

The `epstopdf` package

Heiko Oberdiek*

2019/11/27 v2.8

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	3
1.3	Usage	3
1.4	Options	4
1.5	Configuration	5
1.5.1	System configuration file <code>epstopdf-sys.cfg</code>	5
1.5.2	User configuration file <code>epstopdf.cfg</code>	5
1.5.3	Conversion program	5
1.6	Other image formats	6
2	Implementation	6
2.1	Wrapper package	6
2.1.1	Option handling	8
2.2	Base package	9
2.3	Preparations	9
2.3.1	Relead check and identification	9
2.3.2	Catcodes	10
2.3.3	Load packages	11
2.4	Checks	12
2.5	Options	12
2.5.1	Default setting	13
2.6	Make and verbose	13
2.7	Adding conversion support	14
2.8	Declare graphics rule	17
3	Test	18
3.1	Preface for standard catcode check	18
3.2	Catcode checks for loading	18

*Please report any issues at <https://github.com/ho-tex/epstopdf/issues>

4	Installation	20
4.1	Download	20
4.2	Bundle installation	20
4.3	Package installation	20
4.4	Refresh file name databases	21
4.5	Some details for the interested	21
5	History	21
[2001/01/06 v1.0]		21
[2001/02/04 v1.1]		21
[2006/02/20 v1.2]		22
[2006/08/26 v1.3]		22
[2007/04/26 v1.4]		22
[2007/10/02 v1.5]		22
[2007/11/11 v1.6]		22
[2008/05/06 v1.7]		22
[2009/03/01 v1.8]		22
[2009/07/06 v1.9]		22
[2009/07/07 v1.10]		22
[2009/07/12 v2.0]		23
[2009/07/15 v2.1]		23
[2009/07/16 v2.2]		23
[2009/09/24 v2.3]		23
[2009/10/17 v2.4]		23
[2016/05/15 v2.5]		23
[2016/05/15 v2.6]		23
[2019/11/24 v2.7]		23
[2019/11/27 v2.8]		23
6	Index	24

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, pdfT_EX 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 pdfT_EX. 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 . Mik \TeX and \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 (Mik $\TeX$ )
```

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 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`.

program@epstopdf: The name for the conversion program from EPS to PDF, default is `'epstopdf'`.

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

1.5 Configuration

1.5.1 System configuration file `epstopdf-sys.cfg`

If `epstopdf-sys` exists it is loaded at the end of the package `epstopdf-base` and before the user configuration file. It's intended for TeX distributors. Thus they could add additional conversion rules (e.g., `.gif` -> `.png`) or set options.

1.5.2 User configuration file `epstopdf.cfg`

A configuration file `epstopdf.cfg` is loaded at the end of the package if it exists. It can be used for changing the default option setting. Example:

```
\epstopdfsetup{verbose=false}
```

1.5.3 Conversion program

You can use `\DeclareGraphicsRule` in a similar way as the route via `dvips` to specify the conversion command line. The conversion argument starts with a back tick, followed by the conversion command including parameters.

The whole conversion argument should also be wrapped inside `\epstopdfcall`. This reduces the problem with packages (e.g. `pst-pdf`) that use the conversion argument and expands it. Macros `\SourceFile`, `\OutputFile`, and `\SourceExt` are not defined outside `epstopdf-base`'s `\Gin@setfile` and error messages because of undefined command names are the result. If `\epstopdfcall` detects that it is called outside `epstopdf-base`'s `\Gin@setfile` then it replaces the conversion argument by package `graphics`'s default, usually the image file.

The following macros are available inside:

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

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

`\SourceExt:` : source extension without leading dot.

Conversion from EPS to PDF. Other programs than `epstopdf` can be used to convert from EPS to PDF. Example that uses Ghostscript:

```
\DeclareGraphicsRule{.eps}{pdf}{.pdf}{%
  \epstopdfcall{'ps2pdf -dEPSCrop #1 \noexpand\OutputFile}%
}
```

`\DeclareGraphicsRule` expands the argument, therefore `\noexpand` is necessary. As convenience package `epstopdf-base` defines `\epstopdfDeclareGraphicsRule`. Then the conversion argument is not expanded, `\epstopdfcall` and the back tick are added:

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

Also `\OutputFile` respects the setting of option `outdir`.

1.6 Other image formats

The support that package `epstopdf` implements is not limited to EPS files. Other image conversions can be declared. The following example shows it for GIF images under Unix with ImageMagick's `convert`:

```
\epstopdfDeclareGraphicsRule{.gif}{png}{.png}{%
  convert #1 \OutputFile
}
```

The file extension `.gif` can be added to the extension list that package `graphics` searches if the file extension is not given in `\includegraphics`. The list can be set by `\GraphicsExtensions`.

```
\AppendGraphicsExtensions{.gif}
or
\PrependGraphicsExtensions{.gif}
```

2 Implementation

```
1 (*package)
```

2.1 Wrapper package

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

```
2 \begingroup\catcode61\catcode48\catcode32=10\relax%
3 \catcode13=5 % ^M
4 \endlinechar=13 %
5 \catcode35=6 % #
6 \catcode39=12 % '
7 \catcode44=12 % ,
8 \catcode45=12 % -
9 \catcode46=12 % .
10 \catcode58=12 % :
11 \catcode64=11 % @
```

```

12 \catcode123=1 % {
13 \catcode125=2 % }
14 \expandafter\let\expandafter\x\csname ver@epstopdf.sty\endcsname
15 \ifx\x\relax % plain-TeX, first loading
16 \else
17 \def\empty{}%
18 \ifx\x\empty % LaTeX, first loading,
19 % variable is initialized, but \ProvidesPackage not yet seen
20 \else
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 \aftergroup\endinput
30 \fi
31 \fi
32 \endgroup%

```

Package identification:

```

33 \begingroup\catcode61\catcode48\catcode32=10\relax%
34 \catcode13=5 % ^~M
35 \endlinechar=13 %
36 \catcode35=6 % #
37 \catcode39=12 % '
38 \catcode40=12 % (
39 \catcode41=12 % )
40 \catcode44=12 % ,
41 \catcode45=12 % -
42 \catcode46=12 % .
43 \catcode47=12 % /
44 \catcode58=12 % :
45 \catcode64=11 % @
46 \catcode91=12 % [
47 \catcode93=12 % ]
48 \catcode123=1 % {
49 \catcode125=2 % }
50 \expandafter\ifx\csname ProvidesPackage\endcsname\relax
51 \def\x#1#2#3[#4]{\endgroup
52 \immediate\write-1{Package: #3 #4}%
53 \xdef#1{#4}%
54 }%
55 \else
56 \def\x#1#2[#3]{\endgroup
57 #2[#{#3}]%
58 \ifx#1\@undefined
59 \xdef#1{#3}%
60 \fi
61 \ifx#1\relax
62 \xdef#1{#3}%
63 \fi
64 }%
65 \fi
66 \expandafter\x\csname ver@epstopdf.sty\endcsname
67 \ProvidesPackage{epstopdf}%
68 [2019/11/27 v2.8 Conversion with epstopdf on the fly (H0)]%

```

Larger catcode set because of configuration files needed.

```

69 \begingroup\catcode61\catcode48\catcode32=10\relax%
70 \catcode13=5 % ^^M
71 \endlinechar=13 %
72 \catcode123=1 % {
73 \catcode125=2 % }
74 \catcode64=11 % @
75 \def\x{\endgroup
76 \expandafter\edef\csname ETE@AtEnd\endcsname{%
77 \endlinechar=\the\endlinechar\relax
78 \catcode13=\the\catcode13\relax
79 \catcode32=\the\catcode32\relax
80 \catcode35=\the\catcode35\relax
81 \catcode61=\the\catcode61\relax
82 \catcode64=\the\catcode64\relax
83 \catcode123=\the\catcode123\relax
84 \catcode125=\the\catcode125\relax
85 }%
86 }%
87 \x\catcode61\catcode48\catcode32=10\relax%
88 \catcode13=5 % ^^M
89 \endlinechar=13 %
90 \catcode35=6 % #
91 \catcode64=11 % @
92 \catcode123=1 % {
93 \catcode125=2 % }
94 \def\TMP@EnsureCode#1#2{%
95 \edef\ETE@AtEnd{%
96 \ETE@AtEnd
97 \catcode#1=\the\catcode#1\relax
98 }%
99 \catcode#1=#2\relax
100 }
101 \TMP@EnsureCode{39}{12}% '
102 \TMP@EnsureCode{40}{12}% (
103 \TMP@EnsureCode{41}{12}% )
104 \TMP@EnsureCode{42}{12}% *
105 \TMP@EnsureCode{45}{12}% -
106 \TMP@EnsureCode{47}{12}% /
107 \TMP@EnsureCode{91}{12}% [
108 \TMP@EnsureCode{93}{12}% ]
109 \edef\ETE@AtEnd{\ETE@AtEnd\noexpand\endinput}

```

Ensure packages loaded by the full epstopdf, for compatibility.

```

110 \let\ETE@SavedAtEnd\ETE@AtEnd
111 \RequirePackage{infwarerr}[2007/09/09]
112 \RequirePackage{grfext}\relax
113 \RequirePackage{kvoptions}[2007/10/02]
114 \RequirePackage{pdftexcmds}[2007/11/11]
115 \RequirePackage{epstopdf-base}[2019/11/27]
116 \let\ETE@AtEnd\ETE@SavedAtEnd

```

2.1.1 Option handling

```

117 \DeclareOption*{%
118 \expandafter\epstopdfsetup\expandafter{\CurrentOption}%
119 }
120 \ProcessOptions*\relax

```



```

121 \ETE@AtEnd%
122 \endpackage

```

2.2 Base package

```

123 \*base

```

2.3 Preparations

2.3.1 Reread check and identification

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

```

124 \begingroup\catcode61\catcode48\catcode32=10\relax%
125 \catcode13=5 % ^~M
126 \endlinechar=13 %
127 \catcode35=6 % #
128 \catcode39=12 % '
129 \catcode44=12 % ,
130 \catcode45=12 % -
131 \catcode46=12 % .
132 \catcode58=12 % :
133 \catcode64=11 % @
134 \catcode123=1 % {
135 \catcode125=2 % }
136 \expandafter\let\expandafter\x\csname ver@epstopdf-base.sty\endcsname
137 \ifx\x\relax % plain-TeX, first loading
138 \else
139 \def\empty{}%
140 \ifx\x\empty % LaTeX, first loading,
141 % variable is initialized, but \ProvidesPackage not yet seen
142 \else
143 \expandafter\ifx\csname PackageInfo\endcsname\relax
144 \def\x#1#2{%
145 \immediate\write-1{Package #1 Info: #2.}%
146 }%
147 \else
148 \def\x#1#2{\PackageInfo{#1}{#2, stopped}}%
149 \fi
150 \x{epstopdf-base}{The package is already loaded}%
151 \aftergroup\endinput
152 \fi
153 \fi
154 \endgroup%

```

Package identification:

```

155 \begingroup\catcode61\catcode48\catcode32=10\relax%
156 \catcode13=5 % ^~M
157 \endlinechar=13 %
158 \catcode35=6 % #
159 \catcode39=12 % '
160 \catcode40=12 % (
161 \catcode41=12 % )
162 \catcode44=12 % ,
163 \catcode45=12 % -
164 \catcode46=12 % .
165 \catcode47=12 % /
166 \catcode58=12 % :
167 \catcode64=11 % @
168 \catcode91=12 % [
169 \catcode93=12 % ]

```

```

170 \catcode123=1 % {
171 \catcode125=2 % }
172 \expandafter\ifx\csname ProvidesPackage\endcsname\relax
173 \def\x#1#2#3[#4]{\endgroup
174 \immediate\write-1{Package: #3 #4}%
175 \xdef#1{#4}%
176 }%
177 \else
178 \def\x#1#2[#3]{\endgroup
179 #2[#3]}%
180 \ifx#1\undefined
181 \xdef#1{#3}%
182 \fi
183 \ifx#1\relax
184 \xdef#1{#3}%
185 \fi
186 }%
187 \fi
188 \expandafter\x\csname ver@epstopdf-base.sty\endcsname
189 \ProvidesPackage{epstopdf-base}%
190 [2019/11/27 v2.8 Base part for package epstopdf]%

```

2.3.2 Catcodes

```

191 \begingroup\catcode61\catcode48\catcode32=10\relax%
192 \catcode13=5 % ^~M
193 \endlinechar=13 %
194 \catcode123=1 % {
195 \catcode125=2 % }
196 \catcode64=11 % @
197 \def\x{\endgroup
198 \expandafter\edef\csname ETE@AtEnd\endcsname{%
199 \endlinechar=\the\endlinechar\relax
200 \catcode13=\the\catcode13\relax
201 \catcode32=\the\catcode32\relax
202 \catcode35=\the\catcode35\relax
203 \catcode61=\the\catcode61\relax
204 \catcode64=\the\catcode64\relax
205 \catcode123=\the\catcode123\relax
206 \catcode125=\the\catcode125\relax
207 }%
208 }%
209 \x\catcode61\catcode48\catcode32=10\relax%
210 \catcode13=5 % ^~M
211 \endlinechar=13 %
212 \catcode35=6 % #
213 \catcode64=11 % @
214 \catcode123=1 % {
215 \catcode125=2 % }
216 \def\TMP@EnsureCode#1#2{%
217 \edef\ETE@AtEnd{%
218 \ETE@AtEnd
219 \catcode#1=\the\catcode#1\relax
220 }%
221 \catcode#1=#2\relax
222 }
223 \TMP@EnsureCode{33}{12}% !
224 \TMP@EnsureCode{39}{12}% '

```

```

225 \TMP@EnsureCode{42}{12}% *
226 \TMP@EnsureCode{44}{12}% ,
227 \TMP@EnsureCode{45}{12}% -
228 \TMP@EnsureCode{46}{12}% .
229 \TMP@EnsureCode{47}{12}% /
230 \TMP@EnsureCode{58}{12}% :
231 \TMP@EnsureCode{60}{12}% <
232 \TMP@EnsureCode{62}{12}% >
233 \TMP@EnsureCode{96}{12}% '
234 \edef\ETE@AtEnd{\ETE@AtEnd\noexpand\endinput}

```

2.3.3 Load packages

This package is split into `epstopdf-base` and `epstopdf`. The base version is called directly by the `graphics` package. Unfortunately it still includes many contributed packages which breaks the layering of the core L^AT_EX graphics release.

Compatibility concerns mean refactoring the package to only have options in `epstopdf` is tricky, so from release 2.8, if no options have been passed in to `epstopdf-base`, the package loading and option setting are skipped and the default settings are set directly. In the case of LuaT_EX, the Lua portion of `pdftexcmds` is used to provide the file handling functionality.

```

235 \ifx\@curroptions\@empty
236   \ifx\@PackageInfo\@undefined
237     \let\@PackageInfo\PackageInfo
238     \let\@PackageWarningNoLine\PackageWarningNoLine
239     \def\@PackageInfoNoLine#1#2{\PackageInfo{#1}{#2@gobble}}
240     \let\@PackageError\PackageError
241   \fi
242   \ifx\pdf@strcmp\@undefined
243     \ifx\directlua\@undefined
244       \def\pdf@strcmp{\pdfstrcmp}
245       \def\pdf@filemoddate{\pdffilemoddate}
246       \def\pdf@filesize{\pdffilesize}
247       \def\pdf@system#{\immediate\write18 }
248     \else
249       \directlua{require("pdftexcmds")}
250       \long\def\pdf@strcmp#1#2{\directlua{%
251         oberdiek.pdftexcmds.strptime("\luaescapestring{#1}",
252           "\luaescapestring{#2}")}}%
253       \def\pdf@filemoddate#1{\directlua{%
254         oberdiek.pdftexcmds.filemoddate("\luaescapestring{#1}")}}
255       \def\pdf@filesize#1{\directlua0{%
256         oberdiek.pdftexcmds.filesize("\luaescapestring{#1}")}}
257       \def\pdf@system#1{\directlua{%
258         oberdiek.pdftexcmds.system("\luaescapestring{#1}")}}
259     \fi
260   \fi
261 \else
262   \RequirePackage{infwarerr}[2007/09/09]
263   \RequirePackage{grfext}\relax
264   \RequirePackage{kvoptions}[2007/10/02]
265   \RequirePackage{pdftexcmds}[2007/11/11]
266 \fi

```

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.

```
267 \begingroup\expandafter\expandafter\expandafter\endgroup
268 \expandafter\ifx\csname Gin@setfile\endcsname\relax
269   \@PackageWarningNoLine{epstopdf}{%
270     No graphics package \string'graphic{s,x}\string' loaded%
271   }%
272   \newcommand*{\epstopdfsetup}[1]{}%
273   \expandafter\ETE@AtEnd
274 \fi%
```

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

```
275 \begingroup
276   \def\x{luatex.def}%
277   \ifx\Gin@driver\x
278   \else
279     \def\x{pdftex.def}%
280     \ifx\Gin@driver\x
281     \else
282       \@PackageWarningNoLine{epstopdf}{%
283         Drivers other than 'pdftex' and 'luatex' are not supported%
284       }%
285     \endgroup
286     \newcommand*{\epstopdfsetup}[1]{}%
287     \expandafter\expandafter\expandafter\ETE@AtEnd
288   \fi%
289 \fi%
290 \endgroup
```

Check, whether the shell escape feature is enabled.

```
291 \begingroup
292   \expandafter\ifx\csname pdf@shellescape\endcsname\relax
293   \else
294     \ifnum\pdf@shellescape>0 %
295     \else
296       \@PackageWarningNoLine{epstopdf}{%
297         Shell escape feature is not enabled%
298       }%
299     \fi
300   \fi
301 \endgroup
```

2.5 Options

As noted above, if no options have been passed in (typically if called directly from `pdftex.def`) then the `kvoptions` handling is not loaded and the defaults are set directly.

```
302 \newif\ifETE@prepend
303 \ifx\SetupKeyvalOptions\undefined
304   \def\ETE@let#1#2{%
305     \expandafter\let\csname ifETE@#1\expandafter\endcsname
306     \csname if#2\endcsname}
307   \ETE@let{verbose}{true}
308   \ETE@let{disable}{false}
309   \ETE@let{update}{true}
```

```

310 \ETE@let{prepend}{false}
311 \ETE@let{prefersuffix}{true}
312 \def\ETE@outdir{}
313 \def\ETE@suffix{-\SourceExt-converted-to}
314 \def\ETE@program@epstopdf{\epstopdf@sys@cmd}
315 \newcommand*{\epstopdfsetup}[1]{}%
316 \else
317 \SetupKeyvalOptions{family=ETE,prefix=ETE@}
318 \DeclareBoolOption{update}
319 \DeclareBoolOption{verbose}
320 \DeclareVoidOption{prepend}{\ETE@prependtrue}
321 \DeclareVoidOption{append}{\ETE@prependfalse}
322 \DeclareStringOption{outdir}
323 \DeclareStringOption{suffix}
324 \DeclareBoolOption{prefersuffix}
325 \DeclareStringOption{program@epstopdf}

```

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

```

326 \DeclareBoolOption{disable}
327 \DeclareComplementaryOption{enable}{disable}
328 \newcommand*{\epstopdfsetup}{\setkeys{ETE}}

```

2.5.1 Default setting

```

329 \epstopdfsetup{%
330     verbose,%
331     enable,%
332     append,%
333     update,%
334     prefersuffix,%
335     suffix=-\SourceExt-converted-to,%
336     program@epstopdf=epstopdf%
337 }
338 \fi

```

2.6 Make and verbose

```

339 \begingroup\expandafter\expandafter\expandafter\endgroup
340 \expandafter\ifx\csname pdf@filemoddate\endcsname\relax
341 \def\ETE@Make#1#2{%
342     \ifETE@update
343         \ETE@WarnModDate
344     \fi
345     \@firstofone
346 }%
347 \def\ETE@WarnModDate{%
348     \@PackageWarningNoLine{epstopdf}{%
349         \string\pdf@filemoddate\space is not available,\MessageBreak
350         option 'update' will be ignored%
351     }%
352     \global\let\ETE@WarnModDate\relax
353 }%
354 \def\ETE@FileInfo#1#2{#1 file: <#2>}%
355 \else
356 \def\ETE@Make#1#2{%
357     \ifETE@update
358         \ifnum\pdf@strcmp{\pdf@filemoddate{#1}}{\pdf@filemoddate{#2}}>0 %

```

```

359     \expandafter\expandafter\expandafter\@firstofone
360   \else
361     \@PackageInfoNoLine{epstopdf}{%
362       Output file is already uptodate%
363     }%
364     \expandafter\expandafter\expandafter\@gobble
365   \fi
366 \else
367   \expandafter\@firstofone
368 \fi
369 }%
370 \def\ETE@FileInfo#1#2{%
371   #1 file: <#2>%
372   \expandafter\expandafter\expandafter
373   \ETE@Date\pdf@filemoddate{#2}\@nil
374   \expandafter\expandafter\expandafter
375   \ETE@Size\pdf@filesize{#2}\@nil
376 }%
377 \def\ETE@Date#1\@nil{%
378   \ifx\#1\%
379   \else
380     \ETE@@Date#1\@nil
381   \fi
382 }%
383 \def\ETE@@Date#1:#2#3#4#5#6#7#8#9{%
384   \MessageBreak
385   \@spaces\space\space\space date: #2#3#4#5-#6#7-#8#9 %
386   \ETE@@Time
387 }%
388 \def\ETE@@Time#1#2#3#4#5#6#7\@nil{%
389   #1#2:#3#4:#5#6%
390 }%
391 \def\ETE@Size#1\@nil{%
392   \ifx\#1\%
393   \else
394     \MessageBreak
395     \@spaces\space\space\space size: #1 bytes%
396   \fi
397 }%
398 \fi

```

2.7 Adding conversion support

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

```

399 \expandafter\ifx\csname ETE@OrgGin@setfile\endcsname\relax
400   \let\ETE@OrgGin@setfile\Gin@setfile
401 \else
402   \@PackageError{epstopdf}{%
403     Command \string\ETE@OrgGin@setfile\space
404     already defined.\MessageBreak
405   }{%
406     Probably some package has included the code of this package%
407     \MessageBreak
408     instead of using \string\RequirePackage{epstopdf}%.%
409     \MessageBreak
410     \@ehc
411   }%
412 \fi
413 \def\ETE@ifFileExists{%

```

```

414 \begingroup\expandafter\expandafter\expandafter\endgroup
415 \expandafter\ifx\csname grffile@ifFileExists\endcsname\relax
416 \expandafter\ifFileExists
417 \else
418 \global\let\ETE@ifFileExists\grffile@ifFileExists
419 \expandafter\grffile@ifFileExists
420 \fi
421 }
422 \def\ETE@Skip#1#2\x\fi\fi{%
423 \fi
424 \fi
425 \endgroup
426 \fi
427 \fi
428 #1%
429 }
430 \newif\ifETE@InsideSetfile
431 \newcommand*{\epstopdfcall}[1]{%
432 \ifETE@InsideSetfile
433 \expandafter\@firstoftwo
434 \else
435 \expandafter\@secondoftwo
436 \fi
437 {'#1}%
438 {\Gin@base\Gin@ext}%
439 }
440 \def\ETE@DefCommandLine#1{%
441 \edef\CommandLine{\expandafter\fi\if'#1}%
442 }
443 \def\ETE@DefX#1{%
444 \expandafter\expandafter\expandafter\def
445 \expandafter\expandafter\expandafter\x
446 \expandafter\expandafter\expandafter{%
447 \expandafter\fi\if'#1\relax\else
448 }%
449 }
450 \def\ETE@Gin@setfile#1#2#3{%
451 \ifETE@disable
452 \ETE@OrgGin@setfile{#1}{#2}{#3}%
453 \else
454 \begingroup
455 \ETE@InsideSetfiletrue
456 \ETE@DefX{#3}%
457 \expandafter\endgroup
458 \ifx\x\@empty
459 \ETE@OrgGin@setfile{#1}{#2}{#3}%
460 \else
461 \begingroup
462 \ETE@InsideSetfiletrue
463 \def\GraphicsType{#1}%
464 \def\GraphicsRead{#2}%
465 \ifETE@prefersuffix
466 \else
467 \ifx\ETE@suffix\@empty
468 \else
469 \ETE@ifFileExists{\Gin@base\GraphicsRead}{%
470 \ETE@Skip{%
471 \ETE@OrgGin@setfile{#1}{#2}{\Gin@base#2}%

```

```

472         }%
473     }{%
474         \let\next\relax
475     }%
476     \next
477 \fi
478 \fi
479 \ifx\Gin@ext\relax
480     \let\SourceExt\Gin@eext
481     \def\SourceFile{\Gin@base\Gin@eext}%
482 \else
483     \let\SourceExt\Gin@ext
484     \def\SourceFile{\Gin@base\Gin@ext}%
485 \fi
486 \edef\SourceExt{% remove dot
487     \expandafter\@cdr\SourceExt\@empty\@nil
488 }%
489 \let\OutputDirectory\ETE@outdir
490 \ifx\OutputDirectory\@empty
491     \edef\OutputFile{\ETE@GenerateName{\Gin@base}{#2}}%
492 \else
493     \beginingroup
494         \filename@parse{\Gin@base#2}%
495         \edef\x{\endgroup
496             \def\noexpand\OutputFile{%
497                 \ETE@GenerateName{%
498                     \OutputDirectory\filename@base
499                 }{#2}%
500             }%
501         }%
502     \x
503 \fi
504 \ETE@DefCommandLine{#3}%
505 \ifETE@verbose
506     \@PackageInfo{epstopdf}{%
507         \ETE@FileInfo{Source}\SourceFile\MessageBreak
508         \ETE@FileInfo{Output}\OutputFile\MessageBreak
509         Command: <\CommandLine>\MessageBreak
510         \string\includegraphics
511     }%
512 \fi
513 \ETE@Make\SourceFile\OutputFile{%
514     \pdf@system{\CommandLine}%
515     \ifETE@verbose
516         \@PackageInfoNoLine{epstopdf}{%
517             \ETE@FileInfo{Result}\OutputFile
518         }%
519     \fi
520 }%
521 \edef\x{\endgroup
522     \ifx\OutputDirectory\@empty
523     \else
524         \def\noexpand\Gin@base{%
525             \OutputDirectory\noexpand\filename@base
526         }%
527     \fi
528     \ifx\ETE@suffix\@empty
529     \else

```



```

530         \edef\noexpand\Gin@base{%
531             \noexpand\Gin@base\ETE@suffix
532         }%
533     \fi
534     \noexpand\ETE@OrgGin@setfile{%
535         \GraphicsType
536     }{%
537         \GraphicsRead
538     }{%
539         \OutputFile
540     }%
541 }%
542 \x
543 \fi
544 \fi
545 }
546 \let\Gin@setfile\ETE@Gin@setfile
547 \def\ETE@GenerateName#1#2{%
548     #1\ETE@suffix#2%
549 }

```

2.8 Declare graphics rule

```

550 \newcommand*{\epstopdfDeclareGraphicsRule}[4]{%
551     \ifx\#4\%
552         \@PackageError{epstopdf-base}{%
553             Conversion command is missing%
554         }\@ehc
555     \else
556         \begingroup
557         \ifundefined{Gin@rule@#1}{%
558             }{%
559                 \@PackageInfo{epstopdf-base}{%
560                     Redefining graphics rule for ‘#1’%
561                 }%
562             }%
563         \endgroup
564         \@namedef{Gin@rule@#1}##1{##2}{##3}{\epstopdfcall{##4}}}%
565     \fi
566 }

\DeclareGraphicsRule for .eps
567 \epstopdfDeclareGraphicsRule{.eps}{pdf}{.pdf}{%
568     \ETE@epstopdf{#1}%
569 }
570 \def\ETE@epstopdf#1{%
571     \ETE@program@epstopdf\space
572     \ifcase\ifx\OutputDirectory\@empty
573         \ifx\ETE@suffix\@empty
574             1%
575         \fi
576     \fi
577     0 %
578     --outfile=\OutputFile\space
579 \fi
580 #1%
581 }
582 \ifx\AppendGraphicsExtensions\@undefined\else

```

```

583 \ifETE@prepend
584   \expandafter\PrependGraphicsExtensions
585 \else
586   \expandafter\AppendGraphicsExtensions
587 \fi
588 {.eps}
589 \fi

590 \let\ETE@prepend\@undefined

591 \ifx\SetupKeyvalOptions\@undefined
592 \InputIfFileExists{epstopdf-sys.cfg}{}{}
593 \else

594 \DeclareVoidOption{prepend}{%
595   \PrependGraphicsExtensions{.eps}%
596 }
597 \let\ETE@append\@undefined
598 \DeclareVoidOption{append}{%
599   \AppendGraphicsExtensions{.eps}%
600 }

601 \InputIfFileExists{epstopdf-sys.cfg}{}{}
602 \InputIfFileExists{epstopdf.cfg}{}{}

603 \fi

604 \ETE@AtEnd%
605 </base>

```

3 Test

3.1 Preface for standard catcode check

```

606 <*test1>
607 \input miniltx.tex\relax
608 \def\Gin@driver{pdftex.def}
609 \input graphicx.sty\relax
610 \resetatcatcode
611 </test1>

```

3.2 Catcode checks for loading

```

612 <*test1>

613 \catcode'\{=1 %
614 \catcode'\}=2 %
615 \catcode'\#=6 %
616 \catcode'\@=11 %
617 \expandafter\ifx\csname count@\endcsname\relax
618   \countdef\count@=255 %
619 \fi

620 \expandafter\ifx\csname @gobble\endcsname\relax
621   \long\def\@gobble#1{}%
622 \fi

623 \expandafter\ifx\csname @firstofone\endcsname\relax
624   \long\def\@firstofone#1{#1}%
625 \fi

626 \expandafter\ifx\csname loop\endcsname\relax
627   \expandafter\@firstofone
628 \else
629   \expandafter\@gobble
630 \fi

```

```

631 {%
632   \def\loop#1\repeat{%
633     \def\body{#1}%
634     \iterate
635   }%
636   \def\iterate{%
637     \body
638     \let\next\iterate
639   \else
640     \let\next\relax
641   \fi
642   \next
643 }%
644 \let\repeat=\fi
645 }%
646 \def\RestoreCatcodes{}
647 \count@=0 %
648 \loop
649   \edef\RestoreCatcodes{%
650     \RestoreCatcodes
651     \catcode\the\count@=\the\catcode\count@\relax
652   }%
653 \ifnum\count@<255 %
654   \advance\count@ 1 %
655 \repeat
656
657 \def\RangeCatcodeInvalid#1#2{%
658   \count@=#1\relax
659   \loop
660     \catcode\count@=15 %
661     \ifnum\count@<#2\relax
662       \advance\count@ 1 %
663     \repeat
664 }
665 \def\RangeCatcodeCheck#1#2#3{%
666   \count@=#1\relax
667   \loop
668     \ifnum#3=\catcode\count@
669   \else
670     \errmessage{%
671       Character \the\count@\space
672       with wrong catcode \the\catcode\count@\space
673       instead of \number#3%
674     }%
675   \fi
676   \ifnum\count@<#2\relax
677     \advance\count@ 1 %
678   \repeat
679 }
680 \def\space{ }
681 \expandafter\ifx\csname LoadCommand\endcsname\relax
682   \def\LoadCommand{\input epstopdf.sty\relax}%
683 \fi
684 \def\Test{%
685   \RangeCatcodeInvalid{0}{47}%
686   \RangeCatcodeInvalid{58}{64}%
687   \RangeCatcodeInvalid{91}{96}%
688   \RangeCatcodeInvalid{123}{255}%

```

```

689 \catcode'\@=12 %
690 \catcode'\=0 %
691 \catcode'\%=14 %
692 \LoadCommand
693 \RangeCatcodeCheck{0}{36}{15}%
694 \RangeCatcodeCheck{37}{37}{14}%
695 \RangeCatcodeCheck{38}{47}{15}%
696 \RangeCatcodeCheck{48}{57}{12}%
697 \RangeCatcodeCheck{58}{63}{15}%
698 \RangeCatcodeCheck{64}{64}{12}%
699 \RangeCatcodeCheck{65}{90}{11}%
700 \RangeCatcodeCheck{91}{91}{15}%
701 \RangeCatcodeCheck{92}{92}{0}%
702 \RangeCatcodeCheck{93}{96}{15}%
703 \RangeCatcodeCheck{97}{122}{11}%
704 \RangeCatcodeCheck{123}{255}{15}%
705 \RestoreCatcodes
706 }
707 \Test
708 \csname @@end\endcsname
709 \end
710 </test1>

```

4 Installation

4.1 Download

Package. This package is available on CTAN¹:

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

[CTAN:macros/latex/contrib/epstopdf-pkg/epstopdf.pdf](#) Documentation.

Bundle. All the packages of the bundle ‘epstopdf’ 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/epstopdf-pkg.tds.zip](#)

TDS refers to the standard “A Directory Structure for \TeX Files” ([CTAN:pkg/tds](#)). Directories with `texmf` in their name are usually organized this way.

4.2 Bundle installation

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

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

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
```

¹[CTAN:pkg/epstopdf-pkg](#)

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/epstopdf/epstopdf.sty
epstopdf-base.sty → tex/latex/epstopdf/epstopdf-base.sty
epstopdf.pdf      → doc/latex/epstopdf/epstopdf.pdf
test/epstopdf-test1.tex → doc/latex/epstopdf/test/epstopdf-test1.tex
epstopdf.dtx      → source/latex/epstopdf/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 (\TeX Live, `mikTeX`, ...) relies on file name databases, you must refresh these. For example, \TeX Live users run `texhash` or `mktextlsr`.

4.5 Some details for the interested

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 `pdf \TeX` mailing list.

[2001/02/04 v1.1]

- Minor documentation update.
- CTAN.

[2006/02/20 v1.2]

- DTX framework.
- Compatibility for `miniltx.tex`.

[2006/08/26 v1.3]

- Check for `\write18` if available and print a warning if the feature is not enabled.

[2007/04/26 v1.4]

- Documentation rewritten and extended.

[2007/10/02 v1.5]

- New option `update`: If the converted file exists, it will be only converted if it is out of date.
- Updating the extension list is delegated to package `grfext`. Fine tuning is done by the new options `append`, `prepend`.
- New option `outdir` for changing the output directory.
- New option `verbose`.
- `\SourceFile` and `\OutputFile` introduced.
- Configuration file support added.

[2007/11/11 v1.6]

- Use of package `pdftexcmds` for LuaTeX support.

[2008/05/06 v1.7]

- Warning messages uses “loaded” instead of “found”.

[2009/03/01 v1.8]

- Warning message for missing `pdftex.def` changed.

[2009/07/06 v1.9]

- Option `suffix` added.

[2009/07/07 v1.10]

- `\SourceExt` added.
- If option `suffix` is set, the inclusion of an image without the suffix namespace is preferred over generating the the image within the suffix namespace.

[2009/07/12 v2.0]

- New default settings.
- Package is split into `epstopdf` that only takes package options and loads `epstopdf-base` that does the work.
- `\epstopdfDeclareGraphicsRule` and `\epstopdfcall` added.
- `epstopdf-sys.cfg` is loaded before `epstopdf.cfg` if `epstopdf-sys.cfg` exists.

[2009/07/15 v2.1]

- Default setting: `verbose` is now turned on as the documentation for v2.0 said.
- Documentation fixes.

[2009/07/16 v2.2]

- Fixed redefined `\Gin@setfile`.
- Documentation extended for package options.

[2009/09/24 v2.3]

- Bug fix for the case that both option `suffix` and `outdir` are used.

[2009/10/17 v2.4]

- The name of the program ‘`epstopdf`’ can be configured via the new option `program@epstopdf`.

[2016/05/15 v2.5]

- Wording of warning message fixed (Karl Berry).
- `\ETE@Gin@setfile` added (Karl Berry).

[2016/05/15 v2.6]

- luaTeX compatibility

[2019/11/24 v2.7]

- New `epstopdf` repository.

[2019/11/27 v2.8]

- The base package configured not to require additional packages if called with no arguments.
- Code re-arranged to avoid `P`
`\end` occurred when `\ifx` on line 165 was incomplete warning in dvi mode.

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; plain numbers refer to the code lines where the entry is used.

Symbols	
\#	615
\%	691
\@	616, 689
\@PackageError	240, 402, 552
\@PackageInfo	236, 237, 506, 559
\@PackageInfoNoLine	239, 361, 516
\@PackageWarningNoLine	238, 269, 282, 296, 348
\@cdr	487
\@curroptions	235
\@ehc	410, 554
\@empty	235, 458, 467, 487, 490, 522, 528, 572, 573
\@firstofone	345, 359, 367, 624, 627
\@firstoftwo	433
\@gobble	239, 364, 621, 629
\@ifundefined	557
\@namedef	564
\@nil	373, 375, 377, 380, 388, 391, 487
\@secondoftwo	435
\@spaces	385, 395
\@undefined	58, 180, 236, 242, 243, 303, 582, 590, 591, 597
\\	378, 392, 551, 690
\{	613
\}	614
A	
\advance	654, 662, 677
\aftergroup	29, 151
\AppendGraphicsExtensions	582, 586, 599
B	
\body	633, 637
C	
\catcode	2, 3, 5, 6, 7, 8, 9, 10, 11, 12, 13, 33, 34, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 69, 70, 72, 73, 74, 78, 79, 80, 81, 82, 83, 84, 87, 88, 90, 91, 92, 93, 97, 99, 124, 125, 127, 128, 129, 130, 131, 132, 133, 134, 135, 155, 156, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171, 191, 192, 194, 195, 196, 200, 201, 202, 203, 204, 205, 206, 209, 210, 212, 213, 214, 215, 219, 221, 613, 614, 615, 616, 651, 660, 668, 672, 689, 690, 691
\CommandLine	441, 509, 514
\count@	618, 647, 651, 653, 654, 658, 660, 661, 662, 666, 668, 671, 672, 676, 677
\countdef	618
\csname	14, 21, 50, 66, 76, 136, 143, 172, 188, 198, 268, 292, 305, 306, 340, 399, 415, 617, 620, 623, 626, 681, 708
\CurrentOption	118
D	
\DeclareBoolOption	318, 319, 324, 326
\DeclareComplementaryOption	327
\DeclareOption	117
\DeclareStringOption	322, 323, 325
\DeclareVoidOption	320, 321, 594, 598
\directlua	243, 249, 250, 253, 255, 257
E	
\empty	17, 18, 139, 140
\end	709
\endcsname	14, 21, 50, 66, 76, 136, 143, 172, 188, 198, 268, 292, 305, 306, 340, 399, 415, 617, 620, 623, 626, 681, 708
\endinput	29, 109, 151, 234
\endlinechar	4, 35, 71, 77, 89, 126, 157, 193, 199, 211
\epstopdf@sys@cmd	314
\epstopdfcall	431, 564
\epstopdfDeclareGraphicsRule	550, 567
\epstopdfsetup	4, 118, 272, 286, 315, 328, 329
\errmessage	670
\ETE@@Date	380, 383
\ETE@@Time	386, 388
\ETE@append	597
\ETE@AtEnd	95, 96, 109, 110, 116, 121, 217, 218, 234, 273, 287, 604
\ETE@Date	373, 377
\ETE@DefCommandLine	440, 504
\ETE@DefX	443, 456
\ETE@epstopdf	568, 570
\ETE@FileInfo	354, 370, 507, 508, 517
\ETE@GenerateName	491, 497, 547
\ETE@Gin@setfile	450, 546
\ETE@IfFileExists	413, 418, 469
\ETE@InsideSetfiletrue	455, 462
\ETE@let	304, 307, 308, 309, 310, 311

\ETE@Make	341, 356, 513		
\ETE@OrgGin@setfile	400, 403, 452, 459, 471, 534		
\ETE@outdir	312, 489		
\ETE@prepend	590		
\ETE@prependfalse	321		
\ETE@prependtrue	320		
\ETE@program@epstopdf	314, 571		
\ETE@SavedAtEnd	110, 116		
\ETE@Size	375, 391		
\ETE@Skip	422, 470		
\ETE@suffix	313, 467, 528, 531, 548, 573		
\ETE@WarnModDate	343, 347, 352		
F			
\filename@base	498, 525		
\filename@parse	494		
G			
\Gin@base	438, 469, 471, 481, 484, 491, 494, 524, 530, 531		
\Gin@driver	277, 280, 608		
\Gin@eext	480, 481		
\Gin@ext	438, 479, 483, 484		
\Gin@setfile	400, 546		
\GraphicsRead	464, 469, 537		
\GraphicsType	463, 535		
\grffile@IfFileExists	418, 419		
I			
\if	441, 447		
\ifcase	572		
\ifETE@disable	451		
\ifETE@InsideSetfile	430, 432		
\ifETE@prefersuffix	465		
\ifETE@prepend	302, 583		
\ifETE@update	342, 357		
\ifETE@verbose	505, 515		
\IfFileExists	416		
\ifnum	294, 358, 653, 661, 668, 676		
\ifx	15, 18, 21, 50, 58, 61, 137, 140, 143, 172, 180, 183, 235, 236, 242, 243, 268, 277, 280, 292, 303, 340, 378, 392, 399, 415, 458, 467, 479, 490, 522, 528, 551, 572, 573, 582, 591, 617, 620, 623, 626, 681		
\immediate	23, 52, 145, 174, 247		
\includegraphics	510		
\input	607, 609, 682		
\InputIfFileExists	592, 601, 602		
\iterate	634, 636, 638		
L			
\LoadCommand	682, 692		
\loop	632, 648, 659, 667		
\luaescapestring	251, 252, 254, 256, 258		
M			
\MessageBreak	349, 384, 394, 404, 407, 409, 507, 508, 509		
N			
\newcommand	272, 286, 315, 328, 431, 550		
\newif	302, 430		
\next	474, 476, 638, 640, 642		
\number	673		
O			
\OutputDirectory	489, 490, 498, 522, 525, 572		
\OutputFile	491, 496, 508, 513, 517, 539, 578		
P			
\PackageError	240		
\PackageInfo	26, 148, 237, 239		
\PackageWarningNoLine	238		
\pdf@filemoddate	245, 253, 358, 373		
\pdf@filesize	246, 255, 375		
\pdf@shellescape	294		
\pdf@strcmp	242, 244, 250, 358		
\pdf@system	247, 257, 514		
\pdf@filemoddate	245, 349		
\pdf@filesize	246		
\pdfstrcmp	244		
\PrependGraphicsExtensions	584, 595		
\ProcessOptions	120		
\ProvidesPackage	19, 67, 141, 189		
R			
\RangeCatcodeCheck	665, 693, 694, 695, 696, 697, 698, 699, 700, 701, 702, 703, 704		
\RangeCatcodeInvalid	657, 685, 686, 687, 688		
\repeat	632, 644, 655, 663, 678		
\RequirePackage	111, 112, 113, 114, 115, 262, 263, 264, 265, 408		
\resetatcatcode	610		
\RestoreCatcodes	646, 649, 650, 705		
S			
\setkeys	328		
\SetupKeyvalOptions	303, 317, 591		
\SourceExt	313, 335, 480, 483, 486, 487		
\SourceFile	481, 484, 507, 513		
\space	349, 385, 395, 403, 571, 578, 671, 672, 680		
T			
\Test	684, 707		
\the	77, 78, 79, 80, 81, 82, 83, 84, 97, 199, 200, 201, 202, 203, 204, 205, 206, 219, 651, 671, 672		
\TMP@EnsureCode	94, 101, 102, 103, 104, 105, 106, 107,		

```

108, 216, 223, 224, 225, 226,
227, 228, 229, 230, 231, 232, 233  \x ..... 14, 15, 18, 22, 26, 28,
                                         51, 56, 66, 75, 87, 136, 137, 140,
                                         144, 148, 150, 173, 178, 188,
                                         197, 209, 276, 277, 279, 280,
W                                         422, 445, 458, 495, 502, 521, 542
\write ..... 23, 52, 145, 174, 247

```