtEnd{%
81     \ETE@AtEnd
82     \catcode#1 \the\catcode#1\relax
83   }%
84   \catcode#1 #2\relax
85 }
86 \TMP@EnsureCode{42}{12}% *
87 \TMP@EnsureCode{45}{12}% -
88 \TMP@EnsureCode{47}{12}% /

89 \let\ETE@SavedAtEnd\ETE@AtEnd
90 \RequirePackage{epstopdf-base}[2009/10/17]
91 \let\ETE@AtEnd\ETE@SavedAtEnd
```

2.1.1   Option handling

```
92 \DeclareOption*{%
93   \expandafter\epstopdfsetup\expandafter{\CurrentOption}%
94 }
95 \ProcessOptions*\relax

96 \ETE@AtEnd
97 ⟨/package⟩
```

2.2   Base package

```
98 ⟨*base⟩
```

2.3 Preparations

2.3.1 Reload check and identification

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

```
 99 \begingroup
100   \catcode44 12 % ,
101   \catcode45 12 % -
102   \catcode46 12 % .
103   \catcode58 12 % :
104   \catcode64 11 % @
105   \catcode123 1 % {
106   \catcode125 2 % }
107   \expandafter\let\expandafter\x\csname ver@epstopdf-base.sty\endcsname
108   \ifx\x\relax % plain-TeX, first loading
109   \else
110     \def\empty{}%
111     \ifx\x\empty % LaTeX, first loading,
112       % variable is initialized, but \ProvidesPackage not yet seen
113     \else
114       \catcode35 6 % #
115       \expandafter\ifx\csname PackageInfo\endcsname\relax
116         \def\x#1#2{%
117           \immediate\write-1{Package #1 Info: #2.}%
118         }%
119       \else
120         \def\x#1#2{\PackageInfo{#1}{#2, stopped}}%
121       \fi
122       \x{epstopdf-base}{The package is already loaded}%
123       \aftergroup\endinput
124     \fi
125   \fi
126 \endgroup
```

Package identification:

```
127 \begingroup
128   \catcode35 6 % #
129   \catcode40 12 % (
130   \catcode41 12 % )
131   \catcode44 12 % ,
132   \catcode45 12 % -
133   \catcode46 12 % .
134   \catcode47 12 % /
135   \catcode58 12 % :
136   \catcode64 11 % @
137   \catcode91 12 % [
138   \catcode93 12 % ]
139   \catcode123 1 % {
140   \catcode125 2 % }
141   \expandafter\ifx\csname ProvidesPackage\endcsname\relax
142     \def\x#1#2#3[#4]{\endgroup
143       \immediate\write-1{Package: #3 #4}%
144       \xdef#1{#4}%
145     }%
146   \else
147     \def\x#1#2[#3]{\endgroup
148       #2[{#3}]%
149       \ifx#1\@undefined
150         \xdef#1{#3}%
151       \fi
152       \ifx#1\relax
153         \xdef#1{#3}%
154       \fi
```

```
155     }%
156   \fi
157 \expandafter\x\csname ver@epstopdf-base.sty\endcsname
158 \ProvidesPackage{epstopdf-base}%
159   [2009/10/17 v2.4 Base part for package epstopdf]
```

2.3.2 Catcodes

```
160 \begingroup
161   \catcode123 1 % {
162   \catcode125 2 % }
163   \def\x{\endgroup
164     \expandafter\edef\csname ETE@AtEnd\endcsname{%
165       \catcode35 \the\catcode35\relax
166       \catcode64 \the\catcode64\relax
167       \catcode123 \the\catcode123\relax
168       \catcode125 \the\catcode125\relax
169     }%
170   }%
171 \x
172 \catcode35 6 % #
173 \catcode64 11 % @
174 \catcode123 1 % {
175 \catcode125 2 % }
176 \def\TMP@EnsureCode#1#2{%
177   \edef\ETE@AtEnd{%
178     \ETE@AtEnd
179     \catcode#1 \the\catcode#1\relax
180   }%
181   \catcode#1 #2\relax
182 }
183 \TMP@EnsureCode{33}{12}% !
184 \TMP@EnsureCode{39}{12}% '
185 \TMP@EnsureCode{42}{12}% *
186 \TMP@EnsureCode{44}{12}% ,
187 \TMP@EnsureCode{45}{12}% -
188 \TMP@EnsureCode{46}{12}% .
189 \TMP@EnsureCode{47}{12}% /
190 \TMP@EnsureCode{58}{12}% :
191 \TMP@EnsureCode{60}{12}% <
192 \TMP@EnsureCode{61}{12}% =
193 \TMP@EnsureCode{62}{12}% >
194 \TMP@EnsureCode{96}{12}% `
```

2.3.3 Load packages

```
195 \RequirePackage{infwarerr}[2007/09/09]
196 \RequirePackage{grfext}\relax
197 \RequirePackage{kvoptions}[2007/10/02]
198 \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.

```
199 \begingroup\expandafter\expandafter\expandafter\endgroup
200 \expandafter\ifx\csname Gin@setfile\endcsname\relax
201   \@PackageWarningNoLine{epstopdf}{%
202     No graphics package \string`graphic{s,x}\string' loaded%
203   }%
204   \newcommand*{\epstopdfsetup}[1]{}%
205   \ETE@AtEnd
206   \expandafter\endinput
207 \fi
```

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

```
208 \begingroup
209   \def\x{pdftex.def}%
210   \ifx\Gin@driver\x
211   \else
212     \@PackageWarningNoLine{epstopdf}{%
213       Other drivers than 'pdftex.def' are not supported%
214     }%
215     \endgroup
216     \newcommand*{\epstopdfsetup}[1]{}%
217     \ETE@AtEnd
218     \expandafter\endinput
219   \fi
220 \endgroup
```

Check, whether the shell escape feature is enabled.

```
221 \begingroup
222   \expandafter\ifx\csname pdf@shellescape\endcsname\relax
223   \else
224     \ifnum\pdf@shellescape>0 %
225     \else
226       \@PackageWarningNoLine{epstopdf}{%
227         Shell escape feature is not enabled%
228       }%
229     \fi
230   \fi
231 \endgroup
```

2.5 Package loading

2.6 Options

```
232 \SetupKeyvalOptions{family=ETE,prefix=ETE@}
233 \DeclareBoolOption{update}
234 \DeclareBoolOption{verbose}
235 \newif\ifETE@prepend
236 \DeclareVoidOption{prepend}{\ETE@prependtrue}
237 \DeclareVoidOption{append}{\ETE@prependfalse}
238 \DeclareStringOption{outdir}
239 \DeclareStringOption{suffix}
240 \DeclareBoolOption{prefersuffix}
241 \DeclareStringOption{program@epstopdf}
```

Options disable and enable are for testing only. Therefore they are not documented on purpose.
```
242 \DeclareBoolOption{disable}
243 \DeclareComplementaryOption{enable}{disable}
```

```
244 \newcommand*{\epstopdfsetup}{\setkeys{ETE}}
```

2.6.1 Default setting

```
245 \epstopdfsetup{%
246   verbose,%
247   enable,%
248   append,%
249   update,%
250   prefersuffix,%
251   suffix=-\SourceExt-converted-to,%
252   program@epstopdf=epstopdf%
253 }
```

2.7 Make and verbose

```
254 \begingroup\expandafter\expandafter\expandafter\endgroup
255 \expandafter\ifx\csname pdf@filemoddate\endcsname\relax
256   \def\ETE@Make#1#2{%
257     \ifETE@update
258       \ETE@WarnModDate
259     \fi
260     \@firstofone
261   }%
262   \def\ETE@WarnModDate{%
263     \@PackageWarningNoLine{epstopdf}{%
264       \string\pdffilemoddate\space is not available,\MessageBreak
265       option 'update' will be ignored%
266     }%
267     \global\let\ETE@WarnModDate\relax
268   }%
269   \def\ETE@FileInfo#1#2{#1 file: <#2>}%
270 \else
271   \def\ETE@Make#1#2{%
272     \ifETE@update
273       \ifnum\pdf@strcmp{\pdf@filemoddate{#1}}{\pdf@filemoddate{#2}}>0 %
274         \expandafter\expandafter\expandafter\@firstofone
275       \else
276         \@PackageInfoNoLine{epstopdf}{%
277           Output file is already uptodate%
278         }%
279         \expandafter\expandafter\expandafter\@gobble
280       \fi
281     \else
282       \expandafter\@firstofone
283     \fi
284   }%
285   \def\ETE@FileInfo#1#2{%
286     #1 file: <#2>%
287     \expandafter\expandafter\expandafter
288     \ETE@Date\pdf@filemoddate{#2}\@nil
289     \expandafter\expandafter\expandafter
290     \ETE@Size\pdf@filesize{#2}\@nil
291   }%
292   \def\ETE@Date#1\@nil{%
293     \ifx\\#1\\%
294     \else
295       \ETE@@Date#1\@nil
296     \fi
297   }%
298   \def\ETE@@Date#1:#2#3#4#5#6#7#8#9{%
299     \MessageBreak
300     \@spaces\space\space\space date: #2#3#4#5-#6#7-#8#9 %
301     \ETE@@Time
302   }%
303   \def\ETE@@Time#1#2#3#4#5#6#7\@nil{%
304     #1#2:#3#4:#5#6%
305   }%
306   \def\ETE@Size#1\@nil{%
307     \ifx\\#1\\%
308     \else
309       \MessageBreak
310       \@spaces\space\space\space size: #1 bytes%
311     \fi
312   }%
313 \fi
```

2.8 Adding conversion support

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

```
314 \expandafter\ifx\csname ETE@OrgGin@setfile\endcsname\relax
315   \let\ETE@OrgGin@setfile\Gin@setfile
316 \else
317   \@PackageError{epstopdf}{%
318     Command \string\ETE@OrgGin@setfile\space
319     already defined.\MessageBreak
320   }{%
321     Probably some package has included the code of this package%
322     \MessageBreak
323     instead of using \string\RequirePackage{epstopdf}.%
324     \MessageBreak
325     \@ehc
326   }%
327 \fi
328 \def\ETE@IfFileExists{%
329   \begingroup\expandafter\expandafter\expandafter\endgroup
330   \expandafter\ifx\csname grffile@IfFileExists\endcsname\relax
331     \expandafter\IfFileExists
332   \else
333     \global\let\ETE@IfFileExists\grffile@IfFileExists
334     \expandafter\grffile@IfFileExists
335   \fi
336 }
337 \def\ETE@Skip#1#2\x\fi\fi{%
338   \fi
339   \fi
340   \endgroup
341   \fi
342   \fi
343   #1%
344 }
345 \newif\ifETE@InsideSetfile
346 \newcommand*{\epstopdfcall}[1]{%
347   \ifETE@InsideSetfile
348     \expandafter\@firstoftwo
349   \else
350     \expandafter\@secondoftwo
351   \fi
352   {`#1}%
353   {\Gin@base\Gin@ext}%
354 }
355 \def\ETE@DefCommandLine#1{%
356   \edef\CommandLine{\expandafter\fi\if`#1}%
357 }
358 \def\ETE@DefX#1{%
359   \expandafter\expandafter\expandafter\def
360   \expandafter\expandafter\expandafter\x
361   \expandafter\expandafter\expandafter{%
362     \expandafter\fi\if`#1\relax\else
363   }%
364 }
365 \def\Gin@setfile#1#2#3{%
366   \ifETE@disable
367     \ETE@OrgGin@setfile{#1}{#2}{#3}%
368   \else
369     \begingroup
370       \ETE@InsideSetfiletrue
371       \ETE@DefX{#3}%
372     \expandafter\endgroup
373     \ifx\x\@empty
```

```
374        \ETE@OrgGin@setfile{#1}{#2}{#3}%
375      \else
376        \begingroup
377          \ETE@InsideSetfiletrue
378          \def\GraphicsType{#1}%
379          \def\GraphicsRead{#2}%
380          \ifETE@prefersuffix
381          \else
382            \ifx\ETE@suffix\@empty
383            \else
384              \ETE@IfFileExists{\Gin@base\GraphicsRead}{%
385                \ETE@Skip{%
386                  \ETE@OrgGin@setfile{#1}{#2}{\Gin@base#2}%
387                }%
388              }{%
389                \let\next\relax
390              }%
391              \next
392            \fi
393          \fi
394          \ifx\Gin@ext\relax
395            \let\SourceExt\Gin@eext
396            \def\SourceFile{\Gin@base\Gin@eext}%
397          \else
398            \let\SourceExt\Gin@ext
399            \def\SourceFile{\Gin@base\Gin@ext}%
400          \fi
401          \edef\SourceExt{% remove dot
402            \expandafter\@cdr\SourceExt\@empty\@nil
403          }%
404          \let\OutputDirectory\ETE@outdir
405          \ifx\OutputDirectory\@empty
406            \edef\OutputFile{\ETE@GenerateName{\Gin@base}{#2}}%
407          \else
408            \begingroup
409              \filename@parse{\Gin@base#2}%
410              \edef\x{\endgroup
411                \def\noexpand\OutputFile{%
412                  \ETE@GenerateName{%
413                    \OutputDirectory\filename@base
414                  }{#2}%
415                }%
416              }%
417              \x
418          \fi
419          \ETE@DefCommandLine{#3}%
420          \ifETE@verbose
421            \@PackageInfo{epstopdf}{%
422              \ETE@FileInfo{Source}\SourceFile\MessageBreak
423              \ETE@FileInfo{Output}\OutputFile\MessageBreak
424              Command: <\CommandLine>\MessageBreak
425              \string\includegraphics
426            }%
427          \fi
428          \ETE@Make\SourceFile\OutputFile{%
429            \pdf@system{\CommandLine}%
430            \ifETE@verbose
431              \@PackageInfoNoLine{epstopdf}{%
432                \ETE@FileInfo{Result}\OutputFile
433              }%
434            \fi
435          }%
```

```
436        \edef\x{\endgroup
437          \ifx\OutputDirectory\@empty
438          \else
439            \def\noexpand\Gin@base{%
440              \OutputDirectory\noexpand\filename@base
441            }%
442          \fi
443          \ifx\ETE@suffix\@empty
444          \else
445            \edef\noexpand\Gin@base{%
446              \noexpand\Gin@base\ETE@suffix
447            }%
448          \fi
449          \noexpand\ETE@OrgGin@setfile{%
450            \GraphicsType
451          }{%
452            \GraphicsRead
453          }{%
454            \OutputFile
455          }%
456        }%
457      \x
458    \fi
459  \fi
460 }
461 \def\ETE@GenerateName#1#2{%
462   #1\ETE@suffix#2%
463 }
```

2.9   Declare graphics rule

```
464 \newcommand*{\epstopdfDeclareGraphicsRule}[4]{%
465   \ifx\\#4\\%
466     \@PackageError{epstopdf-base}{%
467       Conversion command is missing%
468     }\@ehc
469   \else
470     \begingroup
471       \@ifundefined{Gin@rule@#1}{%
472       }{%
473         \@PackageInfo{epstopdf-base}{%
474           Redefining graphics rule for '#1'%
475         }%
476       }%
477     \endgroup
478     \@namedef{Gin@rule@#1}##1{{#2}{#3}{\epstopdfcall{#4}}}%
479   \fi
480 }
```

\DeclareGraphicsRule for .eps
```
481 \epstopdfDeclareGraphicsRule{.eps}{pdf}{.pdf}{%
482   \ETE@epstopdf{#1}%
483 }
484 \def\ETE@epstopdf#1{%
485   \ETE@program@epstopdf\space
486   \ifcase\ifx\OutputDirectory\@empty
487            \ifx\ETE@suffix\@empty
488              1%
489            \fi
490          \fi
491          0 %
492    --outfile=\OutputFile\space
493   \fi
```

```
494    #1%
495 }
496 \ifETE@prepend
497    \expandafter\PrependGraphicsExtensions
498 \else
499    \expandafter\AppendGraphicsExtensions
500 \fi
501 {.eps}
502 \let\ETE@prepend\@undefined
503 \DeclareVoidOption{prepend}{%
504    \PrependGraphicsExtensions{.eps}%
505 }
506 \let\ETE@append\@undefined
507 \DeclareVoidOption{append}{%
508    \AppendGraphicsExtensions{.eps}%
509 }
510 \InputIfFileExists{epstopdf-sys.cfg}{}{}
511 \InputIfFileExists{epstopdf.cfg}{}{}
512 \ETE@AtEnd
513 ⟨/base⟩
```

3   Test

3.1   Preface for standard catcode check

```
514 ⟨*test1⟩
515 \input miniltx.tex\relax
516 \def\Gin@driver{pdftex.def}
517 \input graphicx.sty\relax
518 \resetatcatcode
519 ⟨/test1⟩
```

3.2   Catcode checks for loading

```
520 ⟨*test1⟩
521 \catcode`\{=1 %
522 \catcode`\}=2 %
523 \catcode`\#=6 %
524 \catcode`\@=11 %
525 \expandafter\ifx\csname count@\endcsname\relax
526    \countdef\count@=255 %
527 \fi
528 \expandafter\ifx\csname @gobble\endcsname\relax
529    \long\def\@gobble#1{}%
530 \fi
531 \expandafter\ifx\csname @firstofone\endcsname\relax
532    \long\def\@firstofone#1{#1}%
533 \fi
534 \expandafter\ifx\csname loop\endcsname\relax
535    \expandafter\@firstofone
536 \else
537    \expandafter\@gobble
538 \fi
539 {%
540    \def\loop#1\repeat{%
541       \def\body{#1}%
542       \iterate
543    }%
544    \def\iterate{%
545       \body
546          \let\next\iterate
547       \else
```

```
548      \let\next\relax
549    \fi
550    \next
551  }%
552  \let\repeat=\fi
553 }%
554 \def\RestoreCatcodes{}
555 \count@=0 %
556 \loop
557   \edef\RestoreCatcodes{%
558     \RestoreCatcodes
559     \catcode\the\count@=\the\catcode\count@\relax
560   }%
561 \ifnum\count@<255 %
562   \advance\count@ 1 %
563 \repeat
564
565 \def\RangeCatcodeInvalid#1#2{%
566   \count@=#1\relax
567   \loop
568     \catcode\count@=15 %
569   \ifnum\count@<#2\relax
570     \advance\count@ 1 %
571   \repeat
572 }
573 \expandafter\ifx\csname LoadCommand\endcsname\relax
574   \def\LoadCommand{\input epstopdf.sty\relax}%
575 \fi
576 \def\Test{%
577   \RangeCatcodeInvalid{0}{47}%
578   \RangeCatcodeInvalid{58}{64}%
579   \RangeCatcodeInvalid{91}{96}%
580   \RangeCatcodeInvalid{123}{255}%
581   \catcode'\@=12 %
582   \catcode'\\=0 %
583   \catcode'\{=1 %
584   \catcode'\}=2 %
585   \catcode'\#=6 %
586   \catcode'\[=12 %
587   \catcode'\]=12 %
588   \catcode'\%=14 %
589   \catcode'\ =10 %
590   \catcode13=5 %
591   \LoadCommand
592   \RestoreCatcodes
593 }
594 \Test
595 \csname @@end\endcsname
596 \end
597 ⟨/test1⟩
```

4   Installation

4.1   Download

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

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

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

---

[1]ftp://ftp.ctan.org/tex-archive/

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

Attached source. The PDF documentation on CTAN also includes the `.dtx` source file. It can be extracted by AcrobatReader 6 or higher. Another option is `pdftk`, e.g. unpack the file into the current directory:

```
pdftk epstopdf.pdf unpack_files output .
```

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 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.

6 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; numbers in roman refer to the code lines where the entry is used.

21