

The **epstopdf** package

Heiko Oberdiek
<oberdiek@uni-freiburg.de>

2009/07/12 v2.0

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	4
1.5.1	System configuration file epstopdf-sys.cfg	4
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	7
2.3.1	Relead check and identification	7
2.3.2	Catcodes	8
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	11
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	16
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]	18
[2007/11/11 v1.6]	18
[2008/05/06 v1.7]	18
[2009/03/01 v1.8]	18
[2009/07/06 v1.9]	19
[2009/07/07 v1.10]	19
[2009/07/12 v2.0]	19
6 Index	19

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 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, 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 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 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 (TeX Live) it can be enabled in the configuration file `texmf.cnf`:

```
shell_escape = 1
```

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

```
--shell-escape (TeX Live)
--enable-write18 (MikTeX)
```

Example:

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

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

```

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

```

1.3 Usage

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

```

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

```

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

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

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

```

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

```

Package `epstopdf` puts `.eps` at the end of the graphics extension search list. This is the behaviour of option `append` that is enabled by default. That means, the conversion is called last unless a supported file type cannot be found earlier. This avoids unnecessary conversion steps that slow down the \LaTeX run. If you want to use option `update` and your pdf\TeX supports it, then an outdated PDF file also would be found earlier. Therefore extension `.eps` should be put in front of the list. This is achieved by option `prepend`. Then the EPS file is found before the PDF file.

However, if there is a case where 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.

On the other hand, option `append` prevents the utility of option ‘update’. After the first conversion, the PDF file is found and used first, before even considering the conversion from EPS to PDF.

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>}`

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 `prefersuffix` 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` -> `.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  [2009/07/12 v2.0 Conversion with epstopdf on the fly (H0)]
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}[2009/07/12]
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 Reread check and identification

Reread 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 }%
158 \fi
159 \expandafter\x\csname ver@epstopdf-base.sty\endcsname
160 \ProvidesPackage{epstopdf-base}%
161 [2009/07/12 v2.0 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 miniltx.tex, 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     enable,%
248     append,%
249     update,%
250     prefersuffix,%
251     suffix=-\SourceExt-converted-to%
252 }

```

2.7 Make and verbose

```

253 \begingroup\expandafter\expandafter\expandafter\endgroup
254 \expandafter\ifx\csname pdf@filemoddate\endcsname\relax
255     \def\ETE@Make#1#2{%
256         \ifETE@update
257             \ETE@WarnModDate
258         \fi
259         \@firstofone
260     }%
261 \def\ETE@WarnModDate{%
262     \@PackageWarningNoLine{epstopdf}{%
263         \string\pdf@filemoddate\space is not available,\MessageBreak
264         option ‘update’ will be ignored%
265     }%
266     \global\let\ETE@WarnModDate\relax

```

```

267 }%
268 \def\ETE@FileInfo#1#2{#1 file: <#2>}%
269 \else
270 \def\ETE@Make#1#2{%
271   \ifETE@update
272     \ifnum\pdf@strcmp{\pdf@filemoddate{#1}}{\pdf@filemoddate{#2}}>0 %
273       \expandafter\expandafter\expandafter\@firstofone
274     \else
275       \@PackageInfoNoLine{epstopdf}{%
276         Output file is already uptodate%
277       }%
278     \expandafter\expandafter\expandafter\@gobble
279   \fi
280 \else
281   \expandafter\@firstofone
282 \fi
283 }%
284 \def\ETE@FileInfo#1#2{%
285   #1 file: <#2>%
286   \expandafter\expandafter\expandafter
287   \ETE@Date\pdf@filemoddate{#2}\@nil
288   \expandafter\expandafter\expandafter
289   \ETE@Size\pdf@filesize{#2}\@nil
290 }%
291 \def\ETE@Date#1\@nil{%
292   \ifx\#1\%
293   \else
294     \ETE@@Date#1\@nil
295   \fi
296 }%
297 \def\ETE@@Date#1:#2#3#4#5#6#7#8#9{%
298   \MessageBreak
299   \@spaces\space\space\space date: #2#3#4#5-#6#7-#8#9 %
300   \ETE@@Time
301 }%
302 \def\ETE@@Time#1#2#3#4#5#6#7\@nil{%
303   #1#2:#3#4:#5#6%
304 }%
305 \def\ETE@Size#1\@nil{%
306   \ifx\#1\%
307   \else
308     \MessageBreak
309     \@spaces\space\space\space size: #1 bytes%
310   \fi
311 }%
312 \fi

```

2.8 Adding conversion support

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

```

313 \expandafter\ifx\csname ETE@OrgGin@setfile\endcsname\relax
314   \let\ETE@OrgGin@setfile\Gin@setfile
315 \else
316   \@PackageError{epstopdf}{%
317     Command \string\ETE@OrgGin@setfile\space
318     already defined.\MessageBreak
319   }{%
320     Probably some package has included the code of this package%
321     \MessageBreak
322     instead of using \string\RequirePackage{epstopdf}.%
323     \MessageBreak
324     \@ehc
325   }%

```

```

326 \fi
327 \def\ETE@ifFileExists{%
328   \begingroup\expandafter\expandafter\expandafter\endgroup
329   \expandafter\ifx\csname grffile@ifFileExists\endcsname\relax
330     \expandafter\IfFileExists
331   \else
332     \global\let\ETE@ifFileExists\grffile@ifFileExists
333     \expandafter\grffile@ifFileExists
334   \fi
335 }
336 \def\ETE@Skip#1#2\x\else{%
337   \fi
338   \fi
339   \fi
340 \endgroup
341 #1%
342 \iffalse
343 }
344 \newif\ifETE@InsideSetfile
345 \newcommand*\epstopdfcall}[1]{%
346   \ifETE@InsideSetfile
347     \expandafter\@firstoftwo
348   \else
349     \expandafter\@secondoftwo
350   \fi
351   {'#1}%
352   {\Gin@base\Gin@ext}%
353 }
354 \def\ETE@DefCommandLine#1{%
355   \edef\CommandLine{\expandafter\fi\if'#1}%
356 }
357 \def\ETE@DefX#1{%
358   \expandafter\expandafter\expandafter\def
359   \expandafter\expandafter\expandafter\x
360   \expandafter\expandafter\expandafter{%
361     \expandafter\fi\if'#1\relax\else
362   }%
363 }
364 \def\Gin@setfile#1#2#3{%
365   \ifETE@disable
366     \ETE@OrgGin@setfile{#1}{#2}{#3}%
367   \else
368     \begingroup
369       \ETE@InsideSetfiletrue
370       \ETE@DefX{#3}%
371     \expandafter\endgroup
372     \ifx\x\@empty
373       \ETE@OrgGin@setfile{#1}{#2}{#3}%
374     \else
375       \begingroup
376         \ETE@InsideSetfiletrue
377         \def\GraphicsType{#1}%
378         \def\GraphicsRead{#2}%
379         \ifETE@prefersuffix
380           \else
381             \ifx\ETE@suffix\@empty
382             \else
383               \ETE@ifFileExists{\Gin@base\GraphicsRead}{%
384                 \ETE@Skip{%
385                   \ETE@OrgGin@setfile{#1}{#2}{\Gin@base#2}%
386                 }%
387               }%

```

```

388         \let\next\relax
389     }%
390     \next
391 \fi
392 \fi
393 \ifx\Gin@ext\relax
394     \let\SourceExt\Gin@eext
395     \def\SourceFile{\Gin@base\Gin@eext}%
396 \else
397     \let\SourceExt\Gin@ext
398     \def\SourceFile{\Gin@base\Gin@ext}%
399 \fi
400 \edef\SourceExt{% remove dot
401     \expandafter\@cdr\SourceExt\@empty\@nil
402 }%
403 \let\OutputDirectory\ETE@outdir
404 \ifx\OutputDirectory\@empty
405     \edef\OutputFile{\ETE@GenerateName{\Gin@base}{#2}}%
406 \else
407     \begingroup
408         \filename@parse{\Gin@base#2}%
409         \edef\x{\endgroup
410             \def\noexpand\OutputFile{%
411                 \ETE@GenerateName{%
412                     \OutputDirectory\filename@base
413                 }{#2}%
414             }%
415         }%
416     \x
417 \fi
418 \ETE@DefCommandLine{#3}%
419 \ifETE@verbose
420     \@PackageInfo{epstopdf}{%
421         \ETE@FileInfo{Source}\SourceFile\MessageBreak
422         \ETE@FileInfo{Output}\OutputFile\MessageBreak
423         Command: <\CommandLine>\MessageBreak
424         \string\includegraphics
425     }%
426 \fi
427 \ETE@Make\SourceFile\OutputFile{%
428     \pdf@system{\CommandLine}%
429     \ifETE@verbose
430         \@PackageInfoNoLine{epstopdf}{%
431             \ETE@FileInfo{Result}\OutputFile
432         }%
433     \fi
434 }%
435 \edef\x{\endgroup
436     \ifx\OutputDirectory\@empty
437     \else
438         \def\noexpand\Gin@base{%
439             \OutputDirectory\noexpand\filename@base
440         }%
441     \fi
442     \ifx\ETE@suffix\@empty
443     \else
444         \edef\noexpand\Gin@base{\Gin@base\ETE@suffix}%
445     \fi
446     \noexpand\ETE@OrgGin@setfile{%
447         \GraphicsType
448     }{%
449         \GraphicsRead

```

```

450         }{%
451         \OutputFile
452         }%
453     }%
454     \x
455     \fi
456 \fi
457 }
458 \def\ETE@GenerateName#1#2{%
459     #1\ETE@suffix#2%
460 }

```

2.9 Declare graphics rule

```

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

\DeclareGraphicsRule for .eps
478 \epstopdfDeclareGraphicsRule{.eps}{pdf}{.pdf}{%
479     \ETE@epstopdf{#1}%
480 }
481 \def\ETE@epstopdf#1{%
482     epstopdf %
483     \ifcase\ifx\OutputDirectory\@empty
484         \ifx\ETE@suffix\@empty
485             1%
486         \fi
487         \fi
488         0 %
489     --outfile=\OutputFile\space
490     \fi
491     #1%
492 }
493 \ifETE@prepend
494     \expandafter\PrependGraphicsExtensions
495 \else
496     \expandafter\AppendGraphicsExtensions
497 \fi
498 {.eps}
499 \let\ETE@prepend\@undefined
500 \DeclareVoidOption{prepend}{%
501     \PrependGraphicsExtensions{.eps}%
502 }
503 \let\ETE@append\@undefined
504 \DeclareVoidOption{append}{%
505     \AppendGraphicsExtensions{.eps}%
506 }

```

```

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

509 \ETE@AtEnd
510 </base>

```

3 Test

3.1 Preface for standard catcode check

```

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

```

3.2 Catcode checks for loading

```

517 <*test1>

518 \catcode'\{=1 %
519 \catcode'\}=2 %
520 \catcode'\#=6 %
521 \catcode'\@=11 %
522 \expandafter\ifx\csname count@\endcsname\relax
523   \countdef\count@=255 %
524 \fi
525 \expandafter\ifx\csname @gobble\endcsname\relax
526   \long\def\@gobble#1{}%
527 \fi
528 \expandafter\ifx\csname @firstofone\endcsname\relax
529   \long\def\@firstofone#1{#1}%
530 \fi
531 \expandafter\ifx\csname loop\endcsname\relax
532   \expandafter\@firstofone
533 \else
534   \expandafter\@gobble
535 \fi
536 {%
537   \def\loop#1\repeat{%
538     \def\body{#1}%
539     \iterate
540   }%
541   \def\iterate{%
542     \body
543     \let\next\iterate
544   \else
545     \let\next\relax
546   \fi
547   \next
548 }%
549 \let\repeat=\fi
550 }%
551 \def\RestoreCatcodes{}
552 \count@=0 %
553 \loop
554   \edef\RestoreCatcodes{%
555     \RestoreCatcodes
556     \catcode\the\count@=\the\catcode\count@\relax
557   }%
558 \ifnum\count@<255 %
559   \advance\count@ 1 %
560 \repeat
561

```

```

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

```

4 Installation

4.1 Download

Package. This package is available on CTAN¹:

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

[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
```

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

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 (te \TeX , mi \TeX , ...) relies on file name databases, you must refresh these. For example, te \TeX 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 \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 pdf \LaTeX :

```
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 `pdfdoccmds` 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 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.

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		\AppendGraphicsExtensions . . 496, 505	
\#	520, 582		
\%	585		
\@	521, 578		
\@PackageError	316, 463		
\@PackageInfo	420, 470		
\@PackageInfoNoLine	275, 430		
\@PackageWarningNoLine	203, 214, 228, 262		
\@cdr	401		
\@ehc	324, 465		
\@empty	372, 381, 401, 404, 436, 442, 483, 484		
\@firstofone . .	259, 273, 281, 529, 532		
\@firstoftwo	347		
\@gobble	278, 526, 534		
\@ifundefined	468		
\@namedef	475		
\@nil .	287, 289, 291, 294, 302, 305, 401		
\@secondoftwo	349		
\@spaces	299, 309		
\@undefined	53, 151, 499, 503		
\[583		
\]	292, 306, 462, 579		
\{	518, 580		
\}	519, 581		
\]	584		
		B	
_	586	\body	538, 542
		C	
		\catcode	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, 518, 519, 520, 521, 556, 565, 578, 579, 580, 581, 582, 583, 584, 585, 586, 587
		\CommandLine	355, 423, 428
		\count@	523, 552, 556, 558, 559, 563, 565, 566, 567
		\countdef	523
		\csname	8, 21, 45, 61, 68, 106, 119, 143, 159, 166, 202, 224, 254, 313, 329, 522, 525, 528, 531, 570, 592
		\CurrentOption	94
		D	
		\DeclareBoolOption .	235, 236, 242, 243
		\DeclareComplementaryOption	244
		\DeclareOption	93
		\DeclareStringOption	240, 241
		\DeclareVoidOption .	238, 239, 500, 504
A			
\advance	559, 567		

E		\ifnum 226, 272, 558, 566
\empty 12, 110		\ifx 10,
\end 593		12, 21, 45, 53, 56, 108, 110, 119,
\endcsname 8,		143, 151, 154, 202, 212, 224,
21, 45, 61, 68, 106, 119, 143,		254, 292, 306, 313, 329, 372,
159, 166, 202, 224, 254, 313,		381, 393, 404, 436, 442, 462,
329, 522, 525, 528, 531, 570, 592		483, 484, 522, 525, 528, 531, 570
\endinput 30, 128, 208, 220		\immediate 23, 47, 121, 145
\epstopdfcall 345, 475		\includegraphics 424
\epstopdfDeclareGraphicsRule 461, 478		\input 512, 514, 571
\epstopdfsetup 3, 94, 206, 218, 245, 246		\InputIfFileExists 507, 508
\ETE@@Date 294, 297		\iterate 539, 541, 543
\ETE@@Time 300, 302		
\ETE@append 503	L	
\ETE@AtEnd 81, 82,	\LoadCommand 571, 588	
90, 92, 97, 179, 180, 207, 219, 509	\loop 537, 553, 564	
\ETE@Date 287, 291		
\ETE@DefCommandLine 354, 418	M	
\ETE@DefX 357, 370	\MessageBreak 263, 298,	
\ETE@epstopdf 479, 481	308, 318, 321, 323, 421, 422, 423	
\ETE@FileInfo . 268, 284, 421, 422, 431		
\ETE@GenerateName 405, 411, 458	N	
\ETE@IfFileExists 327, 332, 383	\newcommand . . . 206, 218, 245, 345, 461	
\ETE@InsideSetfiletrue 369, 376	\newif 237, 344	
\ETE@Make 255, 270, 427	\next 388, 390, 543, 545, 547	
\ETE@OrgGin@setfile		
. 314, 317, 366, 373, 385, 446	O	
\ETE@outdir 403	\OutputDirectory	
\ETE@prepend 499 403, 404, 412, 436, 439, 483	
\ETE@prependfalse 239	\OutputFile	
\ETE@prependtrue 238	. 405, 410, 422, 427, 431, 451, 489	
\ETE@SavedAtEnd 90, 92		
\ETE@Size 289, 305	P	
\ETE@Skip 336, 384	\PackageInfo 26, 124	
\ETE@suffix . . . 381, 442, 444, 459, 484	\pdf@filemoddate 272, 287	
\ETE@WarnModDate 257, 261, 266	\pdf@filesize 289	
	\pdf@shellescape 226	
	\pdf@strcmp 272	
F		\pdf@system 428
\filename@base 412, 439	\pdf@filemoddate 263	
\filename@parse 408	\PrependGraphicsExtensions . 494, 501	
	\ProcessOptions 96	
G		\ProvidesPackage 62, 160
\Gin@base 352, 383,		
385, 395, 398, 405, 408, 438, 444	R	
\Gin@driver 212, 513	\RangeCatcodeInvalid	
\Gin@eext 394, 395 562, 574, 575, 576, 577	
\Gin@ext 352, 393, 397, 398	\repeat 537, 549, 560, 568	
\Gin@setfile 314, 364	\RequirePackage	
\GraphicsRead 378, 383, 449 91, 197, 198, 199, 200, 322	
\GraphicsType 377, 447	\resetatcatcode 515	
\grffile@IfFileExists 332, 333	\RestoreCatcodes . . 551, 554, 555, 589	
I		S
\if 355, 361	\setkeys 245	
\ifcase 9, 107, 483	\SetupKeyvalOptions 234	
\ifETE@disable 365	\SourceExt . . . 251, 394, 397, 400, 401	
\ifETE@InsideSetfile 344, 346	\SourceFile 395, 398, 421, 427	
\ifETE@prefersuffix 379	\space 263, 299, 309, 317, 489	
\ifETE@prepend 237, 493		
\ifETE@update 256, 271	T	
\ifETE@verbose 419, 429	\Test 573, 591	
\iffalse 342	\the 69, 70, 71,	
\IfFileExists 330	72, 83, 167, 168, 169, 170, 181, 556	

		X	
\TMP@EnsureCode	80, 87, 88,	\x	8, 10, 12, 22,
	89, 178, 185, 186, 187, 188, 189,		26, 28, 46, 51, 61, 67, 75, 106,
	190, 191, 192, 193, 194, 195, 196		108, 110, 120, 124, 126, 144,
		W	
\write	23, 47, 121, 145		149, 159, 165, 173, 211, 212,
			336, 359, 372, 409, 416, 435, 454