

The ifpdf package

Heiko Oberdiek
<oberdiek@uni-freiburg.de>

2009/04/10 v2.0

Abstract

This package looks for pdfTeX in pdf mode and implements and sets the switch `\ifpdf`. The detection is based on `\pdfoutput` and the package will not change this value. It works with plain or L^AT_EX formats.

Contents

| | | |
|----------|---|-----------|
| 1 | Documentation | 2 |
| 1.1 | Introduction | 2 |
| 1.2 | Usage | 2 |
| 1.3 | Specification | 3 |
| 2 | Implementation | 3 |
| 2.1 | Reload check and package identification | 3 |
| 2.2 | Catcodes | 4 |
| 2.3 | Check for previously defined <code>\ifpdf</code> | 5 |
| 2.4 | <code>\pdfoutput</code> and L ^A T _E X | 5 |
| 2.5 | <code>\ifpdf</code> | 6 |
| 2.6 | Test for fool attempts | 6 |
| 2.7 | Protocol entry | 7 |
| 3 | Test | 7 |
| 3.1 | Catcode checks for loading | 7 |
| 4 | Installation | 8 |
| 4.1 | Download | 8 |
| 4.2 | Bundle installation | 9 |
| 4.3 | Package installation | 9 |
| 4.4 | Refresh file name databases | 9 |
| 4.5 | Some details for the interested | 9 |
| 5 | History | 10 |
| | [2001/06/14 v1.0] | 10 |
| | [2001/07/14 v1.1] | 10 |
| | [2001/09/26 v1.2] | 10 |
| | [2005/07/22 v1.3] | 10 |
| | [2006/02/20 v1.4] | 10 |
| | [2007/09/09 v1.5] | 11 |
| | [2007/12/12 v1.6] | 11 |
| | [2008/12/12 v1.7] | 11 |
| | [2009/04/10 v2.0] | 11 |
| 6 | Index | 11 |

1 Documentation

1.1 Introduction

It is commonly known that Hàn Thê Thành's pdfT_EX generates PDF output directly and many people uses pdfT_EX for this purpose. However the DVI output was never thrown away. In contrary, he new features for typesetting that works in both PDF and DVI mode.

In the meantime many T_EX distributions replace the traditional T_EX binary with pdfT_EX. Then, for example, called as `latex` pdfT_EX works in DVI mode with the L^AT_EX format preloaded, called as `pdflatex` pdfT_EX starts in PDF mode.

Often packages or users want to know, whether the current document is typeset by pdfT_EX in PDF mode, because the different modes have different capabilities (color setting, graphics inclusion, ...). For this purpose pdfT_EX's `\pdfoutput` can be asked.

As regulary reader of T_EX newsgroups and mailing lists I could observe many problems with this task. Common errors are:

- pdfT_EX has *two* modes. Using pdfT_EX does not mean that the user always want to have PDF mode. For example, the PostScript support is better in DVI mode in conjunction with a PostScript aware DVI driver (e.g. `dvips`). Also the additional typesetting features are mode independent and also available in DVI mode.
- L^AT_EX's `\@ifundefined` inherited the side effect from `\csname`. Unknown commands are defined with the meaning of `\relax`. If it is checked, whether `\pdfoutput` is defined, then this should not be forgotten.
- Having `\pdfoutput` does not automatically mean PDF mode. Also the value of `\pdfoutput` must be asked.
- `\pdfoutput` must not be destroyed in some way. Later code and packages are fooled then and will perhaps make wrong decisions. For example they may drop support for PDF mode, because they do not know that pdfT_EX is running at all.

Robin Fairbairns provides an entry for this topic in his excellent FAQ (<http://www.tex.ac.uk/faq>): **Am I using PDFT_EX?**

1.2 Usage

The package `ifpdf` can be used with both plain-T_EX and L^AT_EX:

plain-T_EX: `\input ifpdf.sty`

L^AT_EX 2_ε: `\usepackage{ifpdf}`

`\ifpdf` The package provides the switch `\ifpdf`:

```
\ifpdf
... do things, if pdfTEX is running in pdf mode ...
\else
... other TEX or pdfTEX in dvi mode ...
\fi
```

Users of the package `ifthen` can use the switch as boolean:

```
\boolean{pdf}
```

The package can also be used to set global documentclass options:

```

\RequirePackage{ifpdf}
\ifpdf
  \documentclass[pdftex,...]{...}
\else
  \documentclass[...]{...}
\fi

```

1.3 Specification

The package have the following properties:

- It asks the setting of `\pdfoutput` for detecting pdfTeX in PDF mode.
- It never changes `\pdfoutput`.
- If `\pdfoutput` is undefined or has the meaning `\relax`, but the engine provides the primitive `\pdfoutput`, then `\pdfoutput` is enabled or restored if possible (only L^AT_EX, version 0.36.0 or higher).
- It can be used with many formats including plain-T_EX and L^AT_EX.

The mode detection implements the following algorithm:

```

if undefined(\pdfoutput)
  \ifpdf := \iffalse % pdfTeX is not running
else
  if \pdfoutput ≤ 0
    \ifpdf := \iffalse % pdfTeX in DVI mode
  else
    \ifpdf := \iftrue % pdfTeX in PDF mode
  fi
fi

```

The function `undefined` checks both cases, undefined command and `\relax`.

2 Implementation

```
1 < *package >
```

2.1 Reload check and package identification

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

```

2 \begingroup
3   \catcode44 12 % ,
4   \catcode45 12 % -
5   \catcode46 12 % .
6   \catcode58 12 % :
7   \catcode64 11 % @
8   \expandafter\let\expandafter\x\csname ver@ifpdf.sty\endcsname
9   \ifcase 0%
10    \ifx\x\relax % plain
11    \else
12      \ifx\x\empty % LaTeX
13      \else
14        1%
15      \fi
16    \fi
17  \else
18    \catcode35 6 % #
19    \catcode123 1 % {
20    \catcode125 2 % }
21    \expandafter\ifx\csname PackageInfo\endcsname\relax

```

```

22     \def\x#1#2{%
23         \immediate\write-1{Package #1 Info: #2.}%
24     }%
25     \else
26         \def\x#1#2{\PackageInfo{#1}{#2, stopped}}%
27     \fi
28     \x{ifpdf}{The package is already loaded}%
29 \endgroup
30 \expandafter\endinput
31 \fi
32 \endgroup

```

Package identification:

```

33 \begingroup
34 \catcode35 6 % #
35 \catcode40 12 % (
36 \catcode41 12 % )
37 \catcode44 12 % ,
38 \catcode45 12 % -
39 \catcode46 12 % .
40 \catcode47 12 % /
41 \catcode58 12 % :
42 \catcode64 11 % @
43 \catcode123 1 % {
44 \catcode125 2 % }
45 \expandafter\ifx\csname ProvidesPackage\endcsname\relax
46     \def\x#1#2#3[#4]{\endgroup
47         \immediate\write-1{Package: #3 #4}%
48         \xdef#1{#4}%
49     }%
50 \else
51     \def\x#1#2[#3]{\endgroup
52         #2[#3]}%
53     \ifx#1\@undefined
54         \xdef#1{#3}%
55     \fi
56     \ifx#1\relax
57         \xdef#1{#3}%
58     \fi
59 }%
60 \fi
61 \expandafter\x\csname ver@ifpdf.sty\endcsname
62 \ProvidesPackage{ifpdf}%
63 [2009/04/10 v2.0 Provides the ifpdf switch (H0)]

```

2.2 Catcodes

```

64 \begingroup
65 \catcode123 1 % {
66 \catcode125 2 % }
67 \def\x{\endgroup
68     \expandafter\edef\csname ifpdf@AtEnd\endcsname{%
69         \catcode35 \the\catcode35\relax
70         \catcode64 \the\catcode64\relax
71         \catcode123 \the\catcode123\relax
72         \catcode125 \the\catcode125\relax
73     }%
74 }%
75 \x
76 \catcode35 6 % #
77 \catcode64 11 % @
78 \catcode123 1 % {
79 \catcode125 2 % }

```

```

80 \def\TMP@EnsureCode#1#2{%
81   \edef\ifpdf@AtEnd{%
82     \ifpdf@AtEnd
83     \catcode#1 \the\catcode#1\relax
84   }%
85   \catcode#1 #2\relax
86 }
87 \TMP@EnsureCode{10}{12}% ^^J
88 \TMP@EnsureCode{39}{12}% '
89 \TMP@EnsureCode{40}{12}% (
90 \TMP@EnsureCode{41}{12}% )
91 \TMP@EnsureCode{44}{12}% ,
92 \TMP@EnsureCode{45}{12}% -
93 \TMP@EnsureCode{46}{12}% .
94 \TMP@EnsureCode{47}{12}% /
95 \TMP@EnsureCode{58}{12}% :
96 \TMP@EnsureCode{60}{12}% <
97 \TMP@EnsureCode{61}{12}% =
98 \TMP@EnsureCode{94}{7}% ^
99 \TMP@EnsureCode{96}{12}% '

```

2.3 Check for previously defined \ifpdf

```

100 \begingroup
101   \expandafter\ifx\csname ifpdf\endcsname\relax
102   \else
103     \edef\i/{\expandafter\string\csname ifpdf\endcsname}%
104     \expandafter\ifx\csname PackageError\endcsname\relax
105       \def\x#1#2{%
106         \edef\z{#2}%
107         \expandafter\errhelp\expandafter{\z}%
108         \errmessage{Package ifpdf Error: #1}%
109       }%
110       \def\y{^^J}%
111       \newlinechar=10 %
112     \else
113       \def\x#1#2{%
114         \PackageError{ifpdf}{#1}{#2}%
115       }%
116       \def\y{\MessageBreak}%
117     \fi
118     \x{Name clash, \i/ is already defined}{%
119       Incompatible versions of \i/ can cause problems,\y
120       therefore package loading is aborted.%
121     }%
122   \endgroup
123   \ifpdf@AtEnd
124   \expandafter\endinput
125 \fi
126 \endgroup

```

2.4 \pdfoutput and LuaTeX

It might happen, that L^AT_EX is running, but \pdfoutput does not exist. In version 0.40 only \directlua is available at startup time. The enabling Lua function was already added in version 0.36. Thus we can ignore older versions, here \pdfoutput is available at startup time.

```

127 \begingroup
128   \expandafter\ifx\csname pdfoutput\endcsname\relax
129   \else
130     \def\skip#1\relax\endgroup{\csname fi\endcsname\endgroup}%
131     \skip
132   \fi

```

```

133 \expandafter\ifx\csname directlua\endcsname\relax
134   \def\skip#1\endgroup{\csname fi\endcsname\endgroup}%
135   \skip
136 \fi
137 \expandafter\ifx\csname RequirePackage\endcsname\relax
138   \input ifluatex.sty\relax
139 \else
140   \RequirePackage{ifluatex}[2009/04/10]%
141 \fi
142 \ifluatex
143   \ifnum\luatexversion<36 %

```

Unhappily L^AT_EX's `\primitive` (derived from pdfT_EX's `\pdfprimitive`) cannot be used:

```
\protected\gdef\pdfoutput{\primitive\pdfoutput}
```

Setting a value works, but getting fails, because T_EX does no longer see it as number. It is unexpandable and breaks numerical contexts.

```

144   \else
145     \directlua{tex.enableprimitives('ifpdf', {'pdfoutput'})}%
146     \global\let\pdfoutput\ifpdfpdfoutput
147   \fi
148 \fi
149 \relax
150 \endgroup

```

2.5 \ifpdf

`\ifpdf` Create and set the switch. `\newif` initializes the switch with `\iffalse`.

```

151 \newif\ifpdf

```

Test `\pdfoutput`. Is it defined and different from `\relax`? Someone could have used L^AT_EX internal `\@ifundefined`, or something else involving. Notice, `\csname` is executed inside a group for the test to cancel the side effect of `\csname`.

```

152 \begingroup\expandafter\expandafter\expandafter\endgroup
153 \expandafter\ifx\csname pdfoutput\endcsname\relax
154 \else
155   \ifnum\pdfoutput<1 %

```

`\pdfoutput=0` or negative, so not generating pdf.

```

156 \else
157   \pdftrue
158 \fi
159 \fi

```

2.6 Test for fool attempts

```

160 \begingroup
161   \expandafter\ifx\csname pdfoutput\endcsname\relax
162   \else
163     \escapechar=-1 %
164     \edef\m{\meaning\pdfoutput}%
165     \edef\p{%
166       \string p\string d\string f%
167       \string o\string u\string t\string p\string u\string t%
168     }%
169     \ifx\m\p
170     \else
171       \expandafter\ifx\csname PackageWarningNoLine\endcsname\relax
172       \def\PackageWarningNoLine#1#2{%
173         \immediate\write16{%
174           Package '#1' Warning: #2.%

```

```

175         }%
176     }%
177     \fi
178     \PackageWarningNoLine{ifpdf}{%
179         Someone has redefined \string\pdfoutput%
180     }%
181     \fi
182 \fi
183 \endgroup

```

2.7 Protocol entry

Log comment:

```

184 \begingroup
185   \expandafter\ifx\csname PackageInfo\endcsname\relax
186     \def\x#1#2{%
187       \immediate\write-1{Package #1 Info: #2.}%
188     }%
189   \else
190     \let\x\PackageInfo
191     \expandafter\let\csname on@line\endcsname\empty
192   \fi
193   \x{ifpdf}{pdfTeX in pdf mode \ifpdf\else not \fi detected}%
194 \endgroup
195 \ifpdf@AtEnd
196 \</package>

```

3 Test

3.1 Catcode checks for loading

```

197 \<test1>
198 \catcode'\{=1 %
199 \catcode'\}=2 %
200 \catcode'\#=6 %
201 \catcode'\@=11 %
202 \expandafter\ifx\csname count@\endcsname\relax
203   \countdef\count@=255 %
204 \fi
205 \expandafter\ifx\csname @gobble\endcsname\relax
206   \long\def\@gobble#1{}%
207 \fi
208 \expandafter\ifx\csname @firstofone\endcsname\relax
209   \long\def\@firstofone#1{#1}%
210 \fi
211 \expandafter\ifx\csname loop\endcsname\relax
212   \expandafter\@firstofone
213 \else
214   \expandafter\@gobble
215 \fi
216 {%
217   \def\loop#1\repeat{%
218     \def\body{#1}%
219     \iterate
220   }%
221   \def\iterate{%
222     \body
223     \let\next\iterate
224   \else
225     \let\next\relax
226   \fi

```

```

227     \next
228 }%
229 \let\repeat=\fi
230 }%
231 \def\RestoreCatcodes{}
232 \count@=0 %
233 \loop
234   \edef\RestoreCatcodes{%
235     \RestoreCatcodes
236     \catcode\the\count@=\the\catcode\count@\relax
237   }%
238 \ifnum\count@<255 %
239   \advance\count@ 1 %
240 \repeat
241
242 \def\RangeCatcodeInvalid#1#2{%
243   \count@=#1\relax
244   \loop
245     \catcode\count@=15 %
246     \ifnum\count@<#2\relax
247       \advance\count@ 1 %
248     \repeat
249 }
250 \expandafter\ifx\csname LoadCommand\endcsname\relax
251   \def\LoadCommand{\input ifpdf.sty\relax}%
252 \fi
253 \def\Test{%
254   \RangeCatcodeInvalid{0}{47}%
255   \RangeCatcodeInvalid{58}{64}%
256   \RangeCatcodeInvalid{91}{96}%
257   \RangeCatcodeInvalid{123}{255}%
258   \catcode'\@=12 %
259   \catcode'\=0 %
260   \catcode'\{=1 %
261   \catcode'\}=2 %
262   \catcode'\#=6 %
263   \catcode'\[=12 %
264   \catcode'\]=12 %
265   \catcode'\%=14 %
266   \catcode'\ =10 %
267   \catcode13=5 %
268   \LoadCommand
269   \RestoreCatcodes
270 }
271 \Test
272 \csname @@end\endcsname
273 \end
274 </test1>

```

4 Installation

4.1 Download

Package. This package is available on CTAN¹:

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

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

¹<http://ftp.ctan.org/tex-archive/>

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

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

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

4.2 Bundle installation

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

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

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

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

4.3 Package installation

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

```
tex ifpdf.dtx
```

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

| | |
|-----------------------------------|--|
| <code>ifpdf.sty</code> | → <code>tex/generic/oberdiek/ifpdf.sty</code> |
| <code>ifpdf.pdf</code> | → <code>doc/latex/oberdiek/ifpdf.pdf</code> |
| <code>test/ifpdf-test1.tex</code> | → <code>doc/latex/oberdiek/test/ifpdf-test1.tex</code> |
| <code>ifpdf.dtx</code> | → <code>source/latex/oberdiek/ifpdf.dtx</code> |

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

4.4 Refresh file name databases

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

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 ifpdf.pdf unpack_files output .
```

Unpacking with L^AT_EX. The .dtx chooses its action depending on the format:

plain-T_EX: Run docstrip and extract the files.

L^AT_EX: Generate the documentation.

If you insist on using L^AT_EX for docstrip (really, docstrip does not need L^AT_EX), then inform the autodetect routine about your intention:

```
latex \let\install=y\input{ifpdf.dtx}
```

Do not forget to quote the argument according to the demands of your shell.

Generating the documentation. You can use both the .dtx or the .drv to generate the documentation. The process can be configured by the configuration file ltxdoc.cfg. For instance, put this line into this file, if you want to have A4 as paper format:

```
\PassOptionsToClass{a4paper}{article}
```

An example follows how to generate the documentation with pdfL^AT_EX:

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

5 History

[2001/06/14 v1.0]

- First public version.

[2001/07/14 v1.1]

- Documentation addition: global options

[2001/09/26 v1.2]

- Documentation typo corrected.
- Version number corrected.
- Line number in log entry removed.

[2005/07/22 v1.3]

- Some source code comments from Robin Fairbairns added.
- Bug fix for negative values of \pdfoutput (Oleg Katsitadze)
- LPPL 1.3
- Installation section with locations added.

[2006/02/20 v1.4]

- DTX framework.
- More robust check in case of undefined \pdfoutput.
- Extended documentation.

[2007/09/09 v1.5]

- Catcode settings added.

[2007/12/12 v1.6]

- Minor update.

[2008/12/12 v1.7]

- Fix in documentation for `\boolean` (found by S. Venkataraman).
- Code is not changed.

[2009/04/10 v2.0]

- Support for L^AT_EX 0.40 added.
- Checks, whether `\pdfoutput` was changed.

6 Index

Numbers written in *italic* refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in roman refer to the code lines where the entry is used.

| Symbols | D |
|--|---|
| <code>\#</code> 200, 262 | <code>\directlua</code> 145 |
| <code>\%</code> 265 | |
| <code>\@</code> 201, 258 | E |
| <code>\@firstofone</code> 209, 212 | <code>\empty</code> 12, 191 |
| <code>\@gobble</code> 206, 214 | <code>\end</code> 273 |
| <code>\@undefined</code> 53 | <code>\endcsname</code> 8, 21, 45, 61, |
| <code>\[</code> 263 | 68, 101, 103, 104, 128, 130, 133, |
| <code>\]</code> 179, 259 | 134, 137, 153, 161, 171, 185, |
| <code>\{</code> 198, 260 | 191, 202, 205, 208, 211, 250, 272 |
| <code>\}</code> 199, 261 | <code>\endinput</code> 30, 124 |
| <code>\]</code> 264 | <code>\errhelp</code> 107 |
| | <code>\errmessage</code> 108 |
| | <code>\escapechar</code> 163 |
| <code>_</code> 266 | |
| A | I |
| <code>\advance</code> 239, 247 | <code>\i</code> 103, 118, 119 |
| | <code>\ifcase</code> 9 |
| B | <code>\ifluatex</code> 142 |
| <code>\body</code> 218, 222 | <code>\ifnum</code> 143, 155, 238, 246 |
| | <code>\ifpdf</code> 2, <u>151</u> , 193 |
| C | <code>\ifpdf@AtEnd</code> 81, 82, 123, 195 |
| <code>\catcode</code> ... 3, 4, 5, 6, 7, 18, 19, 20, | <code>\ifpdfpdfoutput</code> 146 |
| 34, 35, 36, 37, 38, 39, 40, 41, 42, | <code>\ifx</code> . 10, 12, 21, 45, 53, 56, 101, 104, |
| 43, 44, 65, 66, 69, 70, 71, 72, 76, | 128, 133, 137, 153, 161, 169, |
| 77, 78, 79, 83, 85, 198, 199, 200, | 171, 185, 202, 205, 208, 211, 250 |
| 201, 236, 245, 258, 259, 260, | <code>\immediate</code> 23, 47, 173, 187 |
| 261, 262, 263, 264, 265, 266, 267 | <code>\input</code> 138, 251 |
| <code>\count@</code> 203, 232, | <code>\iterate</code> 219, 221, 223 |
| 236, 238, 239, 243, 245, 246, 247 | |
| <code>\countdef</code> 203 | L |
| <code>\csname</code> 8, 21, 45, 61, | <code>\LoadCommand</code> 251, 268 |
| 68, 101, 103, 104, 128, 130, 133, | <code>\loop</code> 217, 233, 244 |
| 134, 137, 153, 161, 171, 185, | <code>\luatexversion</code> 143 |
| 191, 202, 205, 208, 211, 250, 272 | |
| | M |
| | <code>\m</code> 164, 169 |

| | | | |
|-----------------------------|-------------------------|---------------------|--|
| \meaning | 164 | \RestoreCatcodes .. | 231, 234, 235, 269 |
| \MessageBreak | 116 | | |
| N | | | |
| \newif | 151 | \skip | 130, 131, 134, 135 |
| \newlinechar | 111 | T | |
| \next | 223, 225, 227 | \Test | 253, 271 |
| P | | | |
| \p | 165, 169 | \the | 69, 70, 71, 72, 83, 236 |
| \PackageError | 114 | \TMP@EnsureCode ... | 80, 87, 88, 89, |
| \PackageInfo | 26, 190 | | 90, 91, 92, 93, 94, 95, 96, 97, 98, 99 |
| \PackageWarningNoLine | 172, 178 | W | |
| \pdfoutput | 146, 155, 164 | \write | 23, 47, 173, 187 |
| \pdftrue | 157 | X | |
| \ProvidesPackage | 62 | \x .. | 8, 10, 12, 22, 26, 28, 46, 51, 61, |
| R | | | |
| \RangeCatcodeInvalid | 242, 254, 255, 256, 257 | \y | 110, 116, 119 |
| \repeat | 217, 229, 240, 248 | Z | |
| \RequirePackage | 140 | \z | 106, 107 |