The epstopdf package

Heiko Oberdiek*

2020-01-24 v2.11

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	Doc	cumentation
	1.1	Introduction
	1.2	Requirements
	1.3	Usage
	1.4	Options
	1.5	Configuration
		1.5.1 System configuration file epstopdf-sys.cfg
		1.5.2 User configuration file epstopdf.cfg
		1.5.3 Conversion program
	1.6	Other image formats
2	Imr	plementation
	2.1	Wrapper package
		2.1.1 Option handling
	2.2	Base package
	2.3	Preparations
		2.3.1 Relead check and identification
		2.3.2 Catcodes
		2.3.3 Load packages
	2.4	Checks
	2.5	Options
		2.5.1 Default setting
	2.6	Make and verbose
	2.7	Adding conversion support
	2.8	Declare graphics rule
3	Inst	allation 18
•	3.1	Download
	3.2	Bundle installation
	3.3	Package installation
	3.4	Refresh file name databases
	3.5	Some details for the interested

^{*}Please report any issues at https://github.com/ho-tex/epstopdf/issues

4	History	1
	[2001/01/06 v1.0]	1
	[2001/02/04 v1.1]	1
	[2006/02/20 v1.2]	2
	[2006/08/26 v1.3]	2
	[2007/04/26 v1.4]	2
	$[2007/10/02 \text{ v}1.5] \dots \dots$	2
	[2007/11/11 v1.6]	2
	[2008/05/06 v1.7]	2
	[2009/03/01 v1.8]	2
	[2009/07/06 v1.9]	2
	[2009/07/07 v1.10]	2
	[2009/07/12 v2.0]	2
	[2009/07/15 v2.1]	2
	[2009/07/16 v2.2]	2
	[2009/09/24 v2.3]	2
	[2009/10/17 v2.4]	2
	[2016/05/15 v2.5]	2
	[2016/05/15 v2.6]	2
	[2019/11/24 v2.7]	2
	[2019/11/27 v2.8]	2
	[2019-11-30 v2.9]	2
	[2020-01-24 v2.11]	2
	[2020-01-24 v2.11]	2

1 Documentation

1.1 Introduction

5 Index

E^ATEX 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.

 $\mathbf{22}$

Usually the drivers do not support all kind of graphics files. Other image types must be converted, before they become usuable. In case of driver dvips, the graphics rule may contain a conversion rule. Then all that package graphics must know is the bounding box, the command is passed to dvips that calls it and embeds the converted image.

However, pdfTEX has its driver for PDF output already build in. It's graphics inclusion commands (\pdfximage) does not allow the execution of external commands. Therefore commands in the last argument of \DeclareGraphicsRule were of no use. But external programs can be called within pdfTEX. This feature is called "shell escape" or "write 18" and must usually enabled explicitely because of security reasons. Now, this package epstopdf hooks into package graphics' code to catch that argument with the external command and executes it to convert the graphics file to a supported format and passes the control of graphics inclusion back to package graphics.

1.2 Requirements

• The feature \write18 must be enabled. This allows the running of external programs during TeX's compile run. Keep in mind that this is a security

risk. The feature is an addition to TEX. MikTEX 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 (MiKTEX)
```

Example:

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

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

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

1.3 Usage

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

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

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

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

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

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

Package epstopdf puts .eps at the end of the graphics extension search list. This is the behaviour of option append that is enabled by default. That means, the conversion is called last unless a supported file type cannot be found earlier. This avoids unnecessary conversion steps that slow down the LATEX run. If you want to use option update and your pdfTEX supports it, then an outdated PDF file also would be found earlier unless suffix is used that is the default since version 2.0.

With an empty option suffix and option prepend there is a risk that an original PDF file is overwritten:

If the original image file is the PDF file and there is also a generated EPS file, then the original PDF file can be regenerated (depending on the option settings) and the original PDF file gets lost. Therefore option suffix is introduced in version 1.9 to create a separate name space for generated output files.

Note: Usually the conversion program needs the exact location of the image file. Usually the current directory works. Also if the image file is found using \graphicspath, the location is known. However, if the image is somewhere in a directory of environment variable TEXINPUTS, then the package does not know the exact location and the conversion program will not find the image file unless it implements a search using TEXINPUTS (program kpsewhich may be of help in this task).

1.4 Options

Options can be given as package options or later using:

\epstopdfsetup $\{\langle key\ value\ list \rangle\}$

IATEX 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. IATEX'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 'regnerate' 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%
    \catcode13=5 % ^^M
    \endlinechar=13 %
    \catcode35=6 % #
    \catcode39=12 % '
    \colone{1} \catcode44=12 % ,
    \catcode45=12 % -
    \catcode46=12 % .
10
    \catcode58=12 % :
    \catcode64=11 % @
    \catcode123=1 % {
    \catcode125=2 % }
    \expandafter\let\expandafter\x\csname ver@epstopdf.sty\endcsname
    \ifx\x\relax % plain-TeX, first loading
    \else
16
```

```
17
       \def\empty{}%
       \ifx\x\empty % LaTeX, first loading,
18
19
         % variable is initialized, but \ProvidesPackage not yet seen
20
21
         \expandafter\ifx\csname PackageInfo\endcsname\relax
22
           \def\x#1#2{%}
             \immediate\write-1{Package #1 Info: #2.}%
23
           }%
24
         \else
25
           26
27
         \x{epstopdf}{The package is already loaded}%
28
29
         \aftergroup\endinput
30
       \fi
     \fi
31
32 \endgroup%
Package identification:
33 \begingroup\catcode61\catcode48\catcode32=10\relax%
     \catcode13=5 % ^^M
35
     \endlinechar=13 %
     \catcode35=6 % #
36
     \catcode39=12 % '
37
     \catcode40=12 % (
38
     \catcode41=12 % )
39
     \catcode44=12 % ,
40
     \catcode45=12 % -
     \colored{catcode46=12 \%} .
     \catcode47=12 % /
43
     \catcode58=12 % :
44
     \catcode64=11 % @
45
     \catcode91=12 % [
46
     \catcode93=12 % ]
47
     \catcode123=1 % {
     \catcode125=2 % }
     \expandafter\ifx\csname ProvidesPackage\endcsname\relax
50
       \def\x#1#2#3[#4]{\endgroup}
51
         \immediate\write-1{Package: #3 #4}%
52
         \xdef#1{#4}%
53
54
       }%
55
     \else
       \def \x#1#2[#3] {\endgroup}
56
         #2[{#3}]%
57
         \ifx#1\@undefined
58
           \xdef#1{#3}%
59
         \fi
60
61
         \ifx#1\relax
           \xdef#1{#3}%
62
         \fi
63
       }%
64
    \fi
65
66 \expandafter\x\csname ver@epstopdf.sty\endcsname
67 \ProvidesPackage{epstopdf}%
     [2020-01-24 v2.11 Conversion with epstopdf on the fly (HO)]%
Larger catcode set because of configuration files needed.
69 \begingroup\catcode61\catcode48\catcode32=10\relax%
    \catcode13=5 % ^^M
     \endlinechar=13 %
```

```
\catcode123=1 % {
     \catcode125=2 % }
73
74
     \catcode64=11 % @
75
     \def\x{\endgroup
       \expandafter\edef\csname ETE@AtEnd\endcsname{%
76
77
         \endlinechar=\the\endlinechar\relax
         \catcode13=\the\catcode13\relax
78
         \catcode32=\the\catcode32\relax
79
         \catcode35=\the\catcode35\relax
80
81
         \catcode61=\the\catcode61\relax
         \catcode64=\the\catcode64\relax
83
         \catcode123=\the\catcode123\relax
         \catcode125=\the\catcode125\relax
84
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{%
     \edef\ETE@AtEnd{%
       \ETE@AtEnd
96
97
       \catcode#1=\the\catcode#1\relax
98
    }%
     \catcode#1=#2\relax
99
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*{%
    \expandafter\epstopdfsetup\expandafter{\CurrentOption}%
119 }
120 \ProcessOptions*\relax
121 \ETE@AtEnd%
122 (/package)
```

2.2 Base package

2.3 Preparations

2.3.1 Relead check and identification

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

```
124 \begingroup\catcode61\catcode48\catcode32=10\relax%
     \catcode13=5 % ^^M
     \endlinechar=13 %
126
     \catcode35=6 % #
127
     \catcode39=12 % '
128
     \catcode44=12 % ,
129
130
    \catcode45=12 % -
    \catcode46=12 % .
131
132
    \catcode58=12 % :
133
    \catcode64=11 % @
     \catcode123=1 % {
134
     \catcode125=2 % }
135
     \expandafter\let\expandafter\x\csname ver@epstopdf-base.sty\endcsname
136
137
     \ifx\x\relax % plain-TeX, first loading
138
139
       \def\empty{}%
       \ifx\x\empty % LaTeX, first loading,
140
         % variable is initialized, but \ProvidesPackage not yet seen
141
142
         \expandafter\ifx\csname PackageInfo\endcsname\relax
143
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}}%
         \fi
149
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%
     \catcode13=5 % ^^M
156
     \endlinechar=13 %
157
158
     \catcode35=6 % #
     \catcode39=12 % '
     \catcode40=12 % (
160
     \catcode41=12 % )
161
     \colone{1} \catcode44=12 % ,
162
     \catcode45=12 % -
163
    \catcode46=12 % .
164
165
    \catcode47=12 % /
    \catcode58=12 % :
166
     \catcode64=11 % @
167
     \catcode91=12 % [
168
     \catcode93=12 % ]
169
     \verb|\catcode123=1 % | \{
170
171
     \catcode125=2 % }
172
     \expandafter\ifx\csname ProvidesPackage\endcsname\relax
       \def\x#1#2#3[#4]{\endgroup
173
```

```
174
         \immediate\write-1{Package: #3 #4}%
         \xdef#1{#4}%
175
       }%
176
177
     \else
       \def\x#1#2[#3]{\endgroup
178
         #2[{#3}]%
179
         \ifx#1\@undefined
180
           \xdef#1{#3}%
181
         \fi
182
         \int x#1\relax
183
           \xdef#1{#3}%
184
         \fi
185
186
       }%
187
     \fi
188 \expandafter\x\csname ver@epstopdf-base.sty\endcsname
189 \ProvidesPackage{epstopdf-base}%
     [2020-01-24 v2.11 Base part for package epstopdf]%
```

2.3.2 Catcodes

```
191 \begingroup\catcode61\catcode48\catcode32=10\relax%
    \catcode13=5 % ^^M
192
     \endlinechar=13 %
193
    \catcode123=1 % {
194
195
     \catcode125=2 % }
196
     \catcode64=11 % @
197
     \def\x{\endgroup
       \expandafter\edef\csname ETE@AtEnd\endcsname{%
198
         \endlinechar=\the\endlinechar\relax
199
         \catcode13=\the\catcode13\relax
200
201
         \catcode32=\the\catcode32\relax
202
         \catcode35=\the\catcode35\relax
         \catcode61=\the\catcode61\relax
203
204
         \catcode64=\the\catcode64\relax
         \catcode123=\the\catcode123\relax
205
         \color=\the\color=125\relax
206
       }%
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{%
       \ETE@AtEnd
218
       \color=\the\color=1
219
    }%
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\\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 LATEX 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 LuaTeX, the Lua portion of pdftexcmds is used to provide the file handling functionality.

```
235 \ifx\@curroptions\@empty
     \ifx\@PackageInfo\@undefined
^{236}
       \let\@PackageInfo\PackageInfo
237
238
       \let\@PackageWarningNoLine\PackageWarningNoLine
239
       \def\@PackageInfoNoLine#1#2{\PackageInfo{#1}{#2\@gobble}}
       \let\@PackageError\PackageError
240
241
     \fi
     \ifx\pdf@strcmp\@undefined
242
243
       \ifx\directlua\@undefined
         \def\pdf@strcmp{\pdfstrcmp}
244
         \def\pdf@filemoddate{\pdffilemoddate}
245
         \def\pdf@filesize{\pdffilesize}
246
         \def\pdf@system#{\immediate\write18 }
247
248
         \directlua{require('pdftexcmds')}
249
         \long\def\pdf@strcmp#1#2{\directlua{%
250
251
           oberdiek.pdftexcmds.strcmp('\luaescapestring{#1}',
252
                                        '\luaescapestring{#2}')}}%
         \def\pdf@filemoddate#1{\directlua{%
253
           oberdiek.pdftexcmds.filemoddate('\luaescapestring{#1}')}}
254
         \def\pdf@filesize#1{\directlua{%
255
256
           oberdiek.pdftexcmds.filesize('\luaescapestring{#1}')}}
         \def\pdf@system#1{\directlua{%
257
           oberdiek.pdftexcmds.system('\luaescapestring{#1}')}}
258
       \fi
259
     \fi
260
261 \else
     \RequirePackage{infwarerr}[2007/09/09]
262
263
     \RequirePackage{grfext}\relax
     \RequirePackage{kvoptions}[2007/10/02]
264
     \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
     \def\x{pdftex.def}%
279
     \ifx\Gin@driver\x
280
281
     \else
       \@PackageWarningNoLine{epstopdf}{%
282
         Drivers other than 'pdftex' and 'luatex' are not supported%
283
284
       }%
285
       \endgroup
       \newcommand*{\epstopdfsetup}[1]{}%
286
       \expandafter\expandafter\ETE@AtEnd
287
     \fi%
288
     \pi
289
290 \endgroup
   Check, whether the shell escape feature is enabled.
291 \begingroup
     \expandafter\ifx\csname pdf@shellescape\endcsname\relax
292
293
     \else
       \ifnum\pdf@shellescape>0 %
294
295
       \else
         \@PackageWarningNoLine{epstopdf}{%
296
           Shell escape feature is not enabled%
297
         }%
298
299
       \fi
     \fi
300
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
     \def\ETE@let#1#2{%
304
       \expandafter\let\csname ifETE0#1\expandafter\endcsname
305
       \csname if#2\endcsname}
306
     \ETE@let{verbose}{true}
307
308
     \ETE@let{disable}{false}
309
     \ETE@let{update}{true}
     \ETE@let{prepend}{false}
310
     \ETE@let{prefersuffix}{true}
311
     \def\ETE@outdir{}
312
     \def\ETE@suffix{-\SourceExt-converted-to}
313
     \def\ETE@program@epstopdf{\epstopdf@sys@cmd}
314
     \newcommand*{\epstopdfsetup}[1]{}%
316 \ensuremath{\setminus} else
```

```
\SetupKeyvalOptions{family=ETE,prefix=ETE0}
317
     \DeclareBoolOption{update}
318
319
     \DeclareBoolOption{verbose}
320
     \DeclareVoidOption{prepend}{\ETE@prependtrue}
     \DeclareVoidOption{append}{\ETE@prependfalse}
321
     \DeclareStringOption{outdir}
322
     \DeclareStringOption{suffix}
323
     \DeclareBoolOption{prefersuffix}
324
     \DeclareStringOption{program@epstopdf}
325
```

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

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

2.6 Make and verbose

```
339 \verb|\begingroup| expandafter \verb|\expandafter| expandafter \verb|\expandafter| expandafter \verb|\expandafter| expandafter \verb|\expandafter| expandafter expan
340 \end{ter\ifx} csname pdf@filemoddate\endcsname\relax
                   \def\ETE@Make#1#2{%
341
                          \ifETE@update
342
                                  \ETE@WarnModDate
343
                          \fi
344
                          \@firstofone
345
                  }%
346
                   \def\ETE@WarnModDate{%
347
                          \@PackageWarningNoLine{epstopdf}{%
348
                                  \string\pdffilemoddate\space is not available,\MessageBreak
349
350
                                  option 'update' will be ignored%
351
                          \global\let\ETE@WarnModDate\relax
352
                  }%
353
                   \def\ETE@FileInfo#1#2{#1 file: <#2>}%
354
355 \ensuremath{\setminus} else
                   \def\ETE@Make#1#2{%
356
                          \ifETE@update
357
                                  \ifnum\pdf@strcmp{\pdf@filemoddate{#1}}{\pdf@filemoddate{#2}}>0 %
358
                                          \expandafter\expandafter\expandafter\Ofirstofone
359
                                  \else
360
361
                                          \@PackageInfoNoLine{epstopdf}{%
                                                 Output file is already uptodate%
362
363
                                          \expandafter\expandafter\@gobble
364
                                  \fi
365
```

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

2.7 Adding conversion support

366

\else

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

```
399 \expandafter\ifx\csname ETE@OrgGin@setfile\endcsname\relax
     \let\ETE@OrgGin@setfile\Gin@setfile
400
401 \else
402
     \@PackageError{epstopdf}{%
       Command \string\ETE@OrgGin@setfile\space
403
       already defined.\MessageBreak
404
     }{%
405
       Probably some package has included the code of this package%
406
407
       \MessageBreak
408
       instead of using \string\RequirePackage{epstopdf}.%
       \MessageBreak
409
       \@ehc
410
    }%
411
412 \fi
413 \def\ETE@IfFileExists{%
414
     \begingroup\expandafter\expandafter\expandafter\endgroup
415
     \expandafter\ifx\csname grffile@IfFileExists\endcsname\relax
       \expandafter\IfFileExists
416
     \else
417
       \global\let\ETE@IfFileExists\grffile@IfFileExists
418
       \expandafter\grffile@IfFileExists
419
420
     \fi
```

```
421 }
423
     \fi
424
     \fi
425
     \endgroup
426
     \fi
     \fi
427
     #1%
428
429 }
430 \neq 30 
431 \newcommand*{\epstopdfcall}[1]{%
     \ifETE@InsideSetfile
432
433
       \expandafter\@firstoftwo
434
     \else
       \expandafter\@secondoftwo
435
436
     \fi
     { '#1}%
437
438
     {\Gin@base\Gin@ext}%
439 }
440 \def\ETE@DefCommandLine#1{%
     \edef\CommandLine{\expandafter\fi\if'#1}%
441
442 }
443 \det ETE@DefX#1{%}
     \expandafter\expandafter\def
     \expandafter\expandafter\x
445
446
     \expandafter\expandafter\expandafter{%
447
       \expandafter\fi\if'#1\relax\else
     }%
448
449 }
450 \ensuremath{ \mbox{ \mbox{def}\mbox{ETE@Gin@setfile}#1#2#3{\mbox{\%}}}
451
     \ifETE@disable
       \ETE@OrgGin@setfile{#1}{#2}{#3}%
452
     \else
453
       \begingroup
454
         \ETE@InsideSetfiletrue
455
         \ETE@DefX{#3}%
456
457
       \expandafter\endgroup
       \ifx\x\@empty
458
459
         \ETE@OrgGin@setfile{#1}{#2}{#3}%
460
       \else
461
         \begingroup
           \ETE@InsideSetfiletrue
462
           \def\GraphicsType{#1}%
463
464
           \def\GraphicsRead{#2}%
            \ifETE@prefersuffix
465
           \else
466
             \ifx\ETE@suffix\@empty
467
             \else
468
                \ETE@IfFileExists{\Gin@base\GraphicsRead}{%
469
470
                  \ETE@Skip{%
                    \ETE@OrgGin@setfile{#1}{#2}{\Gin@base#2}%
471
472
                  }%
473
                }{%
474
                  \left( \cdot \right) 
                }%
475
476
                \next
477
             \fi
478
           \fi
```

```
479
            \ifx\Gin@ext\relax
              \let\SourceExt\Gin@eext
480
481
              \def\SourceFile{\Gin@base\Gin@eext}%
            \else
              \let\SourceExt\Gin@ext
483
              \def\SourceFile{\Gin@base\Gin@ext}%
484
            \fi
485
            \edef\SourceExt{% remove dot
486
              \expandafter\@cdr\SourceExt\@empty\@nil
487
           }%
488
            \let\OutputDirectory\ETE@outdir
489
            \ifx\OutputDirectory\@empty
490
              \edef\OutputFile{\ETE@GenerateName{\Gin@base}{#2}}%
491
            \else
492
              \begingroup
493
494
                \filename@parse{\Gin@base#2}%
495
                \edef\x{\endgroup
                  \def\noexpand\OutputFile{%
496
                    \ETE@GenerateName{%
497
                      \OutputDirectory\filename@base
498
                    }{#2}%
499
                  }%
500
                }%
501
             \x
502
            \fi
503
            \ETE@DefCommandLine{#3}%
504
            \ifETE@verbose
505
              \@PackageInfo{epstopdf}{%
506
507
                \ETE@FileInfo{Source}\SourceFile\MessageBreak
508
                \ETE@FileInfo{Output}\OutputFile\MessageBreak
                Command: <\CommandLine>\MessageBreak
509
                \string\includegraphics
510
             }%
511
            \fi
512
            \ETE@Make\SourceFile\OutputFile{%
513
              \pdf@system{\CommandLine}%
514
              \ifETE@verbose
515
                \@PackageInfoNoLine{epstopdf}{%
516
517
                  \ETE@FileInfo{Result}\OutputFile
                }%
518
             \fi
519
           }%
520
521
            \edef\x{\endgroup
522
              \ifx\OutputDirectory\@empty
523
              \else
                \def\noexpand\Gin@base{%
524
                  \OutputDirectory\noexpand\filename@base
525
                }%
526
              \fi
527
              \ifx\ETE@suffix\@empty
528
529
                \edef\noexpand\Gin@base{%
530
                  \noexpand\Gin@base\ETE@suffix
531
532
                }%
              \fi
533
              \noexpand\ETE@OrgGin@setfile{%
534
535
                \GraphicsType
536
```

```
537
                 \GraphicsRead
              }{%
538
539
                 \OutputFile
              }%
540
541
            }%
542
          \x
        \fi
543
     \fi
544
545 }
546 \let\Gin@setfile\ETE@Gin@setfile
547 \def\ETE@GenerateName#1#2{%
     #1\ETE@suffix#2%
549 }
```

2.8 Declare graphics rule

```
550 \newcommand*{\epstopdfDeclareGraphicsRule}[4]{\%
                               \int \frac{\pi}{\pi} \frac{4}{\pi}
551
                                           \@PackageError{epstopdf-base}{%
552
553
                                                      Conversion command is missing%
554
                                          }\@ehc
                               \else
555
                                           \begingroup
556
                                                       \@ifundefined{Gin@rule@#1}{%
557
558
                                                                    \@PackageInfo{epstopdf-base}{%
559
                                                                             Redefining graphics rule for '#1'%
560
561
                                                       }%
562
                                           \endgroup
563
                                           \label{lem:condition} $$\operatorname{Gin@rule@#1}$#1{$\#2}{\#3}{\operatorname{lopstopdfcall}$\#4}}\
564
                               \fi
565
566 }
                     \DeclareGraphicsRule for .eps
567 \verb|\epstopdfDeclareGraphicsRule{.eps}{pdf}{..pdf}{%} in the constant of t
                               \verb|\ETE@epstopdf{#1}||
568
569 }
570 \def\ETE@epstopdf#1{%
                               \verb|\ETE@program@epstopdf\space| \\
571
572
                               \ifcase\ifx\OutputDirectory\@empty
573
                                                                                     \ifx\ETE@suffix\@empty
                                                                                                1%
574
                                                                                     \fi
575
                                                                         \fi
576
                                                                        0 %
577
                                           --outfile=\OutputFile\space
578
579
                               \fi
580
                               #1%
581 }
582 \verb|\frac{\alpha}{\alpha} end Graphics Extensions \verb|\color="end-of-color-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-order-o
583 \ifETE@prepend
584
                              \expandafter\PrependGraphicsExtensions
585 \setminus else
                              \expandafter\AppendGraphicsExtensions
586
587\fi
588 {.eps}
589 \fi
```

```
590 \ \text{let}\ \text{ETEOprepend}\ \text{Oundefined}
591\ \ifx\SetupKeyvalOptions\@undefined
592 \else
593 \DeclareVoidOption{prepend}{%
    \PrependGraphicsExtensions{.eps}%
595 }
596 \let\ETE@append\@undefined
597 \DeclareVoidOption{append}{%
    \AppendGraphicsExtensions{.eps}%
599 }
600 \fi
601 \InputIfFileExists{epstopdf-sys.cfg}{}{}
602 \InputIfFileExists{epstopdf.cfg}{}{}
   Use epstpdf if the cfg files have not set a default.
603 \expandafter\ifx\csname epstopdf@sys@cmd\endcsname\relax
    \def\epstopdf@sys@cmd{epstopdf}
605 \fi
606 \ETE@AtEnd%
607 (/base)
```

3 Installation

3.1 Download

Package. This package is available on CTAN¹:

CTAN:macros/latex/contrib/epstopdf-pkg/epstopdf.dtx The source file.

 ${\tt CTAN:macros/latex/contrib/epstopdf-pkg/epstopdf.pdf\ Documentation.}$

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.

3.2 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
```

3.3 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
```

¹CTAN: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):

```
\begin{array}{lll} \texttt{epstopdf.sty} & \to \texttt{tex/latex/epstopdf/epstopdf.sty} \\ \texttt{epstopdf-base.sty} & \to \texttt{tex/latex/epstopdf/epstopdf-base.sty} \\ \texttt{epstopdf.pdf} & \to \texttt{doc/latex/epstopdf/epstopdf.pdf} \\ \texttt{epstopdf.dtx} & \to \texttt{source/latex/epstopdf/epstopdf.dtx} \end{array}
```

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.

3.4 Refresh file name databases

If your T_EX distribution (T_EX Live, MiKT_EX, ...) relies on file name databases, you must refresh these. For example, T_EX Live users run texhash or mktexlsr.

3.5 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 pdfLATEX:

```
pdflatex epstopdf.dtx
makeindex -s gind.ist epstopdf.idx
pdflatex epstopdf.dtx
makeindex -s gind.ist epstopdf.idx
pdflatex epstopdf.dtx
```

4 History

```
[2001/01/06 v1.0]
```

• First public version, published in the pdfTEX mailing list.

[2001/02/04 v1.1]

- Minor documentation update.
- CTAN.

[2006/02/20 v1.2]

- DTX framework.
- Compatibility for miniltx.tex.

[2006/08/26 v1.3]

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

[2007/04/26 v1.4]

• Documentation rewritten and extended.

[2007/10/02 v1.5]

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

[2007/11/11 v1.6]

• Use of package pdftexcmds for LuaTFX 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: \end occurred when \ifx on line 165 was incomplete warning in dvi mode.

[2019-11-30 v2.9]

• Read the <code>epstopdf.cfg</code> file even in the base code is being used.

[2020-01-24 v2.11]

 \bullet use ' not " for Lua strings, as the latter not made safe.

[2020-01-24 v2.11]

• Ensure that \epstopdf@sys@cmd is defined even if no cfg file is found.

5 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	192, 194, 195, 196, 200, 201,
\@PackageError 240, 402, 552	202, 203, 204, 205, 206, 209,
\@PackageInfo 236, 237, 506, 559	210, 212, 213, 214, 215, 219, 221
\@PackageInfoNoLine 239, 361, 516	\CommandLine 441, 509, 514
\@PackageWarningNoLine	\csname
238, 269, 282, 296, 348	76, 136, 143, 172, 188, 198, 268,
\@cdr	292, 305, 306, 340, 399, 415, 603
\@curroptions	\CurrentOption 118
\@ehc 410, 554	D
\@empty 235, 458,	\DeclareBoolOption . 318, 319, 324, 326
467, 487, 490, 522, 528, 572, 573	\DeclareComplementaryOption 327
\@firstofone 345, 359, 367 \@firstoftwo 433	\DeclareOption
\@gobble	\DeclareStringOption 322, 323, 325
\@ifundefined	\DeclareVoidOption . 320, 321, 593, 597
\@namedef	\directlua 243, 249, 250, 253, 255, 257
\@nil . 373, 375, 377, 380, 388, 391, 487	,,,,,,
\@secondoftwo	${f E}$
\@spaces	\empty 17, 18, 139, 140
\Qundefined 58, 180, 236,	\endcsname 14, 21, 50, 66,
242, 243, 303, 582, 590, 591, 596	76, 136, 143, 172, 188, 198, 268,
\\	292, 305, 306, 340, 399, 415, 603
	\endinput 29, 109, 151, 234
${f A}$	\endlinechar $4, 35,$
\aftergroup 29, 151	71, 77, 89, 126, 157, 193, 199, 211
\AppendGraphicsExtensions	\epstopdf@sys@cmd 314, 604
582, 586, 598	\epstopdfcall 431, 564
	\epstopdfDeclareGraphicsRule 550, 567
\mathbf{C}	\epstopdfsetup
\catcode $2, 3, 5, 6, 7, 8, 9, 10,$	\dots 4, 118, 272, 286, 315, 328, 329
11, 12, 13, 33, 34, 36, 37, 38, 39,	\ETE@@Date 380, 383
40, 41, 42, 43, 44, 45, 46, 47, 48,	\ETE@@Time 386, 388
49, 69, 70, 72, 73, 74, 78, 79, 80,	\ETE@append 596
81, 82, 83, 84, 87, 88, 90, 91, 92,	\ETE@AtEnd 95, 96, 109, 110, 116,
93, 97, 99, 124, 125, 127, 128,	121, 217, 218, 234, 273, 287, 606
129, 130, 131, 132, 133, 134,	\ETE@Date 373, 377
129, 130, 131, 132, 133, 134, 135, 155, 156, 158, 159, 160,	\ETE@Date
129, 130, 131, 132, 133, 134,	\ETE@Date 373, 377

\ETE@E:10Info 254 270 507 509 517	${f M}$
\ETEGFileInfo . 354, 370, 507, 508, 517 \ETEGGenerateName 491, 497, 547	\MessageBreak 349, 384,
\ETE@Gin@setfile 450, 546	394, 404, 407, 409, 507, 508, 509
\ETE@IfFileExists 413, 418, 469	334, 404, 407, 403, 307, 300, 303
\ETE@InsideSetfiletrue 455, 462	${f N}$
\ETE@let 304, 307, 308, 309, 310, 311	\newcommand 272, 286, 315, 328, 431, 550
\ETE@Make	\newif 302, 430
\ETE@OrgGin@setfile	\next 474, 476
400, 403, 452, 459, 471, 534	
\ETE@outdir 312, 489	О
\ETE@prepend 590	\OutputDirectory
\ETE@prependfalse	489, 490, 498, 522, 525, 572
\ETE@prependtrue 320	\OutputFile
\ETE@program@epstopdf 314, 571	. 491, 496, 508, 513, 517, 539, 578
\ETE@SavedAtEnd 110, 116	P
\ETE@Size 375, 391	\PackageError 240
\ETE@Skip 422, 470	\PackageInfo 26, 148, 237, 239
\ETE@suffix 313, 467, 528, 531, 548, 573	\PackageWarningNoLine 238
\ETE@WarnModDate 343, 347, 352	\pdf@filemoddate 245, 253, 358, 373
	\pdf@filesize 246, 255, 375
${f F}$	\pdf@shellescape 294
\filename@base 498, 525	\pdf@strcmp 242, 244, 250, 358
\filename@parse 494	\pdf@system 247, 257, 514
	\pdffilemoddate 245, 349
G	\pdffilesize 246
\Gin@base 438, 469, 471,	\pdfstrcmp 244
481, 484, 491, 494, 524, 530, 531	\PrependGraphicsExtensions . 584, 594
\Gin@driver	\ProcessOptions
\Gin@eext	\ProvidesPackage 19, 67, 141, 189
\Gin@setfile 438, 479, 483, 484	B.
\GraphicsRead 464, 469, 537	\RequirePackage 111, 112, 113,
\GraphicsType 463, 535	114, 115, 262, 263, 264, 265, 408
\grffile@IfFileExists 418, 419	, -, - ,, - ,,
,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	${f S}$
I	\setkeys 328
\if 441, 447	\SetupKeyvalOptions 303, 317, 591
\ifcase 572	\SourceExt 313, 335, 480, 483, 486, 487
\ifETE@disable 451	\SourceFile 481, 484, 507, 513
\ifETE@InsideSetfile 430, 432	\space 349, 385, 395, 403, 571, 578
\ifETE@prefersuffix 465	${f T}$
\ifETE@prepend 302, 583	\the 77, 78, 79,
\ifETE@update 342, 357	80, 81, 82, 83, 84, 97, 199, 200,
\ifETE@verbose 505, 515	201, 202, 203, 204, 205, 206, 219
\IfFileExists	\TMP@EnsureCode 94, 101,
\ifnum 294, 358	102, 103, 104, 105, 106, 107,
\ifx 15, 18, 21, 50, 58, 61, 137,	108, 216, 223, 224, 225, 226,
140, 143, 172, 180, 183, 235,	227, 228, 229, 230, 231, 232, 233
236, 242, 243, 268, 277, 280,	
292, 303, 340, 378, 392, 399,	W
415, 458, 467, 479, 490, 522, 528, 551, 572, 573, 582, 501, 603	\write 23, 52, 145, 174, 247
528, 551, 572, 573, 582, 591, 603	X
\immediate 23, 52, 145, 174, 247 \includegraphics 510	\x 14, 15, 18, 22, 26, 28,
\InputIfFileExists 601, 602	51, 56, 66, 75, 87, 136, 137, 140,
(Impastit IIODAID 00	
${f L}$	144, 148, 150, 173, 178, 188,
L \luaescapestring 251, 252, 254, 256, 258	