

The **epstopdf** package

Heiko Oberdiek
<oberdiek@uni-freiburg.de>

2007/10/02 v1.5

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	3
1.5.1	Configuration file	3
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	6
2.4	Options	6
2.5	Make and verbose	7
2.6	Adding conversion support	8
3	Test	9
3.1	Preface for standard catcode check	9
3.2	Catcode checks for loading	9
4	Installation	10
4.1	Download	10
4.2	Bundle installation	11
4.3	Package installation	11
4.4	Refresh file name databases	11
4.5	Some details for the interested	11
5	History	12
	[2001/01/06 v1.0]	12
	[2001/02/04 v1.1]	12
	[2006/02/20 v1.2]	12
	[2006/08/26 v1.3]	12
	[2007/04/26 v1.4]	12
	[2007/10/02 v1.5]	12

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 `\TeX`. Mik_TE_X, te_TE_X, T_EX Live support it. In Web2C based T_EX distributions (te_TE_X, 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 L^AT_EX run. If you want to use option `update` and your pdfT_EX 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. This option

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   \expandafter\ifx\csname PackageInfo\endcsname\relax
19     \def\x#1#2{%
20       \immediate\write-1{Package #1 Info: #2.}%
21     }%
22   \else
23     \def\x#1#2{\PackageInfo{#1}{#2, stopped}}%
24   \fi
25   \x{epstopdf}{The package is already loaded}%
26   \endgroup
27   \expandafter\endinput
28 \fi
29 \endgroup

```

Package identification:

```

30 \begingroup
31   \catcode40 12 % (
32   \catcode41 12 % )
33   \catcode44 12 % ,
34   \catcode45 12 % -
35   \catcode46 12 % .
36   \catcode47 12 % /
37   \catcode58 12 % :
38   \catcode64 11 % @
39   \expandafter\ifx\csname ProvidesPackage\endcsname\relax
40     \def\x#1#2#3[#4]{\endgroup
41       \immediate\write-1{Package: #3 #4}%
42       \xdef#1{#4}%
43     }%
44   \else
45     \def\x#1#2[#3]{\endgroup
46       #2[{#3}]%
47       \ifx#1\relax
48         \xdef#1{#3}%
49       \fi
50     }%
51   \fi
52   \expandafter\x\csname ver@epstopdf.sty\endcsname
53   \ProvidesPackage{epstopdf}%
54   [2007/10/02 v1.5 Conversion with epstopdf on the fly (H0)]

```

2.1.2 Catcodes

```

55 \expandafter\edef\csname ETE@AtEnd\endcsname{%
56   \catcode64 \the\catcode64\relax
57 }
58 \catcode64 11 % @
59 \def\TMP@EnsureCode#1#2{%
60   \edef\ETE@AtEnd{%
61     \ETE@AtEnd
62     \catcode#1 \the\catcode#1\relax
63   }%
64   \catcode#1 #2\relax
65 }
66 \TMP@EnsureCode{33}{12}% !
67 \TMP@EnsureCode{39}{12}% '
68 \TMP@EnsureCode{42}{12}% *
69 \TMP@EnsureCode{44}{12}% ,
70 \TMP@EnsureCode{45}{12}% -
71 \TMP@EnsureCode{46}{12}% .
72 \TMP@EnsureCode{47}{12}% /
73 \TMP@EnsureCode{58}{12}% :
74 \TMP@EnsureCode{60}{12}% <

```

```

75 \TMP@EnsureCode{61}{12}% =
76 \TMP@EnsureCode{62}{12}% >
77 \TMP@EnsureCode{96}{12}% ‘

```

2.1.3 Load packages

```

78 \RequirePackage{infwarerr}[2007/09/09]
79 \RequirePackage{grfext}\relax
80 \RequirePackage{kvoptions}[2006/10/02]

```

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.

```

81 \begingroup\expandafter\expandafter\expandafter\endgroup
82 \expandafter\ifx\csname Gin@setfile\endcsname\relax
83   \@PackageWarningNoLine{epstopdf}{%
84     No graphics package \string'graphic{s,x}\string' found%
85   }%
86   \newcommand*{\epstopdfsetup}[1]{}%
87   \ETE@AtEnd
88   \expandafter\endinput
89 \fi

```

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

```

90 \begingroup
91   \def\x{pdftex.def}%
92   \ifx\Gin@driver\x
93   \else
94     \@PackageWarningNoLine{epstopdf}{%
95       Graphics driver file \string'pdftex.def\string' not found%
96     }%
97   \endgroup
98   \newcommand*{\epstopdfsetup}[1]{}%
99   \ETE@AtEnd
100   \expandafter\endinput
101 \fi
102 \endgroup

```

Check, whether the shell escape feature is enabled.

```

103 \begingroup
104   \expandafter\ifx\csname pdfshellescape\endcsname\relax
105   \else
106     \ifnum\pdfshellescape>0 %
107     \else
108       \@PackageWarningNoLine{epstopdf}{%
109         Shell escape feature is not enabled%
110       }%
111     \fi
112   \fi
113 \endgroup

```

2.3 Package loading

2.4 Options

```

114 \SetupKeyvalOptions{family=ETE,prefix=ETE@}
115 \DeclareBoolOption{update}
116 \DeclareBoolOption{verbose}
117 \newif\ifETE@prepend
118 \DeclareVoidOption{prepend}{\ETE@prependtrue}
119 \DeclareVoidOption{append}{\ETE@prependfalse}
120 \DeclareStringOption{outdir}

```

```

121 \ProcessKeyvalOptions*
122 \newcommand*{\epstopdfsetup}{\setkeys{ETE}}

```

2.5 Make and verbose

```

123 \begingroup\expandafter\expandafter\expandafter\endgroup
124 \expandafter\ifx\csname pdffilemoddate\endcsname\relax
125   \def\ETE@Make#1#2{%
126     \ifETE@update
127       \ETE@WarnModDate
128     \fi
129     \@firstofone
130   }%
131   \def\ETE@WarnModDate{%
132     \@PackageWarningNoLine{epstopdf}{%
133       \string\pdffilemoddate\space is not available,\MessageBreak
134       option 'update' will be ignored%
135     }%
136     \global\let\ETE@WarnModDate\relax
137   }%
138   \def\ETE@FileInfo#1#2{#1 file: <#2>}%
139 \else
140   \def\ETE@Make#1#2{%
141     \ifETE@update
142       \ifnum\pdfstrcmp{\pdffilemoddate{#1}}{\pdffilemoddate{#2}}>0 %
143         \expandafter\expandafter\expandafter\@firstofone
144       \else
145         \@PackageInfoNoLine{epstopdf}{%
146           Output file is already uptodate%
147         }%
148         \expandafter\expandafter\expandafter\@gobble
149       \fi
150     \else
151       \expandafter\@firstofone
152     \fi
153   }%
154   \def\ETE@FileInfo#1#2{%
155     #1 file: <#2>%
156     \expandafter\ETE@Date\pdffilemoddate{#2}\@nil
157     \expandafter\ETE@Size\pdffilesize{#2}\@nil
158   }%
159   \def\ETE@Date#1\@nil{%
160     \ifx\#1\%
161     \else
162       \ETE@@Date#1\@nil
163     \fi
164   }%
165   \def\ETE@@Date#1:#2#3#4#5#6#7#8#9{%
166     \MessageBreak
167     \@spaces\space\space\space date: #2#3#4#5-#6#7-#8#9 %
168     \ETE@@Time
169   }%
170   \def\ETE@@Time#1#2#3#4#5#6#7\@nil{%
171     #1#2:#3#4:#5#6%
172   }%
173   \def\ETE@Size#1\@nil{%
174     \ifx\#1\%
175     \else
176       \MessageBreak
177       \@spaces\space\space size: #1 bytes%
178     \fi
179   }%
180 \fi

```

2.6 Adding conversion support

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

```

181 \expandafter\ifx\csname ETE@OrgGin@setfile\endcsname\relax
182   \let\ETE@OrgGin@setfile\Gin@setfile
183 \else
184   \@PackageError{epstopdf}{%
185     Command \string\ETE@OrgGin@setfile\space
186     already defined.\MessageBreak
187   }{%
188     Probably some package has included the code of this package%
189     \MessageBreak
190     instead of using \string\RequirePackage{epstopdf}.%
191     \MessageBreak
192     \@ehc
193   }%
194 \fi
195 \def\Gin@setfile#1#2#3{%
196   \if'\@car #3\relax\@nil
197     \begingroup
198       \def\GraphicsType{#1}%
199       \def\GraphicsRead{#2}%
200       \ifx\Gin@ext\relax
201         \def\SourceFile{\Gin@base\Gin@eext}%
202       \else
203         \def\SourceFile{\Gin@base\Gin@ext}%
204       \fi
205       \let\OutputDirectory\ETE@outdir
206       \ifx\OutputDirectory\@empty
207         \def\OutputFile{\Gin@base#2}%
208       \else
209         \begingroup
210           \filename@parse{\Gin@base#2}%
211           \edef\x{\endgroup
212             \def\noexpand\OutputFile{%
213               \OutputDirectory\filename@base#2%
214             }%
215           }%
216         \x
217       \fi
218       \edef\CommandLine{\@cdr#3\@empty\@nil}%
219       \ifETE@verbose
220         \@PackageInfo{epstopdf}{%
221           \ETE@FileInfo{Source}\SourceFile\MessageBreak
222           \ETE@FileInfo{Output}\OutputFile\MessageBreak
223           Command: <\CommandLine>\MessageBreak
224           \string\includegraphics
225         }%
226       \fi
227       \ETE@Make\SourceFile\OutputFile{%
228         \immediate\write18{\CommandLine}%
229       \ifETE@verbose
230         \@PackageInfoNoLine{epstopdf}{%
231           \ETE@FileInfo{Result}\OutputFile
232         }%
233       \fi
234     }%
235     \edef\x{\endgroup
236       \ifx\OutputDirectory\@empty
237       \else
238         \def\noexpand\Gin@base{%
239           \OutputDirectory\noexpand\filename@base
240         }%

```



```

241         \fi
242         \noexpand\ETE@OrgGin@setfile{%
243             \GraphicsType
244         }{%
245             \GraphicsRead
246         }{%
247             \OutputFile
248         }%
249     }%
250     \x
251 \else
252     \ETE@OrgGin@setfile{#1}{#2}{#3}%
253 \fi
254 }

\DeclareGraphicsRule for .eps
255 \expandafter\ifx\csname Gin@rule@.eps\endcsname\relax
256 \else
257     \@PackageInfo{epstopdf}{Overwriting graphics rule for '.eps'}%
258 \fi
259 \@namedef{Gin@rule@.eps}#1{{pdf}}{.pdf}{'\ETE@epstopdf{#1}}
260 \def\ETE@epstopdf#1{%
261     epstopdf %
262     \ifx\OutputDirectory\@empty
263     \else
264         --outfile=\OutputFile\space
265     \fi
266     #1%
267 }

268 \ifETE@prepend
269     \expandafter\PrependGraphicsExtensions
270 \else
271     \expandafter\AppendGraphicsExtensions
272 \fi
273 {.eps}
274 \let\ETE@prepend\@undefined
275 \DeclareVoidOption{prepend}{%
276     \PrependGraphicsExtensions{.eps}%
277 }
278 \let\ETE@append\@undefined
279 \DeclareVoidOption{append}{%
280     \AppendGraphicsExtensions{.eps}%
281 }

282 \InputIfFileExists{epstopdf.cfg}{-}{-}

283 \ETE@AtEnd
284 \end{package}

```

3 Test

3.1 Preface for standard catcode check

```

285 \test1
286 \input miniltx.tex\relax
287 \def\Gin@driver{pdftex.def}
288 \input graphicx.sty\relax
289 \resetatcatcode
290 \test1

```

3.2 Catcode checks for loading

```

291 \test1

```

```

292 \catcode'\@=11 %
293 \def\RestoreCatcodes{}
294 \count@=0 %
295 \loop
296   \edef\RestoreCatcodes{%
297     \RestoreCatcodes
298     \catcode\the\count@=\the\catcode\count@\relax
299   }%
300 \ifnum\count@<255 %
301   \advance\count@\@ne
302 \repeat
303
304 \def\RangeCatcodeInvalid#1#2{%
305   \count@=#1\relax
306   \loop
307     \catcode\count@=15 %
308     \ifnum\count@<#2\relax
309       \advance\count@\@ne
310     \repeat
311 }
312 \def\Test{%
313   \RangeCatcodeInvalid{0}{47}%
314   \RangeCatcodeInvalid{58}{64}%
315   \RangeCatcodeInvalid{91}{96}%
316   \RangeCatcodeInvalid{123}{255}%
317   \catcode'\@=12 %
318   \catcode'\=0 %
319   \catcode'\{=1 %
320   \catcode'\}=2 %
321   \catcode'\#=6 %
322   \catcode'\[=12 %
323   \catcode'\]=12 %
324   \catcode'\%=14 %
325   \catcode'\ =10 %
326   \catcode13=5 %
327   \input epstopdf.sty\relax
328   \RestoreCatcodes
329 }
330 \Test
331 \csname @@end\endcsname
332 \end
333 </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:macros/latex/contrib/oberdiek/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.

¹<http://ftp.ctan.org/tex-archive/>

4.2 Bundle installation

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

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

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

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

4.3 Package installation

Unpacking. The `.dtx` file is a self-extracting `docstrip` archive. The files are extracted by running the `.dtx` through plain- \TeX :

```
tex epstopdf.dtx
```

TDS. Now the different files must be moved into the different directories in your installation TDS tree (also known as `texmf` tree):

<code>epstopdf.sty</code>	\rightarrow <code>tex/latex/oberdiek/epstopdf.sty</code>
<code>epstopdf.pdf</code>	\rightarrow <code>doc/latex/oberdiek/epstopdf.pdf</code>
<code>test/epstopdf-test1.tex</code>	\rightarrow <code>doc/latex/oberdiek/test/epstopdf-test1.tex</code>
<code>epstopdf.dtx</code>	\rightarrow <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 \TeX distribution (te \TeX , mik \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 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 `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.

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	
\#	321
\%	324
\@	292, 317
\@PackageError	184
\@PackageInfo	220, 257
\@PackageInfoNoLine	145, 230
\@PackageWarningNoLine	83, 94, 108, 132
\@car	196
\@cdr	218
\@ehc	192
\@empty	206, 218, 236, 262
\@firstofone	129, 143, 151
\@gobble	148
\@namedef	259
\@ne	301, 309
\@nil	156,
	157, 159, 162, 170, 173, 196, 218
\@spaces	167, 177
\@undefined	274, 278
\[322
\\	160, 174, 318
\{	319
\}	320
\]	323
_	325
A	
\advance	301, 309
\AppendGraphicsExtensions	271, 280
C	
\catcode	3, 4, 5, 6, 7, 31, 32,
	33, 34, 35, 36, 37, 38, 56, 58, 62,
	64, 292, 298, 307, 317, 318, 319,
	320, 321, 322, 323, 324, 325, 326
\CommandLine	218, 223, 228
\count@	294,
	298, 300, 301, 305, 307, 308, 309
\csname	8, 18, 39,
	52, 55, 82, 104, 124, 181, 255, 331
D	
\DeclareBoolOption	115, 116
\DeclareStringOption	120
\DeclareVoidOption	118, 119, 275, 279
E	
\empty	12
\end	332
\endcsname	8, 18, 39,
	52, 55, 82, 104, 124, 181, 255, 331
\endinput	27, 88, 100
\epstopdfsetup	3, 86, 98, 122
\ETE@@Date	162, 165
\ETE@@Time	168, 170
\ETE@append	278
\ETE@AtEnd	60, 61, 87, 99, 283
\ETE@Date	156, 159
\ETE@epstopdf	259, 260
\ETE@FileInfo	138, 154, 221, 222, 231
\ETE@Make	125, 140, 227
\ETE@OrgGin@setfile	182, 185, 242, 252
\ETE@outdir	205
\ETE@prepend	274
\ETE@prependfalse	119
\ETE@prependtrue	118
\ETE@Size	157, 173
\ETE@WarnModDate	127, 131, 136
F	
\filename@base	213, 239
\filename@parse	210
G	
\Gin@base	201, 203, 207, 210, 238
\Gin@driver	92, 287
\Gin@eext	201
\Gin@ext	200, 203
\Gin@setfile	182, 195
\GraphicsRead	199, 245
\GraphicsType	198, 243
I	
\if	196
\ifcase	9
\ifETE@prepend	117, 268
\ifETE@update	126, 141
\ifETE@verbose	219, 229
\ifnum	106, 142, 300, 308
\ifx	10, 12,
	18, 39, 47, 82, 92, 104, 124, 160,
	174, 181, 200, 206, 236, 255, 262
\immediate	20, 41, 228
\includegraphics	224
\input	286, 288, 327
\InputIfFileExists	282
L	
\loop	295, 306
M	
\MessageBreak	133, 166,
	176, 186, 189, 191, 221, 222, 223
N	
\newcommand	86, 98, 122
\newif	117
O	
\OutputDirectory	205, 206, 213, 236, 239, 262
\OutputFile	207, 212, 222, 227, 231, 247, 264

P		S	
\PackageInfo	23	\setkeys	122
\pdffilemoddate	133, 142, 156	\SetupKeyvalOptions	114
\pdffilesize	157	\SourceFile	201, 203, 221, 227
\pdfshellescape	106	\space	133, 167, 177, 185, 264
\pdfstrcmp	142	T	
\PrependGraphicsExtensions ..	269, 276	\Test	312, 330
\ProcessKeyvalOptions	121	\the	56, 62, 298
\ProvidesPackage	53	\TMP@EnsureCode	59, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77
R		W	
\RangeCatcodeInvalid	304, 313, 314, 315, 316	\write	20, 41, 228
\repeat	302, 310	X	
\RequirePackage	78, 79, 80, 190	\x	8, 10, 12, 19, 23, 25, 40, 45, 52, 91, 92, 211, 216, 235, 250
\resetatcatcode	289		
\RestoreCatcodes ..	293, 296, 297, 328		