

The **epstopdf** package

Heiko Oberdiek
<oberdiek@uni-freiburg.de>

2009/03/01 v1.8

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	2
1.4	Options	3
1.5	Configuration	4
1.5.1	Configuration file	4
1.5.2	Conversion program	4
1.6	Other image formats	4
2	Implementation	4
2.1	Preparations	4
2.1.1	Relead check and identification	4
2.1.2	Catcodes	5
2.1.3	Load packages	6
2.2	Checks	6
2.3	Package loading	7
2.4	Options	7
2.5	Make and verbose	7
2.6	Adding conversion support	8
3	Test	10
3.1	Preface for standard catcode check	10
3.2	Catcode checks for loading	10
4	Installation	11
4.1	Download	11
4.2	Bundle installation	12
4.3	Package installation	12
4.4	Refresh file name databases	12
4.5	Some details for the interested	12
5	History	13
	[2001/01/06 v1.0]	13
	[2001/02/04 v1.1]	13
	[2006/02/20 v1.2]	13
	[2006/08/26 v1.3]	13
	[2007/04/26 v1.4]	13

[2007/10/02 v1.5]	14
[2007/11/11 v1.6]	14
[2008/05/06 v1.7]	14
[2009/03/01 v1.8]	14

6 Index	14
----------------	-----------

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 be 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:

```
\DeclareGraphicsRule{.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 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. Then option `update` have the control and can compare file dates.

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`.

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

1.5 Configuration

1.5.1 Configuration file

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. My favourite setting is:

```
\epstopdfsetup{update,prepend,verbose}
```

1.5.2 Conversion program

You can use `\DeclareGraphicsRule` the same way as the route via `dvips` to specify the conversion command line, examples below.

Additionally you can use the following macros:

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

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

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}{%  
  'ps2pdf -dEPSCrop #1 \noexpand\OutputFile  
}
```

`\DeclareGraphicsRule` expands the argument, therefore `\noexpand` is necessary. 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`:

```
\DeclareGraphicsRule{.gif}{png}{.png}{%  
  'convert #1 \noexpand\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 Preparations

2.1.1 Relead check and identification

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 \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 [2009/03/01 v1.8 Conversion with epstopdf on the fly (H0)]

```

2.1.2 Catcodes

```

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{33}{12}% !
88 \TMP@EnsureCode{39}{12}% '
89 \TMP@EnsureCode{42}{12}% *
90 \TMP@EnsureCode{44}{12}% ,
91 \TMP@EnsureCode{45}{12}% -
92 \TMP@EnsureCode{46}{12}% .
93 \TMP@EnsureCode{47}{12}% /
94 \TMP@EnsureCode{58}{12}% :
95 \TMP@EnsureCode{60}{12}% <
96 \TMP@EnsureCode{61}{12}% =
97 \TMP@EnsureCode{62}{12}% >
98 \TMP@EnsureCode{96}{12}% ‘

```

2.1.3 Load packages

```

99 \RequirePackage{infwarerr}[2007/09/09]
100 \RequirePackage{grfext}\relax
101 \RequirePackage{kvoptions}[2007/10/02]
102 \RequirePackage{pdftexcmds}[2007/11/11]

```

2.2 Checks

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

```

103 \begingroup\expandafter\expandafter\expandafter\endgroup
104 \expandafter\ifx\csname Gin@setfile\endcsname\relax
105   \@PackageWarningNoLine{epstopdf}{%
106     No graphics package \string'graphic{s,x}\string' loaded%
107   }%
108   \newcommand*{\epstopdfsetup}[1]{}%
109   \ETE@AtEnd
110 \expandafter\endinput
111 \fi

```

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

```

112 \begingroup
113   \def\x{pdftex.def}%
114   \ifx\Gin@driver\x
115   \else
116     \@PackageWarningNoLine{epstopdf}{%
117       Other drivers than 'pdftex.def' are not supported%
118     }%

```

```

119 \endgroup
120 \newcommand*{\epstopdfsetup}[1]{}%
121 \ETE@AtEnd
122 \expandafter\endinput
123 \fi
124 \endgroup

    Check, whether the shell escape feature is enabled.

125 \begingroup
126 \expandafter\ifx\csname pdf@shellescape\endcsname\relax
127 \else
128 \ifnum\pdf@shellescape>0 %
129 \else
130 \@PackageWarningNoLine{epstopdf}{%
131     Shell escape feature is not enabled%
132 }%
133 \fi
134 \fi
135 \endgroup

```

2.3 Package loading

2.4 Options

```

136 \SetupKeyvalOptions{family=ETE,prefix=ETE@}
137 \DeclareBoolOption{update}
138 \DeclareBoolOption{verbose}
139 \newif\ifETE@prepend
140 \DeclareVoidOption{prepend}{\ETE@prependtrue}
141 \DeclareVoidOption{append}{\ETE@prependfalse}
142 \DeclareStringOption{outdir}
143 \ProcessKeyvalOptions*
144 \newcommand*{\epstopdfsetup}{\setkeys{ETE}}

```

2.5 Make and verbose

```

145 \begingroup\expandafter\expandafter\expandafter\endgroup
146 \expandafter\ifx\csname pdf@filemoddate\endcsname\relax
147 \def\ETE@Make#1#2{%
148     \ifETE@update
149         \ETE@WarnModDate
150     \fi
151     \@firstofone
152 }%
153 \def\ETE@WarnModDate{%
154     \@PackageWarningNoLine{epstopdf}{%
155         \string\pdf@filemoddate\space is not available,\MessageBreak
156         option ‘update’ will be ignored%
157     }%
158     \global\let\ETE@WarnModDate\relax
159 }%
160 \def\ETE@FileInfo#1#2{#1 file: <#2>}%
161 \else
162 \def\ETE@Make#1#2{%
163     \ifETE@update
164         \ifnum\pdf@strcmp{\pdf@filemoddate{#1}}{\pdf@filemoddate{#2}}>0 %
165             \expandafter\expandafter\expandafter\@firstofone
166         \else
167             \@PackageInfoNoLine{epstopdf}{%
168                 Output file is already uptodate%
169             }%
170             \expandafter\expandafter\expandafter\@gobble
171         \fi

```

```

172 \else
173 \expandafter\@firstofone
174 \fi
175 }%
176 \def\ETE@FileInfo#1#2{%
177 #1 file: <#2>%
178 \expandafter\expandafter\expandafter
179 \ETE@Date\pdf@filemoddate{#2}\@nil
180 \expandafter\expandafter\expandafter
181 \ETE@Size\pdf@filesize{#2}\@nil
182 }%
183 \def\ETE@Date#1\@nil{%
184 \ifx\#1\%
185 \else
186 \ETE@@Date#1\@nil
187 \fi
188 }%
189 \def\ETE@@Date#1:#2#3#4#5#6#7#8#9{%
190 \MessageBreak
191 \@spaces\space\space\space date: #2#3#4#5-#6#7-#8#9 %
192 \ETE@Time
193 }%
194 \def\ETE@Time#1#2#3#4#5#6#7\@nil{%
195 #1#2:#3#4:#5#6%
196 }%
197 \def\ETE@Size#1\@nil{%
198 \ifx\#1\%
199 \else
200 \MessageBreak
201 \@spaces\space\space\space size: #1 bytes%
202 \fi
203 }%
204 \fi

```

2.6 Adding conversion support

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

```

205 \expandafter\ifx\csname ETE@OrgGin@setfile\endcsname\relax
206 \let\ETE@OrgGin@setfile\Gin@setfile
207 \else
208 \@PackageError{epstopdf}{%
209 Command \string\ETE@OrgGin@setfile\space
210 already defined.\MessageBreak
211 }{%
212 Probably some package has included the code of this package%
213 \MessageBreak
214 instead of using \string\RequirePackage{epstopdf}.%
215 \MessageBreak
216 \@ehc
217 }%
218 \fi
219 \def\Gin@setfile#1#2#3{%
220 \if'\@car #3\relax\@nil
221 \beginingroup
222 \def\GraphicsType{#1}%
223 \def\GraphicsRead{#2}%
224 \ifx\Gin@ext\relax
225 \def\SourceFile{\Gin@base\Gin@eext}%
226 \else
227 \def\SourceFile{\Gin@base\Gin@ext}%
228 \fi
229 \let\OutputDirectory\ETE@outdir
230 \ifx\OutputDirectory\@empty

```



```

231     \def\OutputFile{\Gin@base#2}%
232 \else
233     \begingroup
234     \filename@parse{\Gin@base#2}%
235     \edef\x{\endgroup
236     \def\noexpand\OutputFile{%
237     \OutputDirectory\filename@base#2%
238     }%
239     }%
240     \x
241 \fi
242 \edef\CommandLine{\@cdr#3\@empty\@nil}%
243 \ifETE@verbose
244     \@PackageInfo{epstopdf}{%
245     \ETE@FileInfo{Source}\SourceFile\MessageBreak
246     \ETE@FileInfo{Output}\OutputFile\MessageBreak
247     Command: <\CommandLine>\MessageBreak
248     \string\includegraphics
249     }%
250 \fi
251 \ETE@Make\SourceFile\OutputFile{%
252     \pdf@system{\CommandLine}%
253     \ifETE@verbose
254     \@PackageInfoNoLine{epstopdf}{%
255     \ETE@FileInfo{Result}\OutputFile
256     }%
257 \fi
258 }%
259 \edef\x{\endgroup
260     \ifx\OutputDirectory\@empty
261     \else
262     \def\noexpand\Gin@base{%
263     \OutputDirectory\noexpand\filename@base
264     }%
265 \fi
266     \noexpand\ETE@OrgGin@setfile{%
267     \GraphicsType
268     }{%
269     \GraphicsRead
270     }{%
271     \OutputFile
272     }%
273     }%
274     \x
275 \else
276     \ETE@OrgGin@setfile{#1}{#2}{#3}%
277 \fi
278 }

\DeclareGraphicsRule for .eps
279 \expandafter\ifx\csname Gin@rule@.eps\endcsname\relax
280 \else
281     \@PackageInfo{epstopdf}{Overwriting graphics rule for '.eps'%}
282 \fi
283 \@namedef{Gin@rule@.eps}#1{{pdf}}{.pdf}{'\ETE@epstopdf{#1}}
284 \def\ETE@epstopdf#1{%
285     epstopdf %
286     \ifx\OutputDirectory\@empty
287     \else
288         --outfile=\OutputFile\space
289     \fi
290     #1%
291 }

```

```

292 \ifETE@prepend
293   \expandafter\PrependGraphicsExtensions
294 \else
295   \expandafter\AppendGraphicsExtensions
296 \fi
297 {.eps}
298 \let\ETE@prepend\@undefined
299 \DeclareVoidOption{prepend}{%
300   \PrependGraphicsExtensions{.eps}%
301 }
302 \let\ETE@append\@undefined
303 \DeclareVoidOption{append}{%
304   \AppendGraphicsExtensions{.eps}%
305 }
306 \InputIfFileExists{epstopdf.cfg}{-}{-}
307 \ETE@AtEnd
308 \</package>

```

3 Test

3.1 Preface for standard catcode check

```

309 <*test1>
310 \input miniltx.tex\relax
311 \def\Gin@driver{pdftex.def}
312 \input graphicx.sty\relax
313 \resetatcatcode
314 </test1>

```

3.2 Catcode checks for loading

```

315 <*test1>
316 \catcode'\{=1 %
317 \catcode'\}=2 %
318 \catcode'\#=6 %
319 \catcode'\@=11 %
320 \expandafter\ifx\csname count@\endcsname\relax
321   \countdef\count@=255 %
322 \fi
323 \expandafter\ifx\csname @gobble\endcsname\relax
324   \long\def\@gobble#1{}%
325 \fi
326 \expandafter\ifx\csname @firstofone\endcsname\relax
327   \long\def\@firstofone#1{#1}%
328 \fi
329 \expandafter\ifx\csname loop\endcsname\relax
330   \expandafter\@firstofone
331 \else
332   \expandafter\@gobble
333 \fi
334 {%
335   \def\loop#1\repeat{%
336     \def\body{#1}%
337     \iterate
338   }%
339   \def\iterate{%
340     \body
341     \let\next\iterate
342   \else
343     \let\next\relax
344   \fi

```

```

345     \next
346   }%
347   \let\repeat=\fi
348 }%
349 \def\RestoreCatcodes{}
350 \count@=0 %
351 \loop
352   \edef\RestoreCatcodes{%
353     \RestoreCatcodes
354     \catcode\the\count@=\the\catcode\count@\relax
355   }%
356 \ifnum\count@<255 %
357   \advance\count@ 1 %
358 \repeat
359
360 \def\RangeCatcodeInvalid#1#2{%
361   \count@=#1\relax
362   \loop
363     \catcode\count@=15 %
364   \ifnum\count@<#2\relax
365     \advance\count@ 1 %
366   \repeat
367 }
368 \expandafter\ifx\csname LoadCommand\endcsname\relax
369   \def\LoadCommand{\input epstopdf.sty\relax}%
370 \fi
371 \def\Test{%
372   \RangeCatcodeInvalid{0}{47}%
373   \RangeCatcodeInvalid{58}{64}%
374   \RangeCatcodeInvalid{91}{96}%
375   \RangeCatcodeInvalid{123}{255}%
376   \catcode'\@=12 %
377   \catcode'\=0 %
378   \catcode'\{=1 %
379   \catcode'\}=2 %
380   \catcode'\#=6 %
381   \catcode'\[=12 %
382   \catcode'\]=12 %
383   \catcode'\%=14 %
384   \catcode'\ =10 %
385   \catcode13=5 %
386   \LoadCommand
387   \RestoreCatcodes
388 }
389 \Test
390 \csname @@end\endcsname
391 \end
392 </test1>

```

4 Installation

4.1 Download

Package. This package is available on CTAN¹:

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

[CTAN:macros/latex/contrib/oberdiek/epstopdf.pdf](http://ctan.org/ctan/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.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, miK_TE_X, ...) 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 `pdftexcmds` 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.

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	A
<code>\#</code> 318, 380	<code>\advance</code> 357, 365
<code>\%</code> 383	<code>\AppendGraphicsExtensions</code> .. 295, 304
<code>\@</code> 319, 376	
<code>\@PackageError</code> 208	B
<code>\@PackageInfo</code> 244, 281	<code>\body</code> 336, 340
<code>\@PackageInfoNoLine</code> 167, 254	
<code>\@PackageWarningNoLine</code>	C
..... 105, 116, 130, 154	<code>\catcode</code> ... 3, 4, 5, 6, 7, 18, 19, 20,
<code>\@car</code> 220	34, 35, 36, 37, 38, 39, 40, 41, 42,
<code>\@cdr</code> 242	43, 44, 65, 66, 69, 70, 71, 72, 76,
<code>\@ehc</code> 216	77, 78, 79, 83, 85, 316, 317, 318,
<code>\@empty</code> 230, 242, 260, 286	319, 354, 363, 376, 377, 378,
<code>\@firstofone</code> .. 151, 165, 173, 327, 330	379, 380, 381, 382, 383, 384, 385
<code>\@gobble</code> 170, 324, 332	<code>\CommandLine</code> 242, 247, 252
<code>\@namedef</code> 283	<code>\count@</code> 321, 350,
<code>\@nil</code> 179,	354, 356, 357, 361, 363, 364, 365
181, 183, 186, 194, 197, 220, 242	<code>\countdef</code> 321
<code>\@spaces</code> 191, 201	<code>\csname</code> 8, 21,
<code>\@undefined</code> 53, 298, 302	45, 61, 68, 104, 126, 146, 205,
<code>\[</code> 381	279, 320, 323, 326, 329, 368, 390
<code>\]</code> 184, 198, 377	
<code>\{</code> 316, 378	D
<code>\}</code> 317, 379	<code>\DeclareBoolOption</code> 137, 138
<code>\]</code> 382	<code>\DeclareStringOption</code> 142
	<code>\DeclareVoidOption</code> . 140, 141, 299, 303
	E
<code>_</code> 384	<code>\empty</code> 12

\end	391	\loop	335, 351, 362
\endcsname	8, 21, 45, 61, 68, 104, 126, 146, 205, 279, 320, 323, 326, 329, 368, 390	M	
\endinginput	30, 110, 122	\MessageBreak	155, 190, 200, 210, 213, 215, 245, 246, 247
\epstopdfsetup	3, 108, 120, 144	N	
\ETE@Date	186, 189	\newcommand	108, 120, 144
\ETE@Time	192, 194	\newif	139
\ETE@append	302	\next	341, 343, 345
\ETE@AtEnd	81, 82, 109, 121, 307	O	
\ETE@Date	179, 183	\OutputDirectory	229, 230, 237, 260, 263, 286
\ETE@epstopdf	283, 284	\OutputFile	231, 236, 246, 251, 255, 271, 288
\ETE@FileInfo	160, 176, 245, 246, 255	P	
\ETE@Make	147, 162, 251	\PackageInfo	26
\ETE@OrgGin@setfile	206, 209, 266, 276	\pdf@filemoddate	164, 179
\ETE@outdir	229	\pdf@filesize	181
\ETE@prepend	298	\pdf@shellescape	128
\ETE@prependfalse	141	\pdf@strcmp	164
\ETE@prependtrue	140	\pdf@system	252
\ETE@Size	181, 197	\pdf@filemoddate	155
\ETE@WarnModDate	149, 153, 158	\PrependGraphicsExtensions	293, 300
F		\ProcessKeyvalOptions	143
\filename@base	237, 263	\ProvidesPackage	62
\filename@parse	234	R	
G		\RangeCatcodeInvalid	360, 372, 373, 374, 375
\Gin@base	225, 227, 231, 234, 262	\repeat	335, 347, 358, 366
\Gin@driver	114, 311	\RequirePackage	99, 100, 101, 102, 214
\Gin@eext	225	\resetatcatcode	313
\Gin@ext	224, 227	\RestoreCatcodes	349, 352, 353, 387
\Gin@setfile	206, 219	S	
\GraphicsRead	223, 269	\setkeys	144
\GraphicsType	222, 267	\SetupKeyvalOptions	136
I		\SourceFile	225, 227, 245, 251
\if	220	\space	155, 191, 201, 209, 288
\ifcase	9	T	
\ifETE@prepend	139, 292	\Test	371, 389
\ifETE@update	148, 163	\the	69, 70, 71, 72, 83, 354
\ifETE@verbose	243, 253	\TMP@EnsureCode	80, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98
\ifnum	128, 164, 356, 364	W	
\ifx	10, 12, 21, 45, 53, 56, 104, 114, 126, 146, 184, 198, 205, 224, 230, 260, 279, 286, 320, 323, 326, 329, 368	\write	23, 47
\immediate	23, 47	X	
\includegraphics	248	\x	8, 10, 12, 22, 26, 28, 46, 51, 61, 67, 75, 113, 114, 235, 240, 259, 274
\input	310, 312, 369		
\InputIfFileExists	306		
\iterate	337, 339, 341		
L			
\LoadCommand	369, 386		