The **ifpdf** package

Heiko Oberdiek
<heiko.oberdiek at googlemail.com>

2010/01/28 v2.1

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 LaTeX formats.

Contents

1 Documentation

1.1 Introduction

It is commonly known that Hàn Thê Thành's pdfTeX generates PDF output directly and many people uses pdfTeX 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 TeX distributions replace the traditional TeX binary with pdfTeX. Then, for example, called as `latex` pdfTeX works in DVI mode with the LaTeX format preloaded, called as `pdflatex` pdfTeX starts in PDF mode.

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

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

- pdfTeX has *two* modes. Using pdfTeX 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.

- LaTeX'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 pdfTeX 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 PDFTeX?

1.2 Usage

The package ifpdf can be used with both plain-TeX and LaTeX:

plain-TeX: `\input ifpdf.sty`

LaTeX 2ε: `\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 LuaTEX, version 0.36.0 or higher).

- It can be used with many formats including plain-TEX and LATEX.

The mode detection implements the following algorithm:

```
if undefined(\pdfoutput)
    \ifpdf := \iffalse % pdfTEXis 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 LATEX.

```
 2 \begingroup
 3   \catcode44 12 % ,
 4   \catcode45 12 % -
 5   \catcode46 12 % .
 6   \catcode58 12 % :
 7   \catcode64 11 % @
 8   \catcode123 1 % {
 9   \catcode125 2 % }
10   \expandafter\let\expandafter\x\csname ver@ifpdf.sty\endcsname
11   \ifx\x\relax % plain-TeX, first loading
12   \else
13     \def\empty{}%
14     \ifx\x\empty % LaTeX, first loading,
15       % variable is initialized, but \ProvidesPackage not yet seen
16     \else
17       \catcode35 6 % #
18       \expandafter\ifx\csname PackageInfo\endcsname\relax
19         \def\x#1#2{%
20           \immediate\write-1{Package #1 Info: #2.}%
21         }%
```

```
22    \else
23      \def\x#1#2{\PackageInfo{#1}{#2, stopped}}%
24    \fi
25    \x{ifpdf}{The package is already loaded}%
26    \aftergroup\endinput
27   \fi
28  \fi
29 \endgroup
```

Package identification:

```
30 \begingroup
31  \catcode35 6 % #
32  \catcode40 12 % (
33  \catcode41 12 % )
34  \catcode44 12 % ,
35  \catcode45 12 % -
36  \catcode46 12 % .
37  \catcode47 12 % /
38  \catcode58 12 % :
39  \catcode64 11 % @
40  \catcode91 12 % [
41  \catcode93 12 % ]
42  \catcode123 1 % {
43  \catcode125 2 % }
44  \expandafter\ifx\csname ProvidesPackage\endcsname\relax
45    \def\x#1#2#3[#4]{\endgroup
46      \immediate\write-1{Package: #3 #4}%
47      \xdef#1{#4}%
48    }%
49  \else
50    \def\x#1#2[#3]{\endgroup
51      #2[{#3}]%
52      \ifx#1\@undefined
53        \xdef#1{#3}%
54      \fi
55      \ifx#1\relax
56        \xdef#1{#3}%
57      \fi
58    }%
59  \fi
60 \expandafter\x\csname ver@ifpdf.sty\endcsname
61 \ProvidesPackage{ifpdf}%
62   [2010/01/28 v2.1 Provides the ifpdf switch (HO)]
```

2.2   Catcodes

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

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

2.3   Check for previously defined \ifpdf

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

2.4   \pdfoutput and LuaTeX

It might happen, that LuaTeX 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.

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

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

Unhappily LUATEX's `\primitive` (derived from pdfTeX's `\pdfprimitive`) cannot
be used:

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

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

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

2.5 `\ifpdf`

`\ifpdf` Create and set the switch. `\newif` initializes the switch with `\iffalse`. `\newif`
is `\outer` in plain-TEX.

```
150 \begingroup\expandafter\expandafter\expandafter\endgroup
151 \expandafter\ifx\csname newif\endcsname\relax
152   \edef\pdffalse{%
153     \let
154     \expandafter\noexpand\csname ifpdf\endcsname
155     \expandafter\noexpand\csname iffalse\endcsname
156   }%
157   \edef\pdftrue{%
158     \let
159     \expandafter\noexpand\csname ifpdf\endcsname
160     \expandafter\noexpand\csname iftrue\endcsname
161   }%
162   \pdffalse
163 \else
164   \csname newif\expandafter\endcsname\csname ifpdf\endcsname
165 \fi
```

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

```
166 \begingroup\expandafter\expandafter\expandafter\endgroup
167 \expandafter\ifx\csname pdfoutput\endcsname\relax
168 \else
169   \ifnum\pdfoutput<1 %
```

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

```
170   \else
171     \pdftrue
172   \fi
173 \fi
```

2.6   Test for fool attempts

```
174 \begingroup
175   \expandafter\ifx\csname pdfoutput\endcsname\relax
176   \else
177     \escapechar=-1 %
178     \edef\m{\meaning\pdfoutput}%
179     \edef\p{%
180       \string p\string d\string f%
181       \string o\string u\string t\string p\string u\string t%
182     }%
183     \ifx\m\p
184     \else
185       \expandafter\ifx\csname PackageWarningNoLine\endcsname\relax
186         \def\PackageWarningNoLine#1#2{%
187           \immediate\write16{%
188             Package '#1' Warning: #2.%
189           }%
190         }%
191       \fi
192       \PackageWarningNoLine{ifpdf}{%
193         Someone has redefined \string\\pdfoutput%
194       }%
195     \fi
196   \fi
197 \endgroup
```

2.7   Protocol entry

Log comment:

```
198 \begingroup
199   \expandafter\ifx\csname PackageInfo\endcsname\relax
200     \def\x#1#2{%
201       \immediate\write-1{Package #1 Info: #2.}%
202     }%
203   \else
204     \let\x\PackageInfo
205     \expandafter\let\csname on@line\endcsname\empty
206   \fi
207   \x{ifpdf}{pdfTeX in pdf mode \ifpdf\else not \fi detected}%
208 \endgroup

209 \ifpdf@AtEnd

210 ⟨/package⟩
```

3   Test

3.1   Catcode checks for loading

```
211 ⟨*test1⟩

212 \catcode'\{=1 %
213 \catcode'\}=2 %
214 \catcode'\#=6 %
215 \catcode'\@=11 %
216 \expandafter\ifx\csname count@\endcsname\relax
217   \countdef\count@=255 %
218 \fi
219 \expandafter\ifx\csname @gobble\endcsname\relax
220   \long\def\@gobble#1{}%
221 \fi
222 \expandafter\ifx\csname @firstofone\endcsname\relax
223   \long\def\@firstofone#1{#1}%
```

```
224 \fi
225 \expandafter\ifx\csname loop\endcsname\relax
226   \expandafter\@firstofone
227 \else
228   \expandafter\@gobble
229 \fi
230 {%
231   \def\loop#1\repeat{%
232     \def\body{#1}%
233     \iterate
234   }%
235   \def\iterate{%
236     \body
237       \let\next\iterate
238     \else
239       \let\next\relax
240     \fi
241     \next
242   }%
243   \let\repeat=\fi
244 }%
245 \def\RestoreCatcodes{}
246 \count@=0 %
247 \loop
248   \edef\RestoreCatcodes{%
249     \RestoreCatcodes
250     \catcode\the\count@=\the\catcode\count@\relax
251   }%
252 \ifnum\count@<255 %
253   \advance\count@ 1 %
254 \repeat
255
256 \def\RangeCatcodeInvalid#1#2{%
257   \count@=#1\relax
258   \loop
259     \catcode\count@=15 %
260   \ifnum\count@<#2\relax
261     \advance\count@ 1 %
262   \repeat
263 }
264 \expandafter\ifx\csname LoadCommand\endcsname\relax
265   \def\LoadCommand{\input ifpdf.sty\relax}%
266 \fi
267 \def\Test{%
268   \RangeCatcodeInvalid{0}{47}%
269   \RangeCatcodeInvalid{58}{64}%
270   \RangeCatcodeInvalid{91}{96}%
271   \RangeCatcodeInvalid{123}{255}%
272   \catcode`\@=12 %
273   \catcode`\\=0 %
274   \catcode`\{=1 %
275   \catcode`\}=2 %
276   \catcode`\#=6 %
277   \catcode`\[=12 %
278   \catcode`\]=12 %
279   \catcode`\%=14 %
280   \catcode`\ =10 %
281   \catcode13=5 %
282   \LoadCommand
283   \RestoreCatcodes
284 }
285 \Test
```

```
286 \csname @@end\endcsname
287 \end
```

```
288 ⟨/test1⟩
```

4 Installation

4.1 Download

Package. This package is available on CTAN[1]:

CTAN:macros/latex/contrib/oberdiek/ifpdf.dtx The source file.

CTAN:macros/latex/contrib/oberdiek/ifpdf.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 TEX 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-TEX:

```
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):

```
ifpdf.sty              → tex/generic/oberdiek/ifpdf.sty
ifpdf.pdf              → doc/latex/oberdiek/ifpdf.pdf
test/ifpdf-test1.tex → doc/latex/oberdiek/test/ifpdf-test1.tex
ifpdf.dtx              → source/latex/oberdiek/ifpdf.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 (teTEX, mikTEX, . . .) relies on file name databases, you must refresh these. For example, teTEX users run texhash or mktexlsr.

---

[1] ftp://ftp.ctan.org/tex-archive/

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 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{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 pdfLaTeX:

```
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 LuaTeX 0.40 added.

- Checks, whether `\pdfoutput` was changed.

[2010/01/28 v2.1]

- Compatibility to iniTeX added.

6 Index

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

11