

The **epstopdf** package

Heiko Oberdiek
<oberdiek@uni-freiburg.de>

2009/07/16 v2.2

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	3
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	5
2	Implementation	6
2.1	Wrapper package	6
2.1.1	Option handling	7
2.2	Base package	7
2.3	Preparations	8
2.3.1	Relead check and identification	8
2.3.2	Catcodes	9
2.3.3	Load packages	9
2.4	Checks	9
2.5	Package loading	10
2.6	Options	10
2.6.1	Default setting	10
2.7	Make and verbose	10
2.8	Adding conversion support	12
2.9	Declare graphics rule	14
3	Test	15
3.1	Preface for standard catcode check	15
3.2	Catcode checks for loading	15
4	Installation	16
4.1	Download	16
4.2	Bundle installation	17
4.3	Package installation	17
4.4	Refresh file name databases	17
4.5	Some details for the interested	17

5	History	18
	[2001/01/06 v1.0]	18
	[2001/02/04 v1.1]	18
	[2006/02/20 v1.2]	18
	[2006/08/26 v1.3]	18
	[2007/04/26 v1.4]	18
	[2007/10/02 v1.5]	19
	[2007/11/11 v1.6]	19
	[2008/05/06 v1.7]	19
	[2009/03/01 v1.8]	19
	[2009/07/06 v1.9]	19
	[2009/07/07 v1.10]	19
	[2009/07/12 v2.0]	19
	[2009/07/15 v2.1]	19
	[2009/07/16 v2.2]	19
6	Index	20

1 Documentation

1.1 Introduction

L^AT_EX 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 usable. 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, pdf_TE_X 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 pdf_TE_X. This feature is called "shell escape" or "write 18" and must usually enabled explicitly 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 T_EX's compile run. Keep in mind that this is a security risk. The feature is an addition to T_EX. MikT_EX, teT_EX, T_EX Live support it. In Web2C based T_EX distributions (teT_EX, T_EX 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 than 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⟩}`

L^AT_EX 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`. L^AT_EX'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 'regenerate' 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`.

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 -i .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 L^AT_EX.

```
2 \begingroup
3 \catcode44 12 % ,
4 \catcode45 12 % -
5 \catcode46 12 % .
6 \catcode58 12 % :
7 \catcode64 11 % @
8 \expandafter\let\expandafter\x\csname ver@epstopdf.sty\endcsname
9 \ifcase 0%
10 \ifx\x\relax % plain
11 \else
12 \ifx\x\empty % LaTeX
13 \else
14 1%
15 \fi
16 \fi
17 \else
18 \catcode35 6 % #
19 \catcode123 1 % {
20 \catcode125 2 % }
21 \expandafter\ifx\csname PackageInfo\endcsname\relax
22 \def\x#1#2{%
23 \immediate\write-1{Package #1 Info: #2.}%
24 }%
25 \else
26 \def\x#1#2{\PackageInfo{#1}{#2, stopped}}%
27 \fi
28 \x{epstopdf}{The package is already loaded}%
29 \endgroup
30 \expandafter\endinput
31 \fi
32 \endgroup
```

Package identification:

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

```

47     \immediate\write-1{Package: #3 #4}%
48     \xdef#1{#4}%
49 }%
50 \else
51   \def\x#1#2[#3]{\endgroup
52     #2[{#3}]}%
53   \ifx#1\@undefined
54     \xdef#1{#3}%
55   \fi
56   \ifx#1\relax
57     \xdef#1{#3}%
58   \fi
59 }%
60 \fi
61 \expandafter\x\csname ver@epstopdf.sty\endcsname
62 \ProvidesPackage{epstopdf}%
63 [2009/07/16 v2.2 Conversion with epstopdf on the fly (HO)]

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

90 \let\ETE@SavedAtEnd\ETE@AtEnd
91 \RequirePackage{epstopdf-base}[2009/07/16]
92 \let\ETE@AtEnd\ETE@SavedAtEnd

```

2.1.1 Option handling

```

93 \DeclareOption*{%
94   \expandafter\epstopdfsetup\expandafter{\CurrentOption}%
95 }
96 \ProcessOptions*\relax
97 \ETE@AtEnd
98 \</package>

```

2.2 Base package

```

99 <*base>

```

2.3 Preparations

2.3.1 Relead check and identification

Reload check, especially if the package is not used with L^AT_EX.

```
100 \begingroup
101   \catcode44 12 % ,
102   \catcode45 12 % -
103   \catcode46 12 % .
104   \catcode58 12 % :
105   \catcode64 11 % @
106   \expandafter\let\expandafter\x\csname ver@epstopdf-base.sty\endcsname
107   \ifcase 0%
108     \ifx\x\relax % plain
109     \else
110       \ifx\x\empty % LaTeX
111       \else
112         1%
113       \fi
114     \fi
115   \else
116     \catcode35 6 % #
117     \catcode123 1 % {
118     \catcode125 2 % }
119     \expandafter\ifx\csname PackageInfo\endcsname\relax
120       \def\x#1#2{%
121         \immediate\write-1{Package #1 Info: #2.}%
122       }%
123     \else
124       \def\x#1#2{\PackageInfo{#1}{#2, stopped}}%
125     \fi
126     \x{epstopdf-base}{The package is already loaded}%
127   \endgroup
128   \expandafter\endinput
129 \fi
130 \endgroup
```

Package identification:

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



```

156      \fi
157    }%
158  \fi
159 \expandafter\x\csname ver@epstopdf-base.sty\endcsname
160 \ProvidesPackage{epstopdf-base}%
161 [2009/07/16 v2.2 Base part for package epstopdf]

```

2.3.2 Catcodes

```

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

```

2.3.3 Load packages

```

197 \RequirePackage{infwarerr}[2007/09/09]
198 \RequirePackage{grfext}\relax
199 \RequirePackage{kvoptions}[2007/10/02]
200 \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.

```

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

```

```

209 \fi

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

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

    Check, whether the shell escape feature is enabled.

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

```

2.5 Package loading

2.6 Options

```

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

```

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

```

243 \DeclareBoolOption{disable}
244 \DeclareComplementaryOption{enable}{disable}
245 \newcommand*{\epstopdfsetup}{\setkeys{ETE}}

```

2.6.1 Default setting

```

246 \epstopdfsetup{%
247   verbose,%
248   enable,%
249   append,%
250   update,%
251   prefersuffix,%
252   suffix=-\SourceExt-converted-to%
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\pdf@filemoddate\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{\Gin@base\ETE@suffix}%
446 \fi
447 \noexpand\ETE@OrgGin@setfile{%
448 \GraphicsType
449 }{%
450 \GraphicsRead
451 }{%
452 \OutputFile
453 }%
454 }%
455 \x
456 \fi
457 \fi
458 }
459 \def\ETE@GenerateName#1#2{%
460 #1\ETE@suffix#2%
461 }

```

2.9 Declare graphics rule

```

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

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

```

```

494 \ifETE@prepend
495   \expandafter\PrependGraphicsExtensions
496 \else
497   \expandafter\AppendGraphicsExtensions
498 \fi
499 {.eps}
500 \let\ETE@prepend\@undefined
501 \DeclareVoidOption{prepend}{%
502   \PrependGraphicsExtensions{.eps}%
503 }
504 \let\ETE@append\@undefined
505 \DeclareVoidOption{append}{%
506   \AppendGraphicsExtensions{.eps}%
507 }

508 \InputIfFileExists{epstopdf-sys.cfg}{-}{-}
509 \InputIfFileExists{epstopdf.cfg}{-}{-}

510 \ETE@AtEnd
511 </base>

```

3 Test

3.1 Preface for standard catcode check

```

512 <*test1>
513 \input miniltx.tex\relax
514 \def\Gin@driver{pdftex.def}
515 \input graphicx.sty\relax
516 \resetatcatcode
517 </test1>

```

3.2 Catcode checks for loading

```

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

```

```

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

```

4 Installation

4.1 Download

Package. This package is available on CTAN¹:

[CTAN:macros/latex/contrib/oberdiek/epstopdf.dtx](http://ctan.org/macros/latex/contrib/oberdiek/epstopdf.dtx) The source file.

[CTAN:macros/latex/contrib/oberdiek/epstopdf.pdf](http://ctan.org/macros/latex/contrib/oberdiek/epstopdf.pdf) Documentation.

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

```
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):

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

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 T_EX distribution (teT_EX, mikT_EX, ...) relies on file name databases, you must refresh these. For example, teT_EX users run `texhash` or `mktextlsr`.

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 L^AT_EX. The .dtx chooses its action depending on the format:

plain-T_EX: Run docstrip and extract the files.

L^AT_EX: Generate the documentation.

If you insist on using L^AT_EX for docstrip (really, docstrip does not need L^AT_EX), 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 pdfL^AT_EX:

```
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 pdfT_EX mailing list.

[2001/02/04 v1.1]

- Minor documentation update.
- CTAN.

[2006/02/20 v1.2]

- DTX framework.
- Compatibility for minil_TX.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 `pdfcmds` for L^AT_EX 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 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.

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.

Symbols	
<code>\#</code>	521, 583
<code>\%</code>	586
<code>\@</code>	522, 579
<code>\@PackageError</code>	317, 464
<code>\@PackageInfo</code>	421, 471
<code>\@PackageInfoNoLine</code>	276, 431
<code>\@PackageWarningNoLine</code>	203, 214, 228, 263
<code>\@cdr</code>	402
<code>\@ehc</code>	325, 466
<code>\@empty</code>	373,
	382, 402, 405, 437, 443, 484, 485
<code>\@firstofone</code> ..	260, 274, 282, 530, 533
<code>\@firstoftwo</code>	348
<code>\@gobble</code>	279, 527, 535
<code>\@ifundefined</code>	469
<code>\@namedef</code>	476
<code>\@nil</code> ..	288, 290, 292, 295, 303, 306, 402
<code>\@secondoftwo</code>	350
<code>\@spaces</code>	300, 310
<code>\@undefined</code>	53, 151, 500, 504
<code>\[</code>	584
<code>\]</code>	293, 307, 463, 580
<code>\{</code>	519, 581
<code>\}</code>	520, 582
<code>\]</code>	585
<code>_</code>	587
A	
<code>\advance</code>	560, 568
<code>\AppendGraphicsExtensions</code> ..	497, 506
B	
<code>\body</code>	539, 543
C	
<code>\catcode</code>	3, 4, 5,
	6, 7, 18, 19, 20, 34, 35, 36, 37,
	38, 39, 40, 41, 42, 43, 44, 65,
	66, 69, 70, 71, 72, 76, 77, 78,
	79, 83, 85, 101, 102, 103, 104,
	105, 116, 117, 118, 132, 133,
	134, 135, 136, 137, 138, 139,
	140, 141, 142, 163, 164, 167,
	168, 169, 170, 174, 175, 176,
	177, 181, 183, 519, 520, 521,
	522, 557, 566, 579, 580, 581,
	582, 583, 584, 585, 586, 587, 588
<code>\CommandLine</code>	356, 424, 429
<code>\count@</code>	524, 553,
	557, 559, 560, 564, 566, 567, 568
<code>\countdef</code>	524
<code>\csname</code> 8, 21, 45, 61, 68, 106, 119, 143,	
	159, 166, 202, 224, 255, 314,
	330, 523, 526, 529, 532, 571, 593
<code>\CurrentOption</code>	94
D	
<code>\DeclareBoolOption</code> ..	235, 236, 242, 243
<code>\DeclareComplementaryOption</code>	244
<code>\DeclareOption</code>	93
<code>\DeclareStringOption</code>	240, 241
<code>\DeclareVoidOption</code> ..	238, 239, 501, 505
E	
<code>\empty</code>	12, 110
<code>\end</code>	594
<code>\endcsname</code>	8,
	21, 45, 61, 68, 106, 119, 143,
	159, 166, 202, 224, 255, 314,
	330, 523, 526, 529, 532, 571, 593
<code>\endinput</code>	30, 128, 208, 220
<code>\epstopdfcall</code>	346, 476
<code>\epstopdfDeclareGraphicsRule</code> 462, 479	
<code>\epstopdfsetup</code> 4, 94, 206, 218, 245, 246	
<code>\ETE@@Date</code>	295, 298
<code>\ETE@@Time</code>	301, 303
<code>\ETE@append</code>	504
<code>\ETE@AtEnd</code>	81, 82,
	90, 92, 97, 179, 180, 207, 219, 510
<code>\ETE@Date</code>	288, 292
<code>\ETE@DefCommandLine</code>	355, 419
<code>\ETE@DefX</code>	358, 371
<code>\ETE@epstopdf</code>	480, 482
<code>\ETE@FileInfo</code> ..	269, 285, 422, 423, 432
<code>\ETE@GenerateName</code>	406, 412, 459
<code>\ETE@IfFileExists</code>	328, 333, 384
<code>\ETE@InsideSetfiletrue</code>	370, 377
<code>\ETE@Make</code>	256, 271, 428
<code>\ETE@OrgGin@setfile</code>	
	315, 318, 367, 374, 386, 447
<code>\ETE@outdir</code>	404
<code>\ETE@prepend</code>	500
<code>\ETE@prependfalse</code>	239
<code>\ETE@prependtrue</code>	238
<code>\ETE@SavedAtEnd</code>	90, 92
<code>\ETE@Size</code>	290, 306
<code>\ETE@Skip</code>	337, 385
<code>\ETE@suffix</code> ...	382, 443, 445, 460, 485
<code>\ETE@WarnModDate</code>	258, 262, 267
F	
<code>\filename@base</code>	413, 440
<code>\filename@parse</code>	409
G	
<code>\Gin@base</code>	353, 384,
	386, 396, 399, 406, 409, 439, 445
<code>\Gin@driver</code>	212, 514

\Gin@eext	395, 396		
\Gin@ext	353, 394, 398, 399		
\Gin@setfile	315, 365		
\GraphicsRead	379, 384, 450		
\GraphicsType	378, 448		
\grffile@ifFileExists	333, 334		
I			
\if	356, 362		
\ifcase	9, 107, 484		
\ifETE@disable	366		
\ifETE@InsideSetfile	345, 347		
\ifETE@prefersuffix	380		
\ifETE@prepend	237, 494		
\ifETE@update	257, 272		
\ifETE@verbose	420, 430		
\IfFileExists	331		
\ifnum	226, 273, 559, 567		
\ifx	10,		
	12, 21, 45, 53, 56, 108, 110, 119,		
	143, 151, 154, 202, 212, 224,		
	255, 293, 307, 314, 330, 373,		
	382, 394, 405, 437, 443, 463,		
	484, 485, 523, 526, 529, 532, 571		
\immediate	23, 47, 121, 145		
\includegraphics	425		
\input	513, 515, 572		
\InputIfFileExists	508, 509		
\iterate	540, 542, 544		
L			
\LoadCommand	572, 589		
\loop	538, 554, 565		
M			
\MessageBreak	264, 299,		
	309, 319, 322, 324, 422, 423, 424		
N			
\newcommand	206, 218, 245, 346, 462		
\newif	237, 345		
\next	389, 391, 544, 546, 548		
O			
\OutputDirectory	404, 405, 413, 437, 440, 484		
\OutputFile	406, 411, 423, 428, 432, 452, 490		
P			
\PackageInfo	26, 124		
\pdf@filemoddate	273, 288		
\pdf@filesize	290		
\pdf@shellescape	226		
\pdf@strcmp	273		
\pdf@system	429		
\pdffilemoddate	264		
\PrependGraphicsExtensions	495, 502		
\ProcessOptions	96		
\ProvidesPackage	62, 160		
R			
\RangeCatcodeInvalid	563, 575, 576, 577, 578		
\repeat	538, 550, 561, 569		
\RequirePackage	91, 197, 198, 199, 200, 323		
\resetatcatcode	516		
\RestoreCatcodes	552, 555, 556, 590		
S			
\setkeys	245		
\SetupKeyvalOptions	234		
\SourceExt	252, 395, 398, 401, 402		
\SourceFile	396, 399, 422, 428		
\space	264, 300, 310, 318, 490		
T			
\Test	574, 592		
\the	69, 70, 71,		
	72, 83, 167, 168, 169, 170, 181, 557		
\TMP@EnsureCode	80, 87, 88,		
	89, 178, 185, 186, 187, 188, 189,		
	190, 191, 192, 193, 194, 195, 196		
W			
\write	23, 47, 121, 145		
X			
\x	8, 10, 12, 22,		
	26, 28, 46, 51, 61, 67, 75, 106,		
	108, 110, 120, 124, 126, 144,		
	149, 159, 165, 173, 211, 212,		
	337, 360, 373, 410, 417, 436, 455		