

# The ifluatex package

Heiko Oberdiek  
<oberdiek@uni-freiburg.de>

2007/12/12 v1.0

## Abstract

This package looks for L<sup>A</sup>T<sub>E</sub>X regardless of its mode and provides the switch `\ifluatex`. It works with plain-T<sub>E</sub>X or L<sup>A</sup>T<sub>E</sub>X.

## Contents

<b>1</b>	<b>Documentation</b>	<b>1</b>
<b>2</b>	<b>Implementation</b>	<b>2</b>
2.1	Reload check and package identification . . . . .	2
2.2	Catcodes . . . . .	3
2.3	Check for previously defined <code>\ifluatex</code> . . . . .	3
2.4	<code>\ifluatex</code> . . . . .	4
2.5	Protocol entry . . . . .	4
<b>3</b>	<b>Test</b>	<b>4</b>
3.1	Catcode checks for loading . . . . .	4
<b>4</b>	<b>Installation</b>	<b>6</b>
4.1	Download . . . . .	6
4.2	Bundle installation . . . . .	6
4.3	Package installation . . . . .	6
4.4	Refresh file name databases . . . . .	6
4.5	Some details for the interested . . . . .	7
<b>5</b>	<b>History</b>	<b>7</b>
	[2007/12/12 v1.0] . . . . .	7
<b>6</b>	<b>Index</b>	<b>7</b>

## 1 Documentation

The package ifluatex can be used with both plain-T<sub>E</sub>X and L<sup>A</sup>T<sub>E</sub>X:

**plain-T<sub>E</sub>X:** `\input ifluatex.sty`

**L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>:** `\usepackage{ifluatex}`

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

```
\ifluatex
LATEX is running
\else
Without LATEX
\fi
```

If you want to detect the mode (DVI or PDF), then use package ifpdf. L<sup>A</sup>T<sub>E</sub>X has inherited `\pdfoutput` from pdfT<sub>E</sub>X.

## 2 Implementation

```
1 (*package)
```

### 2.1 Reload check and package identification

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

```
2 \begingroup
3   \catcode44 12 % ,
4   \catcode45 12 % -
5   \catcode46 12 % .
6   \catcode58 12 % :
7   \catcode64 11 % @
8   \expandafter\let\expandafter\x\csname ver@ifluatex.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@ifluatex{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\relax
54       \xdef#1{#3}%
55     \fi
56   }%
```

```

57 \fi
58 \expandafter\x\csname ver@ifluatex.sty\endcsname
59 \ProvidesPackage{ifluatex}%
60 [2007/12/12 v1.0 Provides the ifluatex switch (H0)]

```

## 2.2 Catcodes

```

61 \begingroup
62 \catcode123 1 % {
63 \catcode125 2 % }
64 \def\x{\endgroup
65 \expandafter\edef\csname ifluatex@AtEnd\endcsname{%
66 \catcode35 \the\catcode35\relax
67 \catcode64 \the\catcode64\relax
68 \catcode123 \the\catcode123\relax
69 \catcode125 \the\catcode125\relax
70 }%
71 }%
72 \x
73 \catcode35 6 % #
74 \catcode64 11 % @
75 \catcode123 1 % {
76 \catcode125 2 % }
77 \def\TMP@EnsureCode#1#2{%
78 \edef\ifluatex@AtEnd{%
79 \ifluatex@AtEnd
80 \catcode#1 \the\catcode#1\relax
81 }%
82 \catcode#1 #2\relax
83 }
84 \TMP@EnsureCode{10}{12}% ^^J
85 \TMP@EnsureCode{44}{12}% ,
86 \TMP@EnsureCode{45}{12}% -
87 \TMP@EnsureCode{46}{12}% .
88 \TMP@EnsureCode{47}{12}% /
89 \TMP@EnsureCode{58}{12}% :
90 \TMP@EnsureCode{61}{12}% =
91 \TMP@EnsureCode{94}{7}% ^

```

## 2.3 Check for previously defined \ifluatex

```

92 \begingroup
93 \expandafter\ifx\csname ifluatex\endcsname\relax
94 \else
95 \edef\i/{\expandafter\string\csname ifluatex\endcsname}%
96 \expandafter\ifx\csname PackageError\endcsname\relax
97 \def\x#1#2{%
98 \edef\z{#2}%
99 \expandafter\errhelp\expandafter{\z}%
100 \errmessage{Package ifluatex Error: #1}%
101 }%
102 \def\y{^^J}%
103 \newlinechar=10 %
104 \else
105 \def\x#1#2{%
106 \PackageError{ifluatex}{#1}{#2}%
107 }%
108 \def\y{\MessageBreak}%
109 \fi
110 \x{Name clash, \i/ is already defined}{%
111 Incompatible versions of \i/ can cause problems,\y
112 therefore package loading is aborted.%
113 }%

```

```

114     \endgroup
115     \ifluatex@AtEnd
116     \expandafter\endinput
117 \fi
118 \endgroup

```

## 2.4 \ifluatex

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

```

119 \newif\ifluatex

```

Test `\pdfoutput`. Is it defined and different from `\relax`? Someone could have used L<sup>A</sup>T<sub>E</sub>X internal `\@ifundefined`, or something else involving. Notice, `\csname` is executed inside a group for the test to cancel the side effect of `\csname`.

```

120 \begingroup\expandafter\expandafter\expandafter\endgroup
121 \expandafter\ifx\csname luatexversion\endcsname\relax
122 \else
123     \luatextrue
124 \fi

```

## 2.5 Protocol entry

Log comment:

```

125 \begingroup
126     \expandafter\ifx\csname PackageInfo\endcsname\relax
127         \def\x#1#2{%
128             \immediate\write-1{Package #1 Info: #2.}%
129         }%
130     \else
131         \let\x\PackageInfo
132     \expandafter\let\csname on@line\endcsname\empty
133 \fi
134 \x{\ifluatex}{LuaTeX \ifluatex\else not \fi detected}%
135 \endgroup
136 \ifluatex@AtEnd
137 </package>

```

# 3 Test

## 3.1 Catcode checks for loading

```

138 <*test1>
139 \catcode'\{=1 %
140 \catcode'\}=2 %
141 \catcode'\#=6 %
142 \catcode'\@=11 %
143 \expandafter\ifx\csname count@\endcsname\relax
144     \countdef\count@=255 %
145 \fi
146 \expandafter\ifx\csname @gobble\endcsname\relax
147     \long\def\@gobble#1{%
148 \fi
149 \expandafter\ifx\csname @firstofone\endcsname\relax
150     \long\def\@firstofone#1{#1}%
151 \fi
152 \expandafter\ifx\csname loop\endcsname\relax
153     \expandafter\@firstofone
154 \else
155     \expandafter\@gobble
156 \fi
157 {%

```

```

158 \def\loop#1\repeat{%
159   \def\body{#1}%
160   \iterate
161 }%
162 \def\iterate{%
163   \body
164   \let\next\iterate
165   \else
166     \let\next\relax
167   \fi
168   \next
169 }%
170 \let\repeat=\fi
171 }%
172 \def\RestoreCatcodes{}
173 \count@=0 %
174 \loop
175   \edef\RestoreCatcodes{%
176     \RestoreCatcodes
177     \catcode\the\count@=\the\catcode\count@\relax
178   }%
179   \ifnum\count@<255 %
180     \advance\count@ 1 %
181   \repeat
182 }
183 \def\RangeCatcodeInvalid#1#2{%
184   \count@=#1\relax
185   \loop
186     \catcode\count@=15 %
187     \ifnum\count@<#2\relax
188       \advance\count@ 1 %
189     \repeat
190 }
191 \expandafter\ifx\csname LoadCommand\endcsname\relax
192   \def\LoadCommand{\input ifluatex.sty\relax}%
193 \fi
194 \def\Test{%
195   \RangeCatcodeInvalid{0}{47}%
196   \RangeCatcodeInvalid{58}{64}%
197   \RangeCatcodeInvalid{91}{96}%
198   \RangeCatcodeInvalid{123}{255}%
199   \catcode'\@=12 %
200   \catcode'\=0 %
201   \catcode'\{=1 %
202   \catcode'\}=2 %
203   \catcode'\#=6 %
204   \catcode'\[=12 %
205   \catcode'\]=12 %
206   \catcode'\%=14 %
207   \catcode'\ =10 %
208   \catcode\l3=5 %
209   \LoadCommand
210   \RestoreCatcodes
211 }
212 \Test
213 \csname @@end\endcsname
214 \end
215 </test1>

```

## 4 Installation

### 4.1 Download

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

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

[CTAN:macros/latex/contrib/oberdiek/ifluatex.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.

### 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<sub>E</sub>X:

```
tex ifluatex.dtx
```

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

<code>ifluatex.sty</code>	→ <code>tex/generic/oberdiek/ifluatex.sty</code>
<code>ifluatex.pdf</code>	→ <code>doc/latex/oberdiek/ifluatex.pdf</code>
<code>test/ifluatex-test1.tex</code>	→ <code>doc/latex/oberdiek/test/ifluatex-test1.tex</code>
<code>ifluatex.dtx</code>	→ <code>source/latex/oberdiek/ifluatex.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<sub>E</sub>X distribution (teT<sub>E</sub>X, miK<sub>T</sub>E<sub>X</sub>, ...) relies on file name databases, you must refresh these. For example, teT<sub>E</sub>X users run `texhash` or `mktextlsr`.

---

<sup>1</sup><http://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 ifluatex.pdf unpack_files output .
```

**Unpacking with  $\text{\LaTeX}$ .** The `.dtx` chooses its action depending on the format:

**plain- $\text{\TeX}$ :** Run `docstrip` and extract the files.

**$\text{\LaTeX}$ :** Generate the documentation.

If you insist on using  $\text{\LaTeX}$  for `docstrip` (really, `docstrip` does not need  $\text{\LaTeX}$ ), then inform the autodetect routine about your intention:

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

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

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

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

An example follows how to generate the documentation with `pdf $\text{\LaTeX}$` :

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

## 5 History

[2007/12/12 v1.0]

- First public version.

## 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	
<code>\#</code> .....	141, 203
<code>\%</code> .....	206
<code>\@</code> .....	142, 199
<code>\@firstofone</code> .....	150, 153
<code>\@gobble</code> .....	147, 155
<code>\[</code> .....	204
<code>\]</code> .....	200
<code>\{</code> .....	139, 201
<code>\}</code> .....	140, 202
<code>\]</code> .....	205
<code>\_</code> .....	207
<b>A</b>	
<code>\advance</code> .....	180, 188
<b>B</b>	
<code>\body</code> .....	159, 163
<b>C</b>	
<code>\catcode</code> . . .	3, 4, 5, 6, 7, 18, 19, 20, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 62, 63, 66, 67, 68, 69, 73, 74, 75, 76, 80, 82, 139, 140, 141,

142, 177, 186, 199, 200, 201, 202, 203, 204, 205, 206, 207, 208	
\count@ . . . . .	144, 173, 177, 179, 180, 184, 186, 187, 188
\countdef . . . . .	144
\csname . . . . .	8, 21, 45, 58, 65, 93, 95, 96, 121, 126, 132, 143, 146, 149, 152, 191, 213
<b>E</b>	
\empty . . . . .	12, 132
\end . . . . .	214
\endcsname . . . . .	8, 21, 45, 58, 65, 93, 95, 96, 121, 126, 132, 143, 146, 149, 152, 191, 213
\endinput . . . . .	30, 116
\errhelp . . . . .	99
\errmessage . . . . .	100
<b>I</b>	
\i . . . . .	95, 110, 111
\ifcase . . . . .	9
\ifluatex . . . . .	1, 119, 134
\ifluatex@AtEnd . . . . .	78, 79, 115, 136
\ifnum . . . . .	179, 187
\ifx . . . . .	10, 12, 21, 45, 53, 93, 96, 121, 126, 143, 146, 149, 152, 191
\immediate . . . . .	23, 47, 128
\input . . . . .	192
\iterate . . . . .	160, 162, 164
<b>L</b>	
\LoadCommand . . . . .	192, 209
\loop . . . . .	158, 174, 185
\luatextrue . . . . .	123
<b>M</b>	
\MessageBreak . . . . .	108
<b>N</b>	
\newif . . . . .	119
\newlinechar . . . . .	103
\next . . . . .	164, 166, 168
<b>P</b>	
\PackageError . . . . .	106
\PackageInfo . . . . .	26, 131
\ProvidesPackage . . . . .	59
<b>R</b>	
\RangeCatcodeInvalid . . . . .	183, 195, 196, 197, 198
\repeat . . . . .	158, 170, 181, 189
\RestoreCatcodes . . . . .	172, 175, 176, 210
<b>T</b>	
\Test . . . . .	194, 212
\the . . . . .	66, 67, 68, 69, 80, 177
\TMP@EnsureCode . . . . .	77, 84, 85, 86, 87, 88, 89, 90, 91
<b>W</b>	
\write . . . . .	23, 47, 128
<b>X</b>	
\x . . . . .	8, 10, 12, 22, 26, 28, 46, 51, 58, 64, 72, 97, 105, 110, 127, 131, 134
<b>Y</b>	
\y . . . . .	102, 108, 111
<b>Z</b>	
\z . . . . .	98, 99