

# The epstopdf package

Heiko Oberdiek  
<heiko.oberdiek at gmail.com>

2010/02/09 v2.5

## 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

<b>1</b>	<b>Documentation</b>	<b>2</b>
1.1	Introduction	2
1.2	Requirements	2
1.3	Usage	3
1.4	Options	3
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	5
<b>2</b>	<b>Implementation</b>	<b>6</b>
2.1	Wrapper package	6
2.1.1	Option handling	8
2.2	Base package	8
2.3	Preparations	8
2.3.1	Relead check and identification	8
2.3.2	Catcodes	9
2.3.3	Load packages	10
2.4	Checks	10
2.5	Package loading	10
2.6	Options	11
2.6.1	Default setting	11
2.7	Make and verbose	11
2.8	Adding conversion support	12
2.9	Declare graphics rule	14
<b>3</b>	<b>Test</b>	<b>15</b>
3.1	Preface for standard catcode check	15
3.2	Catcode checks for loading	16
<b>4</b>	<b>Installation</b>	<b>17</b>
4.1	Download	17
4.2	Bundle installation	18
4.3	Package installation	18
4.4	Refresh file name databases	18
4.5	Some details for the interested	18

<b>5 History</b>	<b>19</b>
[2001/01/06 v1.0]	19
[2001/02/04 v1.1]	19
[2006/02/20 v1.2]	19
[2006/08/26 v1.3]	19
[2007/04/26 v1.4]	19
[2007/10/02 v1.5]	19
[2007/11/11 v1.6]	19
[2008/05/06 v1.7]	20
[2009/03/01 v1.8]	20
[2009/07/06 v1.9]	20
[2009/07/07 v1.10]	20
[2009/07/12 v2.0]	20
[2009/07/15 v2.1]	20
[2009/07/16 v2.2]	20
[2009/09/24 v2.3]	20
[2009/10/17 v2.4]	20
[2010/02/09 v2.5]	20
<b>6 Index</b>	<b>21</b>

# 1 Documentation

## 1.1 Introduction

L<sup>A</sup>T<sub>E</sub>X 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<sub>T</sub>E<sub>X</sub> 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<sub>T</sub>E<sub>X</sub>. 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 T<sub>E</sub>X's compile run. Keep in mind that this is a security risk. The feature is an addition to `\TeX`. Mik<sub>T</sub>E<sub>X</sub>, te<sub>T</sub>E<sub>X</sub>, T<sub>E</sub>X Live support it. In Web2C based T<sub>E</sub>X distributions (te<sub>T</sub>E<sub>X</sub>, T<sub>E</sub>X 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:

```
\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 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 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 {⟨key value list⟩}`

L<sup>A</sup>T<sub>E</sub>X 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<sup>A</sup>T<sub>E</sub>X'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 `prefersuffix` is turned on, the file `foo-suffix.pdf` and its generation is preferred over using `foo.pdf`. Otherwise `foo.pdf` is included without generating `foo-suffix.pdf`. The default of option `prefersuffix` is `true`.

**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 -i .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 [2010/02/09 v2.5 Conversion with epstopdf on the fly (HO)]%

```

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}
110 \let\ETE@SavedAtEnd\ETE@AtEnd
111 \RequirePackage{epstopdf-base}[2010/02/09]
112 \let\ETE@AtEnd\ETE@SavedAtEnd

```

### 2.1.1 Option handling

```

113 \DeclareOption*{%
114   \expandafter\epstopdfsetup\expandafter{\CurrentOption}%
115 }
116 \ProcessOptions*\relax
117 \ETE@AtEnd%
118 \endpackage

```

## 2.2 Base package

```

119 (*base)

```

## 2.3 Preparations

### 2.3.1 Reload check and identification

Reload check, especially if the package is not used with L<sup>A</sup>T<sub>E</sub>X.

```

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

```

Package identification:

```

151 \begingroup\catcode61\catcode48\catcode32=10\relax%
152   \catcode13=5 % ^M
153   \endlinechar=13 %
154   \catcode35=6 % #
155   \catcode39=12 % '
156   \catcode40=12 % (
157   \catcode41=12 % )

```

```

158 \catcode44=12 % ,
159 \catcode45=12 % -
160 \catcode46=12 % .
161 \catcode47=12 % /
162 \catcode58=12 % :
163 \catcode64=11 % @
164 \catcode91=12 % [
165 \catcode93=12 % ]
166 \catcode123=1 % {
167 \catcode125=2 % }
168 \expandafter\ifx\csname ProvidesPackage\endcsname\relax
169   \def\x#1#2#3[#4]{\endgroup
170     \immediate\write-1{Package: #3 #4}%
171     \xdef#1{#4}%
172   }%
173 \else
174   \def\x#1#2[#3]{\endgroup
175     #2[#{#3}]%
176     \ifx#1@undefined
177       \xdef#1{#3}%
178     \fi
179     \ifx#1\relax
180       \xdef#1{#3}%
181     \fi
182   }%
183 \fi
184 \expandafter\x\csname ver@epstopdf-base.sty\endcsname
185 \ProvidesPackage{epstopdf-base}%
186 [2010/02/09 v2.5 Base part for package epstopdf]%

```

### 2.3.2 Catcodes

```

187 \begingroup\catcode61\catcode48\catcode32=10\relax%
188 \catcode13=5 % ^^M
189 \endlinechar=13 %
190 \catcode123=1 % {
191 \catcode125=2 % }
192 \catcode64=11 % @
193 \def\x{\endgroup
194   \expandafter\edef\csname ETE@AtEnd\endcsname{%
195     \endlinechar=\the\endlinechar\relax
196     \catcode13=\the\catcode13\relax
197     \catcode32=\the\catcode32\relax
198     \catcode35=\the\catcode35\relax
199     \catcode61=\the\catcode61\relax
200     \catcode64=\the\catcode64\relax
201     \catcode123=\the\catcode123\relax
202     \catcode125=\the\catcode125\relax
203   }%
204 }%
205 \x\catcode61\catcode48\catcode32=10\relax%
206 \catcode13=5 % ^^M
207 \endlinechar=13 %
208 \catcode35=6 % #
209 \catcode64=11 % @
210 \catcode123=1 % {
211 \catcode125=2 % }
212 \def\TMP@EnsureCode#1#2{%
213   \edef\ETE@AtEnd{%
214     \ETE@AtEnd
215     \catcode#1=\the\catcode#1\relax
216   }%
217   \catcode#1=#2\relax

```

```

218 }
219 \TMP@EnsureCode{33}{12}% !
220 \TMP@EnsureCode{39}{12}% '
221 \TMP@EnsureCode{42}{12}% *
222 \TMP@EnsureCode{44}{12}% ,
223 \TMP@EnsureCode{45}{12}% -
224 \TMP@EnsureCode{46}{12}% .
225 \TMP@EnsureCode{47}{12}% /
226 \TMP@EnsureCode{58}{12}% :
227 \TMP@EnsureCode{60}{12}% <
228 \TMP@EnsureCode{62}{12}% >
229 \TMP@EnsureCode{96}{12}% '
230 \edef\ETE@AtEnd{\ETE@AtEnd\noexpand\endinput}

```

### 2.3.3 Load packages

```

231 \RequirePackage{infwarerr}[2007/09/09]
232 \RequirePackage{grfext}\relax
233 \RequirePackage{kvoptions}[2007/10/02]
234 \RequirePackage{pdftexcmds}[2007/11/11]

```

## 2.4 Checks

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

```

235 \begingroup\expandafter\expandafter\expandafter\endgroup
236 \expandafter\ifx\csname Gin@setfile\endcsname\relax
237 \@PackageWarningNoLine{epstopdf}{%
238   No graphics package \string'graphic{s,x}\string' loaded%
239 }%
240 \newcommand*{\epstopdfsetup}[1]{%
241 \expandafter\ETE@AtEnd
242 \fi%

```

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

```

243 \begingroup
244 \def\x{pdftex.def}%
245 \ifx\Gin@driver\x
246 \else
247 \@PackageWarningNoLine{epstopdf}{%
248   Drivers other than 'pdftex.def' are not supported%
249 }%
250 \endgroup
251 \newcommand*{\epstopdfsetup}[1]{%
252 \expandafter\ETE@AtEnd
253 \fi%
254 \endgroup

```

Check, whether the shell escape feature is enabled.

```

255 \begingroup
256 \expandafter\ifx\csname pdf@shellescape\endcsname\relax
257 \else
258 \ifnum\pdf@shellescape>0 %
259 \else
260 \@PackageWarningNoLine{epstopdf}{%
261   Shell escape feature is not enabled%
262 }%
263 \fi
264 \fi
265 \endgroup

```

## 2.5 Package loading

## 2.6 Options

```
266 \SetupKeyvalOptions{family=ETE,prefix=ETE@}
267 \DeclareBoolOption{update}
268 \DeclareBoolOption{verbose}
269 \newif\ifETE@prepend
270 \DeclareVoidOption{prepend}{\ETE@prependtrue}
271 \DeclareVoidOption{append}{\ETE@prependfalse}
272 \DeclareStringOption{outdir}
273 \DeclareStringOption{suffix}
274 \DeclareBoolOption{prefersuffix}
275 \DeclareStringOption{program@epstopdf}
```

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

```
276 \DeclareBoolOption{disable}
277 \DeclareComplementaryOption{enable}{disable}
278 \newcommand*{\epstopdfsetup}{\setkeys{ETE}}
```

### 2.6.1 Default setting

```
279 \epstopdfsetup{%
280   verbose,%
281   enable,%
282   append,%
283   update,%
284   prefersuffix,%
285   suffix=-\SourceExt-converted-to,%
286   program@epstopdf=epstopdf%
287 }
```

## 2.7 Make and verbose

```
288 \begingroup\expandafter\expandafter\expandafter\endgroup
289 \expandafter\ifx\csname pdf@filemoddate\endcsname\relax
290   \def\ETE@Make#1#2{%
291     \ifETE@update
292       \ETE@WarnModDate
293     \fi
294     \@firstofone
295   }%
296   \def\ETE@WarnModDate{%
297     \@PackageWarningNoLine{epstopdf}{%
298       \string\pdf@filemoddate\space is not available,\MessageBreak
299       option 'update' will be ignored%
300     }%
301     \global\let\ETE@WarnModDate\relax
302   }%
303   \def\ETE@FileInfo#1#2{#1 file: <#2>}%
304 \else
305   \def\ETE@Make#1#2{%
306     \ifETE@update
307       \ifnum\pdf@strcmp{\pdf@filemoddate{#1}}{\pdf@filemoddate{#2}}>0 %
308         \expandafter\expandafter\expandafter\@firstofone
309       \else
310         \@PackageInfoNoLine{epstopdf}{%
311           Output file is already uptodate%
312         }%
313         \expandafter\expandafter\expandafter\@gobble
314       \fi
315     \else
316       \expandafter\@firstofone
317     \fi
318   }%
319   \def\ETE@FileInfo#1#2{%
```

```

320 #1 file: <#2>%
321 \expandafter\expandafter\expandafter
322 \ETE@Date\pdf@filemoddate{#2}\@nil
323 \expandafter\expandafter\expandafter
324 \ETE@Size\pdf@filesize{#2}\@nil
325 }%
326 \def\ETE@Date#1\@nil{%
327 \ifx\#1\%
328 \else
329 \ETE@@Date#1\@nil
330 \fi
331 }%
332 \def\ETE@@Date#1:#2#3#4#5#6#7#8#9{%
333 \MessageBreak
334 \@spaces\space\space\space date: #2#3#4#5-#6#7-#8#9 %
335 \ETE@Time
336 }%
337 \def\ETE@Time#1#2#3#4#5#6#7\@nil{%
338 #1#2:#3#4:#5#6%
339 }%
340 \def\ETE@Size#1\@nil{%
341 \ifx\#1\%
342 \else
343 \MessageBreak
344 \@spaces\space\space\space size: #1 bytes%
345 \fi
346 }%
347 \fi

```

## 2.8 Adding conversion support

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

```

348 \expandafter\ifx\csname ETE@OrgGin@setfile\endcsname\relax
349 \let\ETE@OrgGin@setfile\Gin@setfile
350 \else
351 \@PackageError{epstopdf}{%
352 Command \string\ETE@OrgGin@setfile\space
353 already defined.\MessageBreak
354 }{%
355 Probably some package has included the code of this package%
356 \MessageBreak
357 instead of using \string\RequirePackage{epstopdf}.%
358 \MessageBreak
359 \@ehc
360 }%
361 \fi
362 \def\ETE@ifFileExists{%
363 \begingroup\expandafter\expandafter\expandafter\endgroup
364 \expandafter\ifx\csname grffile@ifFileExists\endcsname\relax
365 \expandafter\ifFileExists
366 \else
367 \global\let\ETE@ifFileExists\grffile@ifFileExists
368 \expandafter\grffile@ifFileExists
369 \fi
370 }
371 \def\ETE@Skip#1#2\x\fi\fi{%
372 \fi
373 \fi
374 \endgroup
375 \fi
376 \fi
377 #1%
378 }

```

```

379 \newif\ifETE@InsideSetfile
380 \newcommand*{\epstopdfcall}[1]{%
381   \ifETE@InsideSetfile
382     \expandafter\@firstoftwo
383   \else
384     \expandafter\@secondoftwo
385   \fi
386   {'#1}%
387   {\Gin@base\Gin@ext}%
388 }
389 \def\ETE@DefCommandLine#1{%
390   \edef\CommandLine{\expandafter\fi\if'#1}%
391 }
392 \def\ETE@DefX#1{%
393   \expandafter\expandafter\expandafter\def
394   \expandafter\expandafter\expandafter\x
395   \expandafter\expandafter\expandafter{%
396     \expandafter\fi\if'#1\relax\else
397   }%
398 }
399 \def\ETE@Gin@setfile#1#2#3{%
400   \ifETE@disable
401     \ETE@OrgGin@setfile{#1}{#2}{#3}%
402   \else
403     \begingroup
404       \ETE@InsideSetfiletrue
405       \ETE@DefX{#3}%
406     \expandafter\endgroup
407     \ifx\x\@empty
408       \ETE@OrgGin@setfile{#1}{#2}{#3}%
409     \else
410       \begingroup
411         \ETE@InsideSetfiletrue
412         \def\GraphicsType{#1}%
413         \def\GraphicsRead{#2}%
414         \ifETE@prefersuffix
415         \else
416           \ifx\ETE@suffix\@empty
417           \else
418             \ETE@IfFileExists{\Gin@base\GraphicsRead}{%
419               \ETE@Skip{%
420                 \ETE@OrgGin@setfile{#1}{#2}{\Gin@base#2}%
421               }%
422             }{%
423               \let\next\relax
424             }%
425           \next
426         \fi
427       \fi
428       \ifx\Gin@ext\relax
429         \let\SourceExt\Gin@eext
430         \def\SourceFile{\Gin@base\Gin@eext}%
431       \else
432         \let\SourceExt\Gin@ext
433         \def\SourceFile{\Gin@base\Gin@ext}%
434       \fi
435       \edef\SourceExt{% remove dot
436         \expandafter\@cdr\SourceExt\@empty\@nil
437       }%
438       \let\OutputDirectory\ETE@outdir
439       \ifx\OutputDirectory\@empty
440         \edef\OutputFile{\ETE@GenerateName{\Gin@base}{#2}}%

```

```

441     \else
442     \begingroup
443     \filename@parse{\Gin@base#2}%
444     \edef\x{\endgroup
445     \def\noexpand\OutputFile{%
446     \ETE@GenerateName{%
447     \OutputDirectory\filename@base
448     }{#2}%
449     }%
450     }%
451     \x
452     \fi
453     \ETE@DefCommandLine{#3}%
454     \ifETE@verbose
455     \@PackageInfo{epstopdf}{%
456     \ETE@FileInfo{Source}\SourceFile\MessageBreak
457     \ETE@FileInfo{Output}\OutputFile\MessageBreak
458     Command: <\CommandLine>\MessageBreak
459     \string\includegraphics
460     }%
461     \fi
462     \ETE@Make\SourceFile\OutputFile{%
463     \pdf@system{\CommandLine}%
464     \ifETE@verbose
465     \@PackageInfoNoLine{epstopdf}{%
466     \ETE@FileInfo{Result}\OutputFile
467     }%
468     \fi
469     }%
470     \edef\x{\endgroup
471     \ifx\OutputDirectory\@empty
472     \else
473     \def\noexpand\Gin@base{%
474     \OutputDirectory\noexpand\filename@base
475     }%
476     \fi
477     \ifx\ETE@suffix\@empty
478     \else
479     \edef\noexpand\Gin@base{%
480     \noexpand\Gin@base\ETE@suffix
481     }%
482     \fi
483     \noexpand\ETE@OrgGin@setfile{%
484     \GraphicsType
485     }{%
486     \GraphicsRead
487     }{%
488     \OutputFile
489     }%
490     }%
491     \x
492     \fi
493     \fi
494 }
495 \let\Gin@setfile\ETE@Gin@setfile
496 \def\ETE@GenerateName#1#2{%
497 #1\ETE@suffix#2%
498 }

```

## 2.9 Declare graphics rule

```

499 \newcommand*{\epstopdfDeclareGraphicsRule}[4]{%

```

```

500 \ifx\#4\%
501   \@PackageError{epstopdf-base}{%
502     Conversion command is missing%
503   }\@ehc
504 \else
505   \begingroup
506     \@ifundefined{Gin@rule@#1}{%
507     }{%
508       \@PackageInfo{epstopdf-base}{%
509         Redefining graphics rule for '#1'%
510       }%
511     }%
512   \endgroup
513   \@namedef{Gin@rule@#1}##1{#{2}-#{3}{\epstopdfcall{#4}}}%
514 \fi
515 }

\DeclareGraphicsRule for .eps
516 \epstopdfDeclareGraphicsRule{.eps}{pdf}{.pdf}{%
517   \ETE@epstopdf{#1}%
518 }
519 \def\ETE@epstopdf#1{%
520   \ETE@program@epstopdf\space
521   \ifcase\ifx\OutputDirectory\@empty
522     \ifx\ETE@suffix\@empty
523       1%
524     \fi
525     \fi
526     0 %
527   --outfile=\OutputFile\space
528   \fi
529   #1%
530 }

531 \ifETE@prepend
532   \expandafter\PrependGraphicsExtensions
533 \else
534   \expandafter\AppendGraphicsExtensions
535 \fi
536 {.eps}
537 \let\ETE@prepend\@undefined
538 \DeclareVoidOption{prepend}{%
539   \PrependGraphicsExtensions{.eps}%
540 }
541 \let\ETE@append\@undefined
542 \DeclareVoidOption{append}{%
543   \AppendGraphicsExtensions{.eps}%
544 }

545 \InputIfFileExists{epstopdf-sys.cfg}{}{}
546 \InputIfFileExists{epstopdf.cfg}{}{}

547 \ETE@AtEnd%
548 </base>

```

## 3 Test

### 3.1 Preface for standard catcode check

```

549 <*test1>
550 \input miniltx.tex\relax
551 \def\Gin@driver{pdftex.def}
552 \input graphicx.sty\relax
553 \resetatcatcode
554 </test1>

```

## 3.2 Catcode checks for loading

```
555 (*test1)
556 \catcode'\{=1 %
557 \catcode'\}=2 %
558 \catcode'\#=6 %
559 \catcode'\@=11 %
560 \expandafter\ifx\csname count@\endcsname\relax
561 \countdef\count@=255 %
562 \fi
563 \expandafter\ifx\csname @gobble\endcsname\relax
564 \long\def\@gobble#1{}%
565 \fi
566 \expandafter\ifx\csname @firstofone\endcsname\relax
567 \long\def\@firstofone#1{#1}%
568 \fi
569 \expandafter\ifx\csname loop\endcsname\relax
570 \expandafter\@firstofone
571 \else
572 \expandafter\@gobble
573 \fi
574 {%
575 \def\loop#1\repeat{%
576 \def\body{#1}%
577 \iterate
578 }%
579 \def\iterate{%
580 \body
581 \let\next\iterate
582 \else
583 \let\next\relax
584 \fi
585 \next
586 }%
587 \let\repeat=\fi
588 }%
589 \def\RestoreCatcodes{}
590 \count@=0 %
591 \loop
592 \edef\RestoreCatcodes{%
593 \RestoreCatcodes
594 \catcode\the\count@=\the\catcode\count@\relax
595 }%
596 \ifnum\count@<255 %
597 \advance\count@ 1 %
598 \repeat
599
600 \def\RangeCatcodeInvalid#1#2{%
601 \count@=#1\relax
602 \loop
603 \catcode\count@=15 %
604 \ifnum\count@<#2\relax
605 \advance\count@ 1 %
606 \repeat
607 }
608 \def\RangeCatcodeCheck#1#2#3{%
609 \count@=#1\relax
610 \loop
611 \ifnum#3=\catcode\count@
612 \else
613 \errmessage{%
614 Character \the\count@\space
615 with wrong catcode \the\catcode\count@\space
```

```

616         instead of \number#3%
617     }%
618     \fi
619     \ifnum\count@<#2\relax
620         \advance\count@ 1 %
621     \repeat
622 }
623 \def\space{ }
624 \expandafter\ifx\csname LoadCommand\endcsname\relax
625     \def\LoadCommand{\input epstopdf.sty\relax}%
626 \fi
627 \def\Test{%
628     \RangeCatcodeInvalid{0}{47}%
629     \RangeCatcodeInvalid{58}{64}%
630     \RangeCatcodeInvalid{91}{96}%
631     \RangeCatcodeInvalid{123}{255}%
632     \catcode'\@=12 %
633     \catcode'\=0 %
634     \catcode'\%=14 %
635     \LoadCommand
636     \RangeCatcodeCheck{0}{36}{15}%
637     \RangeCatcodeCheck{37}{37}{14}%
638     \RangeCatcodeCheck{38}{47}{15}%
639     \RangeCatcodeCheck{48}{57}{12}%
640     \RangeCatcodeCheck{58}{63}{15}%
641     \RangeCatcodeCheck{64}{64}{12}%
642     \RangeCatcodeCheck{65}{90}{11}%
643     \RangeCatcodeCheck{91}{91}{15}%
644     \RangeCatcodeCheck{92}{92}{0}%
645     \RangeCatcodeCheck{93}{96}{15}%
646     \RangeCatcodeCheck{97}{122}{11}%
647     \RangeCatcodeCheck{123}{255}{15}%
648     \RestoreCatcodes
649 }
650 \Test
651 \csname @@end\endcsname
652 \end
653 </test1>

```

## 4 Installation

### 4.1 Download

**Package.** This package is available on CTAN<sup>1</sup>:

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

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

**Bundle.** All the packages of the bundle ‘oberdiek’ are also available in a TDS compliant ZIP archive. There the packages are already unpacked and the documentation files are generated. The files and directories obey the TDS standard.

[CTAN:install/macros/latex/contrib/oberdiek.tds.zip](#)

*TDS* refers to the standard “A Directory Structure for T<sub>E</sub>X Files” ([CTAN:tds/tds.pdf](#)). Directories with `texmf` in their name are usually organized this way.

<sup>1</sup><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):

```
epstopdf.sty          → tex/latex/oberdiek/epstopdf.sty
epstopdf-base.sty    → tex/latex/oberdiek/epstopdf-base.sty
epstopdf.pdf          → doc/latex/oberdiek/epstopdf.pdf
test/epstopdf-test1.tex → doc/latex/oberdiek/test/epstopdf-test1.tex
epstopdf.dtx         → source/latex/oberdiek/epstopdf.dtx
```

If you have a `docstrip.cfg` that configures and enables `docstrip`'s TDS installing feature, then some files can already be in the right place, see the documentation of `docstrip`.

## 4.4 Refresh file name databases

If your  $\TeX$  distribution (`te $\TeX$` , `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<sup>A</sup>T<sub>E</sub>X:

```
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<sub>E</sub>X 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 LuaT<sub>E</sub>X 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`.

**[2010/02/09 v2.5]**

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

## 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	
\#	558
\%	634
\@	559, 632
\@PackageError	351, 501
\@PackageInfo	455, 508
\@PackageInfoNoLine	310, 465
\@PackageWarningNoLine	237, 247, 260, 297
\@cdr	436
\@ehc	359, 503
\@empty	407,
	416, 436, 439, 471, 477, 521, 522
\@firstofone	294, 308, 316, 567, 570
\@firstoftwo	382
\@gobble	313, 564, 572
\@ifundefined	506
\@namedef	513
\@nil	322, 324, 326, 329, 337, 340, 436
\@secondoftwo	384
\@spaces	334, 344
\@undefined	58, 176, 537, 541
\\	327, 341, 500, 633
\{	556
\}	557
<b>A</b>	
\advance	597, 605, 620
\aftergroup	29, 147
\AppendGraphicsExtensions	534, 543
<b>B</b>	
\body	576, 580
<b>C</b>	
\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, 120, 121, 123,
	124, 125, 126, 127, 128, 129,
	130, 131, 151, 152, 154, 155,
	156, 157, 158, 159, 160, 161,
	162, 163, 164, 165, 166, 167,
	187, 188, 190, 191, 192, 196,
	197, 198, 199, 200, 201, 202,
	205, 206, 208, 209, 210, 211,
	215, 217, 556, 557, 558, 559,
	594, 603, 611, 615, 632, 633, 634
\CommandLine	390, 458, 463
\count@	561, 590,
	594, 596, 597, 601, 603, 604,
	605, 609, 611, 614, 615, 619, 620
\countdef	561
\csname	14,
	21, 50, 66, 76, 132, 139, 168,
	184, 194, 236, 256, 289, 348,
	364, 560, 563, 566, 569, 624, 651
\CurrentOption	114
<b>D</b>	
\DeclareBoolOption	267, 268, 274, 276
\DeclareComplementaryOption	277
\DeclareOption	113
\DeclareStringOption	272, 273, 275
\DeclareVoidOption	270, 271, 538, 542
<b>E</b>	
\empty	17, 18, 135, 136
\end	652
\endcsname	14,
	21, 50, 66, 76, 132, 139, 168,
	184, 194, 236, 256, 289, 348,
	364, 560, 563, 566, 569, 624, 651
\endinput	29, 109, 147, 230
\endlinechar	4, 35,
	71, 77, 89, 122, 153, 189, 195, 207
\epstopdfcall	380, 513
\epstopdfDeclareGraphicsRule	499, 516
\epstopdfsetup	4, 114, 240, 251, 278, 279
\errmessage	613
\ETE@@Date	329, 332
\ETE@@Time	335, 337
\ETE@append	541
\ETE@AtEnd	95, 96, 109, 110, 112,
	117, 213, 214, 230, 241, 252, 547
\ETE@Date	322, 326
\ETE@DefCommandLine	389, 453
\ETE@DefX	392, 405
\ETE@epstopdf	517, 519
\ETE@FileInfo	303, 319, 456, 457, 466
\ETE@GenerateName	440, 446, 496
\ETE@Gin@setfile	399, 495
\ETE@IfFileExists	362, 367, 418
\ETE@InsideSetfiletrue	404, 411
\ETE@Make	290, 305, 462
\ETE@OrgGin@setfile	349, 352, 401, 408, 420, 483
\ETE@outdir	438
\ETE@prepend	537
\ETE@prependfalse	271
\ETE@prependtrue	270
\ETE@program@epstopdf	520
\ETE@SavedAtEnd	110, 112
\ETE@Size	324, 340
\ETE@Skip	371, 419
\ETE@suffix	416, 477, 480, 497, 522
\ETE@WarnModDate	292, 296, 301
<b>F</b>	
\filename@base	447, 474
\filename@parse	443

<b>G</b>		<b>P</b>	
\Gin@base	387, 418, 420, 430, 433, 440, 443, 473, 479, 480	\PackageInfo	26, 144
\Gin@driver	245, 551	\pdf@filemoddate	307, 322
\Gin@eext	429, 430	\pdf@filesize	324
\Gin@ext	387, 428, 432, 433	\pdf@shellescape	258
\Gin@setfile	349, 495	\pdf@strcmp	307
\GraphicsRead	413, 418, 486	\pdf@system	463
\GraphicsType	412, 484	\pdffilemoddate	298
\grffile@IfFileExists	367, 368	\PrependGraphicsExtensions	532, 539
<b>I</b>		\ProcessOptions	116
\if	390, 396	\ProvidesPackage	19, 67, 137, 185
\ifcase	521	<b>R</b>	
\ifETE@disable	400	\RangeCatcodeCheck	608, 636, 637, 638, 639, 640, 641, 642, 643, 644, 645, 646, 647
\ifETE@InsideSetfile	379, 381	\RangeCatcodeInvalid	600, 628, 629, 630, 631
\ifETE@prefersuffix	414	\repeat	575, 587, 598, 606, 621
\ifETE@prepend	269, 531	\RequirePackage	111, 231, 232, 233, 234, 357
\ifETE@update	291, 306	\resetatcatcode	553
\ifETE@verbose	454, 464	\RestoreCatcodes	589, 592, 593, 648
\IfFileExists	365	<b>S</b>	
\ifnum	258, 307, 596, 604, 611, 619	\setkeys	278
\ifx	15, 18, 21, 50, 58, 61, 133, 136, 139, 168, 176, 179, 236, 245, 256, 289, 327, 341, 348, 364, 407, 416, 428, 439, 471, 477, 500, 521, 522, 560, 563, 566, 569, 624	\SetupKeyvalOptions	266
\immediate	23, 52, 141, 170	\SourceExt	285, 429, 432, 435, 436
\includegraphics	459	\SourceFile	430, 433, 456, 462
\input	550, 552, 625	\space	298, 334, 344, 352, 520, 527, 614, 615, 623
\InputIfFileExists	545, 546	<b>T</b>	
\iterate	577, 579, 581	\Test	627, 650
<b>L</b>		\the	77, 78, 79, 80, 81, 82, 83, 84, 97, 195, 196, 197, 198, 199, 200, 201, 202, 215, 594, 614, 615
\LoadCommand	625, 635	\TMP@EnsureCode	94, 101, 102, 103, 104, 105, 106, 107, 108, 212, 219, 220, 221, 222, 223, 224, 225, 226, 227, 228, 229
\loop	575, 591, 602, 610	<b>W</b>	
<b>M</b>		\write	23, 52, 141, 170
\MessageBreak	298, 333, 343, 353, 356, 358, 456, 457, 458	<b>X</b>	
<b>N</b>		\x	14, 15, 18, 22, 26, 28, 51, 56, 66, 75, 87, 132, 133, 136, 140, 144, 146, 169, 174, 184, 193, 205, 244, 245, 371, 394, 407, 444, 451, 470, 491
\newcommand	240, 251, 278, 380, 499		
\newif	269, 379		
\next	423, 425, 581, 583, 585		
\number	616		
<b>O</b>			
\OutputDirectory	438, 439, 447, 471, 474, 521		
\OutputFile	440, 445, 457, 462, 466, 488, 527		