

# The ifluatex package

Heiko Oberdiek  
<oberdiek@uni-freiburg.de>

2009/04/17 v1.2

## Abstract

This package looks for L<sup>A</sup>T<sub>E</sub>X regardless of its mode and provides the switch `\ifluatex`. Also it makes `\luatexversion` available if it is not present. It works with plain-T<sub>E</sub>X or L<sup>A</sup>T<sub>E</sub>X.

## Contents

<b>1 Documentation</b>	<b>1</b>
<b>2 Implementation</b>	<b>2</b>
2.1 Reload check and package identification . . . . .	2
2.2 Catcodes . . . . .	3
2.3 Macro for error messages . . . . .	4
2.4 Check for previously defined <code>\ifluatex</code> . . . . .	4
2.5 <code>\ifluatex</code> . . . . .	4
2.6 LuaT <sub>E</sub> X v0.39 . . . . .	4
2.7 Protocol entry . . . . .	6
<b>3 Test</b>	<b>6</b>
3.1 Catcode checks for loading . . . . .	6
<b>4 Reload check for plain</b>	<b>7</b>
<b>5 Installation</b>	<b>7</b>
5.1 Download . . . . .	7
5.2 Bundle installation . . . . .	8
5.3 Package installation . . . . .	8
5.4 Refresh file name databases . . . . .	8
5.5 Some details for the interested . . . . .	8
<b>6 History</b>	<b>9</b>
[2007/12/12 v1.0] . . . . .	9
[2009/04/10 v1.1] . . . . .	9
[2009/04/17 v1.2] . . . . .	9
<b>7 Index</b>	<b>9</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
LUATEX is running
\else
Without LUATEX
\fi

```

Since version 0.39 LUA<sub>T</sub>E<sub>X</sub> only provides `\directlua` at startup time. Also the syntax of `\directlua` changed in version 0.36. Thus the user might want to check the Lua<sub>T</sub>E<sub>X</sub> version. Therefore this package also makes `\luatexversion` and `\luatexrevision` available, if it is not yet done.

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

## 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   \catcode123 1 % {
9   \catcode125 2 % }
10  \expandafter\let\expandafter\x\csname ver@ifluatex.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@ifluatex{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@ifluatex.sty\endcsname
61 ProvidesPackage{ifluatex}%
62 [2009/04/17 v1.2 Provides the ifluatex switch (H0)]

```

## 2.2 Catcodes

```

63 \begingroup
64 \catcode123 1 % {
65 \catcode125 2 % }
66 \def\x{\endgroup
67   \expandafter\edef\csname ifluatex@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@ifluatex@AtEnd{%
81     \ifluatex@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 Macro for error messages

```
\ifluatex@Error

 99 \begingroup\expandafter\expandafter\expandafter\endgroup
100 \expandafter\ifx\csname PackageError\endcsname\relax
101   \def\ifluatex@Error#1#2{%
102     \begingroup
103       \newlinechar=10 %
104       \def\MessageBreak{^^J}%
105       \edef\x{\errhelp{#2}}%
106       \x
107       \errmessage{Package ifluatex Error: #1}%
108     \endgroup
109   }%
110 \else
111   \def\ifluatex@Error{%
112     \PackageError{ifluatex}%
113   }%
114 \fi
```

## 2.4 Check for previously defined \ifluatex

```
115 \begingroup
116   \expandafter\ifx\csname ifluatex\endcsname\relax
117   \else
118     \edef\i/{\expandafter\string\csname ifluatex\endcsname}%
119     \ifluatex@Error{Name clash, \i/ is already defined}{%
120       Incompatible versions of \i/ can cause problems,\MessageBreak
121       therefore package loading is aborted.%
122     }%
123   \endgroup
124   \ifluatex@AtEnd
125   \expandafter\endinput
126 \fi
127 \endgroup
```

## 2.5 \ifluatex

```
\ifluatex

128 \let\ifluatex\iffalse
```

Test `\luatexversion`. 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`.

```
129 \begingroup\expandafter\expandafter\expandafter\endgroup
130 \expandafter\ifx\csname luatexversion\endcsname\relax
131 \else
132   \expandafter\let\csname ifluatex\expandafter\endcsname
133     \csname iftrue\endcsname
134 \fi
```

## 2.6 LuaT<sub>E</sub>X v0.39

Starting with version 0.39 L<sup>U</sup>A<sub>T</sub>E<sub>X</sub> wants to provide `\directlua` as only primitive at startup time beyond vanilla T<sub>E</sub>X's primitives. Then `\directlua` exists, but `\luatexversion` cannot be found. Unhappily also the syntax of `\directlua` changed in v0.36, thus the user would want to check `\luatexversion`. Therefore we make `\luatexversion` available using L<sup>U</sup>A<sub>T</sub>E<sub>X</sub>'s Lua function `tex.enableprimitives`.

```
135 \ifluatex
136 \else
```

```

137 \begingroup\expandafter\expandafter\expandafter\endgroup
138 \expandafter\ifx\csname directlua\endcsname\relax
139 \else
140   \expandafter\let\csname ifluatex\expandafter\endcsname
141     \csname iftrue\endcsname
142   \begingroup
143     \newlinechar=10 %
144     \endlinechar=\newlinechar %
145     \ifnum0%
146       \directlua{%
147         if tex.enableprimitives then
148           tex.enableprimitives('ifluatex', {'luatexversion'})
149           tex.print('1')
150         end
151       }%
152       \ifx\ifluatexluatexversion\@undefined\else 1\fi %
153       =11 %
154       \global\let\luatexversion\ifluatexluatexversion %
155     \else %
156       \ifluatex@Error{%
157         Missing \string\luatexversion%
158       }{%
159         Update LuaTeX.%
160       }%
161     \fi %
162   \endgroup %
163 \fi
164 \fi

165 \ifluatex
166 \begingroup\expandafter\expandafter\expandafter\endgroup
167 \expandafter\ifx\csname luatexrevision\endcsname\relax
168   \ifnum\luatexversion<36 %
169   \else
170     \begingroup
171       \ifx\luatexrevision\relax
172         \let\luatexrevision\@undefined
173       \fi
174       \newlinechar=10 %
175       \endlinechar=\newlinechar %
176       \ifcase0%
177         \directlua{%
178           if tex.enableprimitives then
179             tex.enableprimitives('ifluatex', {'luatexrevision'})
180           else
181             tex.print('1')
182           end
183         }%
184         \ifx\ifluatexluatexrevision\@undefined 1\fi %
185         \relax %
186         \global\let\luatexrevision\ifluatexluatexrevision
187       \fi
188     \endgroup
189   \fi
190 \begingroup\expandafter\expandafter\expandafter\endgroup
191 \expandafter\ifx\csname luatexrevision\endcsname\relax
192   \ifluatex@Error{%
193     Missing \string\luatexrevision%
194   }{%
195     Update LuaTeX.%
196   }%
197 \fi
198 \fi

```

199 \fi

## 2.7 Protocol entry

Log comment:

```
200 \begingroup
201   \expandafter\ifx\csname PackageInfo\endcsname\relax
202     \def\x#1#2{%
203       \immediate\write-1{Package #1 Info: #2.}%
204     }%
205   \else
206     \let\x\PackageInfo
207   \expandafter\let\csname on@line\endcsname\empty
208 \fi
209 \x{ifluatex}{LuaTeX \ifluatex\else not \fi detected}%
210 \endgroup

211 \ifluatex@AtEnd
212 </package>
```

## 3 Test

### 3.1 Catcode checks for loading

```
213 <*test1>

214 \catcode'\{=1 %
215 \catcode'\}=2 %
216 \catcode'\#=6 %
217 \catcode'\@=11 %
218 \expandafter\ifx\csname count@\endcsname\relax
219   \countdef\count@=255 %
220 \fi
221 \expandafter\ifx\csname @gobble\endcsname\relax
222   \long\def\@gobble#1{%
223 \fi
224 \expandafter\ifx\csname @firstofone\endcsname\relax
225   \long\def\@firstofone#1{#1}%
226 \fi
227 \expandafter\ifx\csname loop\endcsname\relax
228   \expandafter\@firstofone
229 \else
230   \expandafter\@gobble
231 \fi
232 {%
233   \def\loop#1\repeat{%
234     \def\body{#1}%
235     \iterate
236   }%
237   \def\iterate{%
238     \body
239     \let\next\iterate
240   \else
241     \let\next\relax
242   \fi
243   \next
244 }%
245 \let\repeat=\fi
246 }%
247 \def\RestoreCatcodes{}
248 \count@=0 %
249 \loop
```

```

250 \edef\RestoreCatcodes{%
251   \RestoreCatcodes
252   \catcode\the\count@=\the\catcode\count@\relax
253 }%
254 \ifnum\count@<255 %
255   \advance\count@ 1 %
256 \repeat
257
258 \def\RangeCatcodeInvalid#1#2{%
259   \count@=#1\relax
260   \loop
261     \catcode\count@=15 %
262     \ifnum\count@<#2\relax
263       \advance\count@ 1 %
264     \repeat
265 }
266 \expandafter\ifx\csname LoadCommand\endcsname\relax
267   \def\LoadCommand{\input ifluatex.sty\relax}%
268 \fi
269 \def\Test{%
270   \RangeCatcodeInvalid{0}{47}%
271   \RangeCatcodeInvalid{58}{64}%
272   \RangeCatcodeInvalid{91}{96}%
273   \RangeCatcodeInvalid{123}{255}%
274   \catcode'\@=12 %
275   \catcode'\=0 %
276   \catcode'\{=1 %
277   \catcode'\}=2 %
278   \catcode'\#=6 %
279   \catcode'\[=12 %
280   \catcode'\]=12 %
281   \catcode'\%=14 %
282   \catcode'\_ =10 %
283   \catcode\l3=5 %
284   \LoadCommand
285   \RestoreCatcodes
286 }
287 \Test
288 \csname @@end\endcsname
289 \end
290 </test1>

```

## 4 Reload check for plain

```

291 <*test-reload1>
292 \input ifluatex.sty\relax
293 \input ifluatex.sty\relax
294 \csname @@end\endcsname\end
295 </test-reload1>
296 <*test-reload2>
297 \input miniltx.tex\relax
298 \input ifluatex.sty\relax
299 \input ifluatex.sty\relax
300 \csname @@end\endcsname\end
301 </test-reload2>

```

## 5 Installation

### 5.1 Download

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

---

<sup>1</sup>[ftp://ftp.ctan.org/tex-archive/](http://ftp.ctan.org/tex-archive/)

[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.

## 5.2 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/
```

## 5.3 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>test/ifluatex-test2.tex</code>	→ <code>doc/latex/oberdiek/test/ifluatex-test2.tex</code>
<code>test/ifluatex-test3.tex</code>	→ <code>doc/latex/oberdiek/test/ifluatex-test3.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`.

## 5.4 Refresh file name databases

If your T<sub>E</sub>X distribution (teT<sub>E</sub>X, mikT<sub>E</sub>X, ...) relies on file name databases, you must refresh these. For example, teT<sub>E</sub>X users run `texhash` or `mktextlsr`.

## 5.5 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 L<sup>A</sup>T<sub>E</sub>X.** The .dtx chooses its action depending on the format:

**plain-T<sub>E</sub>X:** Run docstrip and extract the files.

**L<sup>A</sup>T<sub>E</sub>X:** Generate the documentation.

If you insist on using L<sup>A</sup>T<sub>E</sub>X for docstrip (really, docstrip does not need L<sup>A</sup>T<sub>E</sub>X), 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 pdfL<sup>A</sup>T<sub>E</sub>X:

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

## 6 History

[2007/12/12 v1.0]

- First public version.

[2009/04/10 v1.1]

- Test adopted for L<sup>A</sup>T<sub>E</sub>X 0.39.
- Makes \luatexversion available.

[2009/04/17 v1.2]

- Fixes (Manuel Pégourié-Gonnard).
- \luatextrue and \luatexfalse are no longer defined.
- Makes \luatexrevision available, too.

## 7 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		
		\[ ..... 279
\#	216, 278	\] ..... 275
\%	281	\{ ..... 214, 276
\@	217, 274	\} ..... 215, 277
\@firstofone	225, 228	\] ..... 280
\@gobble	222, 230	
\@undefined	52, 152, 172, 184	\_ ..... 282

<b>A</b>		<code>\ifluatexluatexversion</code> . . . . 152, 154
<code>\advance</code> . . . . .	255, 263	<code>\ifnum</code> . . . . . 145, 168, 254, 262
<code>\aftergroup</code> . . . . .	26	<code>\ifx</code> . 11, 14, 18, 44, 52, 55, 100, 116, 130, 138, 152, 167, 171, 184, 191, 201, 218, 221, 224, 227, 266
<b>B</b>		<code>\immediate</code> . . . . . 20, 46, 203
<code>\body</code> . . . . .	234, 238	<code>\input</code> . . . . 267, 292, 293, 297, 298, 299
<b>C</b>		<code>\iterate</code> . . . . . 235, 237, 239
<code>\catcode</code> 3, 4, 5, 6, 7, 8, 9, 17, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 64, 65, 68, 69, 70, 71, 75, 76, 77, 78, 82, 84, 214, 215, 216, 217, 252, 261, 274, 275, 276, 277, 278, 279, 280, 281, 282, 283		
<code>\count@</code> . . . . .	219, 248, 252, 254, 255, 259, 261, 262, 263	
<code>\countdef</code> . . . . .	219	
<code>\csname</code> 10, 18, 44, 60, 67, 100, 116, 118, 130, 132, 133, 138, 140, 141, 167, 191, 201, 207, 218, 221, 224, 227, 266, 288, 294, 300		
<b>D</b>		
<code>\directlua</code> . . . . .	146, 177	
<b>E</b>		
<code>\empty</code> . . . . .	13, 14, 207	
<code>\end</code> . . . . .	289, 294, 300	
<code>\endcsname</code> 10, 18, 44, 60, 67, 100, 116, 118, 130, 132, 133, 138, 140, 141, 167, 191, 201, 207, 218, 221, 224, 227, 266, 288, 294, 300		
<code>\endinput</code> . . . . .	26, 125	
<code>\endlinechar</code> . . . . .	144, 175	
<code>\errhelp</code> . . . . .	105	
<code>\errmessage</code> . . . . .	107	
<b>I</b>		
<code>\i</code> . . . . .	118, 119, 120	
<code>\ifcase</code> . . . . .	176	
<code>\iffalse</code> . . . . .	128	
<code>\ifluatex</code> . . . . .	1, 128, 135, 165, 209	
<code>\ifluatex@AtEnd</code> . . . . .	80, 81, 124, 211	
<code>\ifluatex@Error</code> . . . .	99, 119, 156, 192	
<code>\ifluatexluatexrevision</code> . . .	184, 186	
		<b>L</b>
		<code>\LoadCommand</code> . . . . . 267, 284
		<code>\loop</code> . . . . . 233, 249, 260
		<code>\luatexrevision</code> . . . 171, 172, 186, 193
		<code>\luatexversion</code> . . . . . 154, 157, 168
		<b>M</b>
		<code>\MessageBreak</code> . . . . . 104, 120
		<b>N</b>
		<code>\newlinechar</code> . . 103, 143, 144, 174, 175
		<code>\next</code> . . . . . 239, 241, 243
		<b>P</b>
		<code>\PackageError</code> . . . . . 112
		<code>\PackageInfo</code> . . . . . 23, 206
		<code>\ProvidesPackage</code> . . . . . 15, 61
		<b>R</b>
		<code>\RangeCatcodeInvalid</code> . . . . . . . . . . 258, 270, 271, 272, 273
		<code>\repeat</code> . . . . . 233, 245, 256, 264
		<code>\RestoreCatcodes</code> . . 247, 250, 251, 285
		<b>T</b>
		<code>\Test</code> . . . . . 269, 287
		<code>\the</code> . . . . . 68, 69, 70, 71, 82, 252
		<code>\TMP@EnsureCode</code> . . . 79, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98
		<b>W</b>
		<code>\write</code> . . . . . 20, 46, 203
		<b>X</b>
		<code>\x</code> . . . 10, 11, 14, 19, 23, 25, 45, 50, 60, 66, 74, 105, 106, 202, 206, 209