

# The `luatex` package

Heiko Oberdiek  
<heiko.oberdiek at googlemail.com>

2010/03/09 v0.4

## Abstract

This package manages the new and extended features and resources that LuaTeX provides. Examples are attributes and catcode tables.

## Contents

<b>1</b>	<b>Documentation</b>	<b>2</b>
1.1	Introduction	2
1.1.1	LaTeX	2
1.1.2	plain TeX	2
1.2	Register allocation	3
1.2.1	Register with 16 bit	3
1.2.2	Insertions	3
1.3	Attributes	3
1.4	Catcode tables	4
1.4.1	Interface proposal	4
1.5	Lua module loading	5
1.5.1	Package <code>luatex-loader</code>	6
<b>2</b>	<b>Implementation</b>	<b>6</b>
2.1	Reload check and package identification	6
2.2	Catcodes	7
2.3	Check for LuaTeX	8
2.4	Provide LuaTeX primitives	8
2.5	Inherit support for $\epsilon$ -TeX	9
2.6	Adaption of $\epsilon$ -TeX's register allocation	10
2.7	plain TeX compatibility	10
2.8	Attributes	12
2.8.1	Allocation	12
2.8.2	Interface	12
2.9	Catcode tables	13
2.9.1	Allocation	13
2.9.2	<code>\SetCatcodeRange</code>	13
2.9.3	Predefined catcode tables	14
2.9.4	Number stack	14
2.9.5	Catcode regime macros	15
2.10	Lua module loader	15
2.11	Lua script	17

<b>3</b>	<b>Test</b>	<b>18</b>
3.1	Catcode checks for loading . . . . .	18
3.2	Catcode tables . . . . .	20
3.2.1	Predefined catcode tables . . . . .	20
3.2.2	Catcode table number stack . . . . .	20
3.2.3	Catcode table stack . . . . .	21
3.2.4	Catcode regime macros . . . . .	21
3.3	Attribute allocation . . . . .	22
3.4	Short test for plain T <sub>E</sub> X . . . . .	22
<b>4</b>	<b>Installation</b>	<b>22</b>
4.1	Download . . . . .	22
4.2	Bundle installation . . . . .	23
4.3	Package installation . . . . .	23
4.4	Refresh file name databases . . . . .	23
4.5	Some details for the interested . . . . .	24
<b>5</b>	<b>History</b>	<b>24</b>
[2007/12/12 v0.1]	. . . . .	24
[2009/04/10 v0.2]	. . . . .	24
[2009/12/02 v0.3]	. . . . .	24
[2010/03/09 v0.4]	. . . . .	24
<b>6</b>	<b>Index</b>	<b>25</b>

# 1 Documentation

## 1.1 Introduction

T<sub>E</sub>X provides global resources such as registers. But it does not provide an interface for managing these resources. For example, two packages want to use a counter register. If they take the same register number, then the use of both packages will conflict and they cannot be used together. Therefore formats such as plain T<sub>E</sub>X or L<sup>A</sup>T<sub>E</sub>X implement an allocation scheme for registers. A package reserves with `\newcount` an unused register number for its own exclusive use.

Nowadays T<sub>E</sub>X is not alone anymore:  $\varepsilon$ -T<sub>E</sub>X, pdfT<sub>E</sub>X and other compilers for T<sub>E</sub>X are developed that extend and add new features and resources.

Now LuaT<sub>E</sub>X has reached beta state. It inherits most of pdfT<sub>E</sub>X's features including  $\varepsilon$ -T<sub>E</sub>X. Also it implements new concepts such as attributes or catcode tables.

### 1.1.1 L<sup>A</sup>T<sub>E</sub>X

L<sup>A</sup>T<sub>E</sub>X 2 <sub>$\varepsilon$</sub>  is frozen and therefore refuses to even notice the new T<sub>E</sub>X variants. Not even the old  $\varepsilon$ -T<sub>E</sub>X is supported by its kernel. At least there is a third party package `etex` that manages the new  $\varepsilon$ -T<sub>E</sub>X resources.

This package tries to do the same for LuaT<sub>E</sub>X and starts to support at least a few of the new features.

### 1.1.2 plain T<sub>E</sub>X

L<sup>A</sup>T<sub>E</sub>X has inherited its resource handling from plain T<sub>E</sub>X. The interface is basically the same: `\newcount`, ... Therefore this package tries to follow this tradition by providing compatibility to plain T<sub>E</sub>X. It can be loaded with plain T<sub>E</sub>X and defines at least some of the features that this packages provides for L<sup>A</sup>T<sub>E</sub>X.

## 1.2 Register allocation

### 1.2.1 Register with 16 bit

Because LuaTeX is a super set of  $\varepsilon$ -TeX regarding registers, the register allocation scheme should not conflict with package `etex`. Therefore this package is loaded to inherit its allocation scheme. The only change is currently that the limit is increased to 65536 registers for the following register classes:

- `count`
- `dimen`
- `skip`
- `muskip`
- `marks`
- `toks`
- `box`

This affects the number of global and local registers. Because it is done in a package and not in the kernel, it is possible that someone loads package `etex` before uses the local allocation variants. This will prevent the extension for this register class. If more registers are needed, just load package `luatex` earlier.

### 1.2.2 Insertions

Insertions need four registers `\count`, `\dimen`, `\skip`, and `\box` with the same number. Usually they are allocated downwards from 254, 253, ... Also `\newcount`, `\newdimen`, ... fill up these register numbers from below before switching to higher register numbers by package `etex`. When this occurs, no insertions can be allocated anymore.

Therefore `\newcount`, `\newdimen`, `\newskip`, and `\newbox` are replaced by their global variants (`\globcount`, ...) that use the higher numbers immediately, leaving the room for insertions. There should not be an efficiency penalty because LuaTeX stores the registers of a class in the same Lua table unlike  $\varepsilon$ -TeX, where registers below 256 are stored in an array and higher numbers are put in a tree structure.

## 1.3 Attributes

Nodes can have custom attributes in LuaTeX. These attributes are organized by a new register class. As the other registers up to  $2^{16}$  attributes are supported. An attribute value can be negative that means the attribute is not set. Otherwise TeX's range of non-negative integers up to  $2^{31}$  are available.

`\newattribute {⟨cmd⟩}`

Macro `\newattribute` defines command `⟨cmd⟩` using `\attributedef` using a new attribute number. The new attribute is initially unset.

`\setattribute {⟨cmd⟩} {⟨value⟩}`

Macro `\setattribute` locally sets attribute command `⟨cmd⟩` to the number `⟨value⟩`. Valid values range from  $-1$  until  $2^{31}$  (the upper limit is the same as for other TeX integer numbers).

`\unsetattribute {⟨cmd⟩}`

Macro `\unsetAttribute` clears the attribute command `⟨cmd⟩`.

## 1.4 Catcode tables

LuaTeX introduces catcode tables as new feature, see documentation. There is need for discussion, how to deal best:

- `\initcatcodetable` and `\setcatcodetable` act globally.
- `\catcodetable` causes an error if used with an uninitialized catcode table.
- Large catcode table numbers should be avoided because of performance breakdown.
- Use case L<sup>A</sup>T<sub>E</sub>X package: The package must not be surprised by changed catcodes and must not surprise by changing catcodes accidentally. Catcode tables could offer a solution. At the begin a catcode regime with standard catcodes is established and the old one is restored afterwards.
- Use case: LuaTeX's `tex.print` might be used with a catcode table number, for example a table where all entries have catcode “other”.
- Readonly catcode tables.
- Is there is a need for local allocations? (Package `etex`'s `\loc` variants are not used in T<sub>E</sub>X Live 2007.)

### 1.4.1 Interface proposal

The idea: `\newcatcodetable` allocates odd numbered catcode tables. Even numbered tables are managed as stack. Also some catcode tables are defined. These must not be changed.

`\newcatcodetable {⟨cmd⟩}`

Macro `\newcatcodetable` reserves a new catcode table and remembers its number in `⟨cmd⟩`. The catcode table is initialized with ini-T<sub>E</sub>X's catcodes.

`\CatcodeTableIniTeX`  
`\CatcodeTableString`  
`\CatcodeTableOther`  
`\CatcodeTableLaTeX`

These are catcode tables and must not be changed. `\CatcodeTableIniTeX` contains the catcode settings of ini-T<sub>E</sub>X. `\CatcodeTableString` follows T<sub>E</sub>X's convention of `\string`, `\meaning` and friends. The space gets catcode 10 (space), the other characters have catcode 12 (other). In `\CatcodeTableOther` all entries have catcode 12 (other). `\CatcodeTableLaTeX` contains the setting of a pure L<sup>A</sup>T<sub>E</sub>X format ('at' is other).

`\CatcodeTableStack`  
`\IncCatcodeTableStack`  
`\DecCatcodeTableStack`

`\CatcodeTableStack` is the stack pointer. Initially it is catcode table zero. `\IncCatcodeTableStack` and `\DecCatcodeTableStack` increments and decrements the stack pointer. Currently `\IncCatcodeTableStack` does not initialize a new catcode table. Both increment and decrement operations do not set a catcode table.

```
\PushCatcodeTableNumStack
\PopCatcodeTableNumStack
```

It can be handy to have a global stack for catcode table numbers to deal with the global assignment property of `\initcatcodetable` and `\savecatcodetable`. `\PushCatcodeTableNumStack` pushes the current catcode table on the stack. `\PopCatcodeTableNumStack` pops the topmost number off the number stack to set the current catcode table. Catcode table zero is used in case of an empty stack.

```
\BeginCatcodeRegime {⟨catcodetable⟩}
\EndCatcodeRegime
```

`\BeginCatcodeRegime` remembers the current catcode table number. Then it creates and uses a fresh catcode table on the stack that is initialized by `⟨catcodetable⟩`:

```
\PushCatcodeTableNumStack
\catcodetable⟨catcodetable⟩ \IncCatcodeTableStack
\savecatcodetable\CatcodeTableStack
\catcodetable\CatcodeTableStack
```

`\EndCatcodeRegime` drops the catcode table, created by `\BeginCatcodeRegime` and sets the catcode table that was active before:

```
\DecCatcodeTableStack
\PopCatcodeTableNumStack
```

These macros solve the use case, described earlier for a  $\text{\LaTeX}$  package:

```
% package foobar.sty
\BeginCatcodeRegime\CatcodeTableLaTeX
\makeatletter
% ... package contents ...
\EndCatcodeRegime
% end of package
```

If the package wants to change catcodes after its loading, `\AtBeginDocument` or `\AtEndOfPackage` can be used.

```
\SetCatcodeRange {⟨from⟩} {⟨to⟩} {⟨catcode⟩}
```

The catcodes of characters in range from `⟨from⟩` to inclusive `⟨to⟩` are set to `⟨catcode⟩`.

## 1.5 Lua module loading

Currently  $\text{\LuaTeX}$  (version 0.20) does not support Lua script files inside `TDS:scripts//`, because Lua's mechanism for module loading does not use the `kpathsea` library. Therefore this packages appends a `kpse` loader to the list of Lua's module loaders. It finds the module `⟨module⟩` by

```
kpse.find_file("⟨module⟩.lua", "texmfscripts")
```

Unhappily `kpathsea` does not support directory components in a file name. Therefore the Lua convention is not followed to replace dots in the module name by the directory separator.

Example: A Lua script of a package `foobar` wants the following modules:

```
require("foobar.hello.world")
require("org.somewhere.xyz")
```

Then they can be find in:

```
TDS:scripts/foobar/foobar.hello.world.lua
TDS:scripts/foobar/org.somewhere.xyz.lua
```

I would have preferred the following locations, following lua conventions, e. g.:

```
TDS:scripts/foobar/hello/world.lua
TDS:scripts/foobar/org/somewhere/xyz.lua
```

But I do not know, how to achieve this in a reliable way using kpathsea.

### 1.5.1 Package luatex-loader

If someone do not need or want package luatex but it's extension for module loading, then he can use package luatex-loader. Both plain T<sub>E</sub>X and L<sup>A</sup>T<sub>E</sub>X are supported.

## 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\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@luatex.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{luatex}{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@luatex.sty\endcsname
67 \ProvidesPackage{luatex}%
68 [2010/03/09 v0.4 LuaTeX basic definition package (HO)]%

```

## 2.2 Catcodes

```

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 LuT@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\LuT@AtEnd{%
96     \LuT@AtEnd
97     \catcode#1=\the\catcode#1\relax
98   }%

```

```

99   \catcode#1=#2\relax
100 }
101 \TMP@EnsureCode{10}{12}% ^^J
102 \TMP@EnsureCode{34}{12}% "
103 \TMP@EnsureCode{36}{3}% $
104 \TMP@EnsureCode{39}{12}% '
105 \TMP@EnsureCode{40}{12}% (
106 \TMP@EnsureCode{41}{12}% )
107 \TMP@EnsureCode{42}{12}% *
108 \TMP@EnsureCode{43}{12}% +
109 \TMP@EnsureCode{44}{12}% ,
110 \TMP@EnsureCode{45}{12}% -
111 \TMP@EnsureCode{46}{12}% .
112 \TMP@EnsureCode{47}{12}% /
113 \TMP@EnsureCode{60}{12}% <
114 \TMP@EnsureCode{62}{12}% >
115 \TMP@EnsureCode{91}{12}% [
116 \TMP@EnsureCode{93}{12}% ]
117 \TMP@EnsureCode{95}{12}% _ (other!)
118 \TMP@EnsureCode{96}{12}% `
119 \edef\LuT@AtEnd{\LuT@AtEnd\noexpand\endinput}

```

## 2.3 Check for LuaTeX

Without LuaTeX there is no point in using this package.

```

120 \begingroup\expandafter\expandafter\expandafter\endgroup
121 \expandafter\ifx\csname RequirePackage\endcsname\relax
122   \input infwarerr.sty\relax
123   \input ifluatex.sty\relax
124 \else
125   \RequirePackage{infwarerr}[2007/09/09]%
126   \RequirePackage{ifluatex}[2009/04/10]%
127 \fi

128 \ifluatex
129 \else
130   \@PackageError{luatex}{%
131     This package may only be run using LuaTeX%
132   }\@ehc
133   \expandafter\LuT@AtEnd
134 \fi%

```

## 2.4 Provide LuaTeX primitives

```

135 \ifnum\luatexversion<36 %
136   \def\LuT@MakePrimitive#1{%
137     \expandafter\let\csname luatex#1\expandafter\endcsname
138     \csname #1\endcsname
139   }%
140 \else
141   \def\LuT@MakeLuatexPrimitive#1{%
142     \begingroup\expandafter\expandafter\expandafter\endgroup
143     \expandafter\ifx\csname luatex#1\endcsname\relax
144       \begingroup\expandafter\expandafter\expandafter\endgroup
145       \expandafter\ifx\csname #1\endcsname\relax
146       \else
147         \expandafter\let
148         \csname luatex#1\expandafter\endcsname
149         \csname #1\endcsname
150       \fi
151     \fi
152     \begingroup\expandafter\expandafter\expandafter\endgroup
153     \expandafter\ifx\csname luatex#1\endcsname\relax

```

```

154 \begingroup
155 \expandafter\let\csname luatex#1\endcsname\@undefined
156 \ifnum0%
157 \directlua{%
158 if tex.enableprimitives then %
159 tex.enableprimitives('luatex',{ '#1'})%
160 tex.print('1')%
161 end%
162 }%
163 \expandafter\ifx\csname luatex#1\endcsname\relax\else1\fi
164 =11 %
165 \global\expandafter\let
166 \csname luatex#1\expandafter\endcsname
167 \csname luatex#1\endcsname
168 \else
169 \@PackageError{luatex}{%
170 tex.enableprimitives failed for '#1'%
171 }\@ehc
172 \fi
173 \endgroup
174 \fi
175 }%
176 \def\LuT@MakePrimitive#1{%
177 \begingroup\expandafter\expandafter\expandafter\endgroup
178 \expandafter\ifx\csname#1\endcsname\relax
179 \begingroup
180 \expandafter\let\csname#1\endcsname\@undefined
181 \ifnum0%
182 \directlua{%
183 if tex.enableprimitives then %
184 tex.enableprimitives('',{ '#1'})%
185 tex.print('1')%
186 end%
187 }%
188 \expandafter\ifx\csname#1\endcsname\relax\else1\fi
189 =11 %
190 \global\expandafter\let
191 \csname#1\expandafter\endcsname
192 \csname#1\endcsname
193 \else
194 \@PackageError{luatex}{%
195 tex.enableprimitives failed for '#1'%
196 }\@ehc
197 \fi
198 \endgroup
199 \fi
200 }%
201 \fi
202 \LuT@MakeLuatexPrimitive{attribute}
203 \LuT@MakeLuatexPrimitive{attributedef}
204 \LuT@MakeLuatexPrimitive{catcodetable}
205 \LuT@MakeLuatexPrimitive{initcatcodetable}
206 \LuT@MakeLuatexPrimitive{luaescapestring}
207 \LuT@MakeLuatexPrimitive{savecatcodetable}
208 \LuT@MakePrimitive{numexpr}

```

## 2.5 Inherit support for $\epsilon$ -TeX

Package etex is not compatible for plain TeX. But it could be present if a format is used that is based on `etex.src`. Therefore we only load the package in case of L<sup>A</sup>T<sub>E</sub>X and tests its presence independently of the format by looking for `\et@xins`.

```

209 \begingroup\expandafter\expandafter\expandafter\endgroup
210 \expandafter\ifx\csname RequirePackage\endcsname\relax

```

```

211 \else
212   \RequirePackage{etex}[1998/03/26]%
213 \fi

```

## 2.6 Adaption of $\epsilon$ -T<sub>E</sub>X's register allocation

$\epsilon$ -T<sub>E</sub>X has increased the number of T<sub>E</sub>X registers from  $2^8$  (256) to  $2^{15}$  (32768) for a register class. LuaT<sub>E</sub>X extends the limit further to  $2^{16}$  (65536). The allocation scheme of package `etex` is not changed. But this can be subject for discussion.

If a register class hasn't registered any local registers yet, then the limit can safely be pushed to 65536.

```

214 \begingroup\expandafter\expandafter\expandafter\endgroup
215 \expandafter\ifx\csname et@xins\endcsname\relax
216   \PackageWarningNoLine{luatex}{%
217     Support for eTeX is not loaded (etex.src)%
218   }%
219 \else
220   \def\LuT@temp#1{%
221     \ifnum\count27#1=32768 %
222       \count27#1=65536 %
223     \fi
224   }%
225   \LuT@temp0%
226   \LuT@temp1%
227   \LuT@temp2%
228   \LuT@temp3%
229   \LuT@temp4%
230   \LuT@temp5%
231   \LuT@temp6%

```

$\epsilon$ -T<sub>E</sub>X uses an array for the first 256 registers and then a tree structure. LuaT<sub>E</sub>X stores all registers of a class in one Lua table. There shouldn't be large performance differences. This allows starting immediately in the extended area, leaving room for insertions.

```

232 \let\newcount\globcount
233 \let\newdimen\globdimen
234 \let\newskip\globskip
235 \let\newbox\globbox
236 \fi

```

## 2.7 plain T<sub>E</sub>X compatibility

```

\@empty
237 \expandafter\ifx\csname @empty\endcsname\relax
238   \def\@empty{}%
239 \fi

\@gobble
240 \expandafter\ifx\csname @gobble\endcsname\relax
241   \long\def\@gobble#1{}%
242 \fi

\@firstofone
243 \expandafter\ifx\csname @firstofone\endcsname\relax
244   \long\def\@firstofone#1{#1}%
245 \fi

\@firstoftwo
246 \expandafter\ifx\csname @firstoftwo\endcsname\relax
247   \long\def\@firstoftwo#1#2{#1}%
248 \fi

```

```

\@car
249 \expandafter\ifx\csname @car\endcsname\relax
250   \def\@car#1#2\@nil{#1}%
251 \fi

\@cdr
252 \expandafter\ifx\csname @cdr\endcsname\relax
253   \def\@cdr#1#2\@nil{#2}%
254 \fi

\@ifstar
255 \expandafter\ifx\csname @ifstar\endcsname\relax
256   \def\@ifstar#1{%
257     \@ifnextchar*{\@firstoftwo{#1}}%
258   }%

\@ifnextchar
259   \long\def\@ifnextchar#1#2#3{%
260     \let\reserved@d=#1%
261     \def\reserved@a{#2}%
262     \def\reserved@b{#3}%
263     \futurelet\@let@token\@ifnch
264   }%

\@ifnch
265   \def\@ifnch{%
266     \ifx\@let@token\@sptoken
267       \let\reserved@c\@xifnch
268     \else
269       \ifx\@let@token\reserved@d
270         \let\reserved@c\reserved@a
271       \else
272         \let\reserved@c\reserved@b
273       \fi
274     \fi
275     \reserved@c
276   }%

\@sptoken
277   \let\LuT@temp\:%
278   \def\:{\let\@sptoken= }%
279   \: % explicit space

\@xifnch
280   \def\:\@xifnch}%
281   \expandafter\def\:{%
282     \futurelet\@let@token\@ifnch
283   }%
284   \let\:\LuT@temp
285 \fi

\@tempcnta
286 \expandafter\ifx\csname @tempcnta\endcsname\relax
287   \csname newcount\endcsname\@tempcnta
288 \fi

\@tempcntb
289 \expandafter\ifx\csname @tempcntb\endcsname\relax
290   \csname newcount\endcsname\@tempcntb
291 \fi

```

`\LuT@newcommand`

```

292 \begingroup\expandafter\expandafter\expandafter\endgroup
293 \expandafter\ifx\csname newcommand\endcsname\relax
294   \def\LuT@newcommand#1[#2]#3{%
295     \ifx#1\undefined
296       \let#1\relax
297     \else
298       \ifx#1\relax
299       \else
300         \@PackageError{luatex}{%
301           \string#1 is already defined.\MessageBreak
302           Redefinition is skipped%
303         }\@ehc
304       \fi
305     \fi
306     \ifx#1\relax
307       \ifcase#2 %
308         \def#1{#3}%
309       \or
310         \def#1##1{#3}%
311       \or
312         \def#1##1##2{#3}%
313       \or
314         \def#1##1##2##3{#3}%
315       \or
316         \@INTERNAL@ERROR
317       \fi
318     \fi
319   }%
320 \else
321   \def\LuT@newcommand{\newcommand*}%
322 \fi

```

## 2.8 Attributes

### 2.8.1 Allocation

`\LuT@AllocAttribute`

```

323 \newcount\LuT@AllocAttribute
324 \LuT@AllocAttribute=\m@ne

```

`\newattribute`

```

325 \LuT@newcommand\newattribute[1]{%
326   \ifnum\LuT@AllocAttribute<65535 %
327     \global\advance\LuT@AllocAttribute\@ne
328     \allocationnumber\LuT@AllocAttribute
329     \global\luatexattributedef#1=\allocationnumber
330     \unsetattribute{#1}%
331     \wlog{\string#1=\string\attribute\the\allocationnumber}%
332   \else
333     \errmessage{No room for a new \string\attribute}%
334   \fi
335 }

```

### 2.8.2 Interface

`\setattribute`

```

336 \LuT@newcommand\setattribute[2]{%
337   #1=\numexpr#2\relax
338 }

```

\unsetattribute

```
339 \ifnum\luatexversion<37
340   \LuT@newcommand\LuT@UnsetAttributeValue[0]{}%
341   \let\LuT@UnsetAttributeValue\m@ne
342 \else
343   \LuT@newcommand\LuT@UnsetAttributeValue[0]{-2147483647 }%
344 \fi
345 \LuT@newcommand\unsetattribute[1]{%
346   #1=\LuT@UnsetAttributeValue
347 }
```

## 2.9 Catcode tables

### 2.9.1 Allocation

\LuT@AllocCatcodeTable

```
348 \newcount\LuT@AllocCatcodeTable
349 \LuT@AllocCatcodeTable=\m@ne
350 \newcount\CatcodeTableStack
351 \CatcodeTableStack=\z@
```

\newcatcodetable

```
352 \LuT@newcommand\newcatcodetable[1]{%
353   \ifnum\LuT@AllocCatcodeTable<1114110 % 0x10FFFF is maximal \chardef
354     % or < 268435455 % 228 - 1
355     \global\advance\LuT@AllocCatcodeTable by\tw@
356     \allocationnumber=\LuT@AllocCatcodeTable
357     \global\chardef#1=\allocationnumber
358     \wlog{%
359       \string#1=\string\catcodetable\the\allocationnumber
360     }%
361   \else
362     \errmessage{No room for a new \string\catcodetable}%
363   \fi
364 }
```

\IncCatcodeTableStack

```
365 \LuT@newcommand\IncCatcodeTableStack[0]{%
366   \ifnum\CatcodeTableStack<268435454 %
367     \global\advance\CatcodeTableStack by\tw@
368   \else
369     \@PackageError{luatex}{%
370       Catcode table stack overflow%
371     }\@ehd
372   \fi
373 }
```

\DecCatcodeTableStack

```
374 \LuT@newcommand\DecCatcodeTableStack[0]{%
375   \ifnum\CatcodeTableStack>\z@
376     \global\advance\CatcodeTableStack by-2 %
377   \else
378     \@PackageError{luatex}{%
379       Catcode table stack is empty%
380     }\@ehd
381   \fi
382 }
```

### 2.9.2 \SetCatcodeRange

\SetCatcodeRange

```

383 \LuT@newcommand\SetCatcodeRange[3]{%
384   \edef\LuT@temp{%
385     \noexpand\@tempcnta=\the\@tempcnta
386     \noexpand\@tempcntb=\the\@tempcntb
387     \noexpand\count@=\the\count@
388     \relax
389   }%
390   \@tempcnta=\numexpr#1\relax
391   \@tempcntb=\numexpr#2\relax
392   \count@=\numexpr#3\relax
393   \loop
394     \unless\ifnum\@tempcnta>\@tempcntb
395       \catcode\@tempcnta=\count@
396       \advance\@tempcnta by \@ne
397     \repeat
398   \LuT@temp
399 }

```

### 2.9.3 Predefined catcode tables

```

400 \newcatcodetable\CatcodeTableIniTeX
401 \newcatcodetable\CatcodeTableString
402 \newcatcodetable\CatcodeTableOther
403 \newcatcodetable\CatcodeTableLaTeX

404 \luatexinitcatcodetable\CatcodeTableIniTeX
405 \begingroup
406   \def\@makeoother#1{\catcode#1=12\relax}%
407   \@firstofone{%
408     \luatexcatcodetable\CatcodeTableIniTeX
409     \begingroup
410       \SetCatcodeRange{0}{8}{15}%
411       \catcode9=10 % tab
412       \catcode11=15 %
413       \catcode12=13 % form feed
414       \SetCatcodeRange{14}{31}{15}%
415       \catcode35=6 % hash
416       \catcode36=3 % dollar
417       \catcode38=4 % ampersand
418       \catcode94=7 % circumflex
419       \catcode95=8 % underscore
420       \catcode123=1 % brace left
421       \catcode125=2 % brace right
422       \catcode126=13 % tilde
423       \catcode127=15 %
424     \luatexsavecatcodetable\CatcodeTableLaTeX
425   \endgroup
426   \@makeoother{0}% nul
427   \@makeoother{13}% carriage return
428   \@makeoother{37}% percent
429   \@makeoother{92}% backslash
430   \@makeoother{127}%
431   \SetCatcodeRange{65}{90}{12}% A-Z
432   \SetCatcodeRange{97}{122}{12}% a-z
433   \luatexsavecatcodetable\CatcodeTableString
434   \@makeoother{32}% space
435   \luatexsavecatcodetable\CatcodeTableOther
436 \endgroup
437 }%

```

### 2.9.4 Number stack

`\LuT@NumStackEmpty` A special empty stack value because of `\@cdr`'s brace removal.

```

438 \def\LuT@NumStackEmpty{0}

```

`\LuT@NumStack`

```
439 \let\LuT@NumStack\LuT@NumStackEmpty
```

`\PushCatcodeTableNumStack`

```
440 \LuT@newcommand\PushCatcodeTableNumStack[0]{%
441   \xdef\LuT@NumStack{%
442     {\the\luatexcatcodetable}\LuT@NumStack
443   }%
444 }
```

`\PopCatcodeTableNumStack`

```
445 \LuT@newcommand\PopCatcodeTableNumStack[0]{%
446   \ifx\LuT@NumStack\LuT@NumStackEmpty
447     \@PackageWarning{luatex}{Empty catcode table number stack}%
448     \luatexcatcodetable\z@
449   \else
450     \luatexcatcodetable=\expandafter\@car\LuT@NumStack\@nil\relax
451     \xdef\LuT@NumStack{%
452       \expandafter\@cdr\LuT@NumStack\@nil
453     }%
454   \fi
455 }
```

### 2.9.5 Catcode regime macros

`\BeginCatcodeRegime`

```
456 \LuT@newcommand\BeginCatcodeRegime[1]{%
457   \PushCatcodeTableNumStack
458   \luatexcatcodetable=\numexpr#1\relax
459   \IncCatcodeTableStack
460   \luatexsavecatcodetable\CatcodeTableStack
461   \luatexcatcodetable\CatcodeTableStack
462 }
```

`\EndCatcodeRegime`

```
463 \LuT@newcommand\EndCatcodeRegime[0]{%
464   \DecCatcodeTableStack
465   \PopCatcodeTableNumStack
466 }
```

## 2.10 Lua module loader

```
467 \begingroup\expandafter\expandafter\expandafter\endgroup
468 \expandafter\ifx\csname RequirePackage\endcsname\relax
469   \input luatex-loader.sty\relax
470 \else
471   \RequirePackage{luatex-loader}[2010/03/09]%
472 \fi
473 \LuT@AtEnd%
474 </package>
475 <*/loader>
```

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

```
476 \begingroup\catcode61\catcode48\catcode32=10\relax%
477   \catcode13=5 % ^~M
478   \endlinechar=13 %
479   \catcode35=6 % #
480   \catcode39=12 % '
481   \catcode44=12 % ,
482   \catcode45=12 % -
```

```

483 \catcode46=12 % .
484 \catcode58=12 % :
485 \catcode64=11 % @
486 \catcode123=1 % {
487 \catcode125=2 % }
488 \expandafter\let\expandafter\x\csname ver@luatex-loader.sty\endcsname
489 \ifx\x\relax % plain-TeX, first loading
490 \else
491   \def\empty{%
492     \ifx\x\empty % LaTeX, first loading,
493       % variable is initialized, but \ProvidesPackage not yet seen
494     \else
495       \expandafter\ifx\csname PackageInfo\endcsname\relax
496         \def\x#1#2{%
497           \immediate\write-1{Package #1 Info: #2.}%
498         }%
499       \else
500         \def\x#1#2{\PackageInfo{#1}{#2, stopped}}%
501       \fi
502       \x{luatex-loader}{The package is already loaded}%
503     \aftergroup\endinput
504   \fi
505 \fi
506 \endgroup%

```

Package identification:

```

507 \begingroup\catcode61\catcode48\catcode32=10\relax%
508 \catcode13=5 % ^^M
509 \endlinechar=13 %
510 \catcode35=6 % #
511 \catcode39=12 % '
512 \catcode40=12 % (
513 \catcode41=12 % )
514 \catcode44=12 % ,
515 \catcode45=12 % -
516 \catcode46=12 % .
517 \catcode47=12 % /
518 \catcode58=12 % :
519 \catcode64=11 % @
520 \catcode91=12 % [
521 \catcode93=12 % ]
522 \catcode123=1 % {
523 \catcode125=2 % }
524 \expandafter\ifx\csname ProvidesPackage\endcsname\relax
525   \def\x#1#2#3[#4]{\endgroup
526     \immediate\write-1{Package: #3 #4}%
527     \xdef#1{#4}%
528   }%
529 \else
530   \def\x#1#2[#3]{\endgroup
531     #2[#3]}%
532   \ifx#1\@undefined
533     \xdef#1{#3}%
534   \fi
535   \ifx#1\relax
536     \xdef#1{#3}%
537   \fi
538 }%
539 \fi
540 \expandafter\x\csname ver@luatex-loader.sty\endcsname
541 \ProvidesPackage{luatex-loader}%
542 [2010/03/09 v0.4 Lua module loader (H0)]%
543 \begingroup\catcode61\catcode48\catcode32=10\relax%

```

```

544 \catcode13=5\endlinechar=13\relax%
545 \catcode10=12 % ^^J
546 \catcode34=12 % "
547 \catcode39=12 % '
548 \catcode40=12 % (
549 \catcode41=12 % )
550 \catcode44=12 % ,
551 \catcode46=12 % .
552 \catcode60=12 % <
553 \catcode61=12 % =
554 \catcode95=12 % _ (other!)
555 \catcode96=12 % '
556 \catcode123=1 % {
557 \catcode125=2 % }
558 \endlinechar=10 %
559 \ifnum\luatexversion<36 %
560   \directlua0%
561 \else %
562   \expandafter\directlua %
563 \fi %
564 {%
565   do
566     local script = "oberdiek.luatex.lua"
567     local file = kpse.find_file(script, "texmfscripts")
568     if file then
569       texio.write_nl("(" .. file .. ")")
570       dofile(file)
571     else
572       error("File " .. script .. " not found")
573     end
574   end
575 }%
576 \endgroup\endinput%
577 </loader>

```

## 2.11 Lua script

Currently LuaTeX does not use KPSE when searching for module files. The following Lua script implements a workaround. It extends `package.loader` by another search method. Modules are found by the module name with extension `.lua` similar to

```
kpsewhich --format=texmfscripts <module>.lua
```

Unhappily `kpsewhich` does not support directory components in the file name. Therefore a module `a.b.c` cannot be installed as `a/b/c.lua`. The script must be named `a.b.c.lua`.

```

578 <*lua>

579 module("oberdiek.luatex", package.seeall)

580 function kpse_module_loader(module)
581   local script = module .. ".lua"
582   local file = kpse.find_file(script, "texmfscripts")
583   if file then
584     local loader, error = loadfile(file)
585     if loader then
586       texio.write_nl("(" .. file .. ")")
587       return loader
588     end
589     return "\n\t[oberdiek.luatex.kpse_module_loader] Loading error:\n\t"
590       .. error
591   end
592   return "\n\t[oberdiek.luatex.kpse_module_loader] Search failed"

```

```

593 end
594 table.insert(package.loaders, kpse_module_loader)
595 

```

### 3 Test

```

596 <*test2>
597 \documentclass{article}
598 \def\LoadCommand{%
599   \RequirePackage{luatex}[2010/03/09]%
600 }
601 </test2>

602 <*test3>
603 \documentclass{article}
604 \def\LoadCommand{%
605   \RequirePackage{luatex-loader}[2010/03/09]%
606 }
607 </test3>

```

#### 3.1 Catcode checks for loading

```

608 <*test1>
609 \catcode'\{=1 %
610 \catcode'\}=2 %
611 \catcode'\#=6 %
612 \catcode'\@=11 %
613 \expandafter\ifx\csname count@\endcsname\relax
614   \countdef\count@=255 %
615 \fi
616 \expandafter\ifx\csname @gobble\endcsname\relax
617   \long\def\@gobble#1{}%
618 \fi
619 \expandafter\ifx\csname @firstofone\endcsname\relax
620   \long\def\@firstofone#1{#1}%
621 \fi
622 \expandafter\ifx\csname loop\endcsname\relax
623   \expandafter\@firstofone
624 \else
625   \expandafter\@gobble
626 \fi
627 {%
628   \def\loop#1\repeat{%
629     \def\body{#1}%
630     \iterate
631   }%
632   \def\iterate{%
633     \body
634     \let\next\iterate
635   \else
636     \let\next\relax
637   \fi
638   \next
639 }%
640 \let\repeat=\fi
641 }%
642 \def\RestoreCatcodes{}
643 \count@=0 %
644 \loop
645   \edef\RestoreCatcodes{%
646     \RestoreCatcodes
647     \catcode\the\count@=\the\catcode\count@\relax

```

```

648 }%
649 \ifnum\count@<255 %
650   \advance\count@ 1 %
651 \repeat
652
653 \def\RangeCatcodeInvalid#1#2{%
654   \count@=#1\relax
655   \loop
656     \catcode\count@=15 %
657   \ifnum\count@<#2\relax
658     \advance\count@ 1 %
659   \repeat
660 }
661 \def\RangeCatcodeCheck#1#2#3{%
662   \count@=#1\relax
663   \loop
664     \ifnum#3=\catcode\count@
665     \else
666       \errmessage{%
667         Character \the\count@\space
668         with wrong catcode \the\catcode\count@\space
669         instead of \number#3%
670       }%
671     \fi
672   \ifnum\count@<#2\relax
673     \advance\count@ 1 %
674   \repeat
675 }
676 \def\space{ }
677 \expandafter\ifx\csname LoadCommand\endcsname\relax
678   \def\LoadCommand{\input luatex.sty\relax}%
679 \fi
680 \def\Test{%
681   \RangeCatcodeInvalid{0}{47}%
682   \RangeCatcodeInvalid{58}{64}%
683   \RangeCatcodeInvalid{91}{96}%
684   \RangeCatcodeInvalid{123}{255}%
685   \catcode'\@=12 %
686   \catcode'\=0 %
687   \catcode'\%=14 %
688   \LoadCommand
689   \RangeCatcodeCheck{0}{36}{15}%
690   \RangeCatcodeCheck{37}{37}{14}%
691   \RangeCatcodeCheck{38}{47}{15}%
692   \RangeCatcodeCheck{48}{57}{12}%
693   \RangeCatcodeCheck{58}{63}{15}%
694   \RangeCatcodeCheck{64}{64}{12}%
695   \RangeCatcodeCheck{65}{90}{11}%
696   \RangeCatcodeCheck{91}{91}{15}%
697   \RangeCatcodeCheck{92}{92}{0}%
698   \RangeCatcodeCheck{93}{96}{15}%
699   \RangeCatcodeCheck{97}{122}{11}%
700   \RangeCatcodeCheck{123}{255}{15}%
701   \RestoreCatcodes
702 }
703 \Test
704 \csname @@end\endcsname
705 \end
706 </test1>

```

## 3.2 Catcode tables

### 3.2.1 Predefined catcode tables

```
707 (*test4)
708 \NeedsTeXFormat{LaTeX2e}

Remember LATEX's initial catcodes in count registers starting at \TestLaTeX.
709 \count0=0 %
710 \chardef\TestLaTeX=1000 %
711 \chardef\TestMax=300 %
712 \loop
713   \count\numexpr\TestLaTeX+\count0\relax=\catcode\count0 %
714   \ifnum\count0<\TestMax
715     \advance\count0 by 1 %
716   \repeat
717 \documentclass{minimal}
718 \usepackage{luatex}[2010/03/09]
719 \usepackage{qstest}
720 \IncludeTests{*}
721 \LogTests{log}{*}{*}
722 \makeatletter
723 \def\Check#1{%
724   \Expect*{\the\count@=\the\catcode\count@}%
725   *{\the\count@=#1}%
726 }
727 \newcount\scratch
728 \def\Test#1#2{%
729   \begin{qstest}{CatcodeTable#1}{CatcodeTable#1}%
730     \luatexcatcodetable\cename{CatcodeTable#1}\endcename
731     \count@=\z@
732     \loop
733       \scratch=#2\relax
734       \Expect*{\the\count@=\the\catcode\count@}%
735       *{\the\count@=\the\scratch}%
736       \ifnum\count@<\TestMax
737         \advance\count@\@ne
738       \repeat
739   \end{qstest}%
740 }
741 \begingroup
742   % luatex-unicode-letters.tex makes some slots to letters
743   \def\TestMax{169}%
744   \Test{LaTeX}{\the\count\numexpr\TestLaTeX+\count@}%
745 \endgroup
746 \Test{String}{\ifnum\count@=32 10\else 12\fi}
747 \Test{Other}{12}
748 \luatexinitcatcodetable99 %
749 \Test{IniTeX}{%
750   0\relax
751   \begingroup
752     \luatexcatcodetable99 %
753     \global\scratch=\the\catcode\count@
754   \endgroup
755 }
```

### 3.2.2 Catcode table number stack

```
756 \begin{qstest}{CatcodeTableNumStack}{CatcodeTableNumStack}
757   \def\TestStack#1{%
758     \Expect*{\LuT@NumStack}{#1}%
759   }%
760   \TestStack{0}%
761   \PushCatcodeTableNumStack
762   \TestStack{{0}0}%

```

```

763 \@firstofone{%
764   \begin{group}
765     \luatexinitcatcodetable12 %
766     \luatexcatcodetable12 %
767     \PushCatcodeTableNumStack
768     \TestStack{{12}{0}}%
769     \PopCatcodeTableNumStack
770     \TestStack{{0}{0}}%
771     \PopCatcodeTableNumStack
772     \TestStack{0}%
773     \def\TestWarning{Missing empty stack warning}%
774     \def\@PackageWarning#1#2{\def\TestWarning{empty stack}}%
775     \PopCatcodeTableNumStack
776     \TestStack{0}%
777     \Expect*{\TestWarning}{empty stack}%
778   \end{group}
779 }%
780 \end{qstest}

```

### 3.2.3 Catcode table stack

```

781 \begin{qstest}{CatcodeTableStack}{CatcodeTableStack}
782 \def\TestStack#1{%
783   \Expect*{\the\CatcodeTableStack}{#1}%
784 }%
785 \TestStack{0}%
786 \IncCatcodeTableStack
787 \TestStack{2}%
788 \IncCatcodeTableStack
789 \TestStack{4}%
790 \begin{group}
791   \IncCatcodeTableStack
792   \TestStack{6}%
793 \end{group}
794 \TestStack{6}%
795 \begin{group}
796   \DecCatcodeTableStack
797   \TestStack{4}%
798 \end{group}
799 \TestStack{4}%
800 \DecCatcodeTableStack
801 \TestStack{2}%
802 \DecCatcodeTableStack
803 \TestStack{0}%
804 \begin{group}
805   \def\TestError{Missing error}%
806   \def\@PackageError#1#2#3{%
807     \def\TestError{Empty stack}%
808   }%
809   \DecCatcodeTableStack
810   \TestStack{0}%
811   \Expect*{\TestError}{Empty stack}%
812 \end{group}
813 \end{qstest}

```

### 3.2.4 Catcode regime macros

```

814 \begin{qstest}{CatcodeRegime}{CatcodeRegime}
815 \def\TestStacks#1#2#3{%
816   \Expect*{\the\luatexcatcodetable}{#1}%
817   \Expect*{\the\CatcodeTableStack}{#2}%
818   \Expect*{\LuT@NumStack}{#3}%
819 }%
820 \TestStacks{0}{0}{0}%
821 \catcode'\|=7 %

```

```

822 \BeginCatcodeRegime\CatcodeTableLaTeX
823 \TestStacks{2}{2}{0}%
824 \Expect*{\the\catcode'\|}{12}%
825 \EndCatcodeRegime
826 \TestStacks{0}{0}{0}%
827 \Expect*{\the\catcode'\|}{7}%
828 \end{qstest}

```

### 3.3 Attribute allocation

```

829 \begin{qstest}{Attributes}{Attributes}
830 \newattribute\TestAttr
831 \Expect*{\meaning\TestAttr}%
832      *{\string\attribute\number\allocationnumber}%
833 \Expect*{\the\allocationnumber}{0}%
834 \begingroup
835   \newattribute\TestAttr
836   \Expect*{\the\allocationnumber}{1}%
837 \endgroup
838 \Expect*{\the\allocationnumber}{0}%
839 \Expect*{\meaning\TestAttr}*{\string\attribute1}%
840 \Expect*{\the\TestAttr}*{\number\LuT@UnsetAttributeValue}%
841 \def\Test#1{%
842   \setattribute\TestAttr{#1}%
843   \Expect*{\the\TestAttr}{#1}%
844 }%
845 \Test{0}%
846 \Test{1}%
847 \Test{-1}%
848 \Test{123}%
849 \unsetattribute\TestAttr
850 \Expect*{\the\TestAttr}*{\number\LuT@UnsetAttributeValue}%
851 \begingroup
852   \Expect*{\the\TestAttr}*{\number\LuT@UnsetAttributeValue}%
853   \Test{1234}%
854 \endgroup
855 \Expect*{\the\TestAttr}*{\number\LuT@UnsetAttributeValue}%
856 \end{qstest}

857 \@@@end
858 \end{test4}

```

### 3.4 Short test for plain T<sub>E</sub>X

```

859 (*test5)
860 \input luatex.sty\relax
861 \newattribute\TestAttr
862 \setattribute\TestAttr{10}
863 \unsetattribute\TestAttr
864 \newcatcodetable\TestCTa
865 \begingroup
866   \SetCatcodeRange{'A'}{'Z'}{12}%
867 \endgroup
868 \BeginCatcodeRegime\CatcodeTableLaTeX
869 \EndCatcodeRegime
870 \end
871 \end{test5}

```

## 4 Installation

### 4.1 Download

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

---

<sup>1</sup><http://ftp.ctan.org/tex-archive/>

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

[CTAN:macros/latex/contrib/oberdiek/luatex.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 luatex.dtx
```

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

<code>luatex.sty</code>	→ <code>tex/generic/oberdiek/luatex.sty</code>
<code>luatex-loader.sty</code>	→ <code>tex/generic/oberdiek/luatex-loader.sty</code>
<code>oberdiek.luatex.lua</code>	→ <code>scripts/oberdiek/oberdiek.luatex.lua</code>
<code>luatex.pdf</code>	→ <code>doc/latex/oberdiek/luatex.pdf</code>
<code>test/luatex-test1.tex</code>	→ <code>doc/latex/oberdiek/test/luatex-test1.tex</code>
<code>test/luatex-test2.tex</code>	→ <code>doc/latex/oberdiek/test/luatex-test2.tex</code>
<code>test/luatex-test3.tex</code>	→ <code>doc/latex/oberdiek/test/luatex-test3.tex</code>
<code>test/luatex-test4.tex</code>	→ <code>doc/latex/oberdiek/test/luatex-test4.tex</code>
<code>test/luatex-test5.tex</code>	→ <code>doc/latex/oberdiek/test/luatex-test5.tex</code>
<code>luatex.dtx</code>	→ <code>source/latex/oberdiek/luatex.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`.

## 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 luatex.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{luatex.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 luatex.dtx
makeindex -s gind.ist luatex.idx
pdflatex luatex.dtx
makeindex -s gind.ist luatex.idx
pdflatex luatex.dtx
```

## 5 History

[2007/12/12 v0.1]

- First public version.

[2009/04/10 v0.2]

- Requires package `ifluatex` in version 2.0 to ensure `\luatexversion`.
- Updates the call of `\directlua`, the syntax has changed in  $\text{\LuaTeX}$  0.36.

[2009/12/02 v0.3]

- Unsetting of attributes updated for  $\text{\LuaTeX}$  0.37.

[2010/03/09 v0.4]

- Support for lua states removed.
- Calling `tex.enableprimitives` for used primitives.

## 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	
<code>\#</code> .....	611
<code>\%</code> .....	687
<code>\:</code> .....	277, 278, 279, 280, 281, 284
<code>\@</code> .....	612, 685
<code>\@end</code> .....	857
<code>\@INTERNAL@ERROR</code> .....	316
<code>\@PackageError</code> .....	130, 169, 194, 300, 369, 378, 806
<code>\@PackageWarning</code> .....	447, 774
<code>\@PackageWarningNoLine</code> .....	216
<code>\@car</code> .....	249, 450
<code>\@cdr</code> .....	252, 452
<code>\@ehc</code> .....	132, 171, 196, 303
<code>\@ehd</code> .....	371, 380
<code>\@empty</code> .....	237
<code>\@firstofone</code> ..	243, 407, 620, 623, 763
<code>\@firstoftwo</code> .....	246, 257
<code>\@gobble</code> .....	240, 617, 625
<code>\@ifnch</code> .....	263, 265, 282
<code>\@ifnextchar</code> .....	257, 259
<code>\@ifstar</code> .....	255
<code>\@let@token</code> .....	263, 266, 269, 282
<code>\@makeoother</code> .....	406, 426, 427, 428, 429, 430, 434
<code>\@ne</code> .....	327, 396, 737
<code>\@nil</code> .....	250, 253, 450, 452
<code>\@sptoken</code> .....	266, 277
<code>\@tempcnta</code> .....	286, 385, 390, 394, 395, 396
<code>\@tempcntb</code> .....	289, 386, 391, 394
<code>\@undefined</code> ....	58, 155, 180, 295, 532
<code>\@xifnch</code> .....	267, 280
<code>\@</code> .....	686
<code>\{</code> .....	609
<code>\}</code> .....	610
<code>\ </code> .....	821, 824, 827
<code>\advance</code> .....	327, 355, 367, 376, 396, 650, 658, 673, 715, 737
<code>\aftergroup</code> .....	29, 503
<code>\allocationnumber</code> ..	328, 329, 331, 356, 357, 359, 832, 833, 836, 838
<code>\attribute</code> .....	331, 333, 832, 839
A	
<code>\begin</code> .....	729, 756, 781, 814, 829
<code>\BeginCatcodeRegime</code> ..	5, 456, 822, 868
<code>\body</code> .....	629, 633
B	
<code>\catcode</code> .....	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, 395, 406, 411, 412, 413, 415, 416, 417, 418, 419, 420, 421, 422, 423, 476, 477, 479, 480, 481, 482, 483, 484, 485, 486, 487, 507, 508, 510, 511, 512, 513, 514, 515, 516, 517, 518, 519, 520, 521, 522, 523, 543, 544, 545, 546, 547, 548, 549, 550, 551, 552, 553, 554, 555, 556, 557, 609, 610, 611, 612, 647, 656, 664, 668, 685, 686, 687, 713, 724, 734, 753, 821, 824, 827
<code>\catcodetable</code> .....	359, 362
<code>\CatcodeTableIniTeX</code> ..	4, 400, 404, 408
<code>\CatcodeTableLaTeX</code> ..	403, 424, 822, 868
<code>\CatcodeTableOther</code> .....	402, 435
<code>\CatcodeTableStack</code> ..	4, 350, 351, 366, 367, 375, 376, 460, 461, 783, 817
<code>\CatcodeTableString</code> .....	401, 433
<code>\chardef</code> .....	353, 357, 710, 711
<code>\check</code> .....	723
<code>\count</code> .....	221, 222, 709, 713, 714, 715, 744
<code>\count@</code> .....	387, 392, 395, 614, 643, 647, 649, 650, 654, 656, 657, 658, 662, 664, 667, 668, 672, 673, 724, 725, 731, 734, 735, 736, 737, 744, 746, 753
<code>\countdef</code> .....	614
<code>\csname</code> .....	14, 21, 50, 66, 76, 121, 137, 138, 143, 145, 148, 149, 153, 155, 163, 166, 167, 178, 180, 188, 191, 192, 210, 215, 237, 240, 243, 246, 249, 252, 255, 286, 287, 289, 290, 293, 468, 488, 495, 524, 540, 613, 616, 619, 622, 677, 704, 730
C	
<code>\DecCatcodeTableStack</code> .....	374, 464, 796, 800, 802, 809
<code>\directlua</code> .....	157, 182, 560, 562
<code>\documentclass</code> .....	597, 603, 717
D	
<code>\empty</code> .....	17, 18, 491, 492
<code>\end</code> ..	705, 739, 780, 813, 828, 856, 870
<code>\EndCatcodeRegime</code> ....	463, 825, 869
<code>\endcsname</code> .....	14, 21, 50, 66, 76, 121, 137, 138, 143, 145, 148, 149, 153, 155, 163, 166, 167, 178, 180, 188, 191, 192, 210, 215, 237, 240, 243, 246, 249, 252, 255, 286, 287, 289, 290, 293, 468, 488, 495, 524, 540, 613, 616, 619, 622, 677, 704, 730
<code>\endinput</code> .....	29, 119, 503, 576
E	

<code>\endlinechar</code> . . . . .	4, 35, 71, 77, 89, 478, 509, 544, 558	<code>\LuT@UnsetAttributeValue</code> . . .	340, 341, 343, 346, 840, 850, 852, 855
<code>\errmessage</code> . . . . .	333, 362, 666	<b>M</b>	
<code>\Expect</code> . . . . .	724, 734, 758, 777, 783, 811, 816, 817, 818, 824, 827, 831, 833, 836, 838, 839, 840, 843, 850, 852, 855	<code>\m@ne</code> . . . . .	324, 341, 349
<b>F</b>		<code>\makeatletter</code> . . . . .	722
<code>\futurelet</code> . . . . .	263, 282	<code>\meaning</code> . . . . .	831, 839
<b>G</b>		<code>\MessageBreak</code> . . . . .	301
<code>\globbox</code> . . . . .	235	<b>N</b>	
<code>\globcount</code> . . . . .	232	<code>\n</code> . . . . .	589, 592
<code>\globdimen</code> . . . . .	233	<code>\NeedsTeXFormat</code> . . . . .	708
<code>\globskip</code> . . . . .	234	<code>\newattribute</code> . . .	3, 325, 830, 835, 861
<b>I</b>		<code>\newbox</code> . . . . .	235
<code>\ifcase</code> . . . . .	307	<code>\newcatcodetable</code> . . . . .	4, 352, 400, 401, 402, 403, 864
<code>\ifluatex</code> . . . . .	128	<code>\newcommand</code> . . . . .	321
<code>\ifnum</code> . . . .	135, 156, 181, 221, 326, 339, 353, 366, 375, 394, 559, 649, 657, 664, 672, 714, 736, 746	<code>\newcount</code> . . . .	232, 323, 348, 350, 727
<code>\ifx</code> . . . . .	15, 18, 21, 50, 58, 61, 121, 143, 145, 153, 163, 178, 188, 210, 215, 237, 240, 243, 246, 249, 252, 255, 266, 269, 286, 289, 293, 295, 298, 306, 446, 468, 489, 492, 495, 524, 532, 535, 613, 616, 619, 622, 677	<code>\newdimen</code> . . . . .	233
<code>\immediate</code> . . . . .	23, 52, 497, 526	<code>\newskip</code> . . . . .	234
<code>\IncCatcodeTableStack</code> . . . . .	365, 459, 786, 788, 791	<code>\next</code> . . . . .	634, 636, 638
<code>\IncludeTests</code> . . . . .	720	<code>\number</code> . . .	669, 832, 840, 850, 852, 855
<code>\input</code> . . . . .	122, 123, 469, 678, 860	<code>\numexpr</code> 337, 390, 391, 392, 458, 713, 744	
<code>\iterate</code> . . . . .	630, 632, 634	<b>P</b>	
<b>L</b>		<code>\PackageInfo</code> . . . . .	26, 500
<code>\LoadCommand</code> . . . . .	598, 604, 678, 688	<code>\PopCatcodeTableNumStack</code> . . . . .	445, 465, 769, 771, 775
<code>\LogTests</code> . . . . .	721	<code>\ProvidesPackage</code> . . . .	19, 67, 493, 541
<code>\loop</code> .	393, 628, 644, 655, 663, 712, 732	<code>\PushCatcodeTableNumStack</code> . . . . .	5, 440, 457, 761, 767
<code>\luatexattributedef</code> . . . . .	329	<b>R</b>	
<code>\luatexcatcodetable</code> . . . . .	408, 442, 448, 450, 458, 461, 730, 752, 766, 816	<code>\RangeCatcodeCheck</code> . . . . .	661, 689, 690, 691, 692, 693, 694, 695, 696, 697, 698, 699, 700
<code>\luatexinitcatcodetable</code> . . . . .	404, 748, 765	<code>\RangeCatcodeInvalid</code> . . . . .	653, 681, 682, 683, 684
<code>\luatexsavecatcodetable</code> . . . . .	424, 433, 435, 460	<code>\repeat</code> . . . . .	397, 628, 640, 651, 659, 674, 716, 738
<code>\luatexversion</code> . . . . .	135, 339, 559	<code>\RequirePackage</code> . . . . .	125, 126, 212, 471, 599, 605
<code>\LuT@AllocAttribute</code> . . . . .	323, 326, 327, 328	<code>\reserved@a</code> . . . . .	261, 270
<code>\LuT@AllocCatcodeTable</code> . . . . .	348, 353, 355, 356	<code>\reserved@b</code> . . . . .	262, 272
<code>\LuT@AtEnd</code> . . . .	95, 96, 119, 133, 473	<code>\reserved@c</code> . . . . .	267, 270, 272, 275
<code>\LuT@MakeLuatexPrimitive</code> . . . . .	141, 202, 203, 204, 205, 206, 207	<code>\reserved@d</code> . . . . .	260, 269
<code>\LuT@MakePrimitive</code> . . . .	136, 176, 208	<code>\RestoreCatcodes</code> . .	642, 645, 646, 701
<code>\LuT@newcommand</code> . . . . .	292, 325, 336, 340, 343, 345, 352, 365, 374, 383, 440, 445, 456, 463	<b>S</b>	
<code>\LuT@NumStack</code> . . . . .	439, 441, 442, 446, 450, 451, 452, 758, 818	<code>\scratch</code> . . . . .	727, 733, 735, 753
<code>\LuT@NumStackEmpty</code> . . . .	438, 439, 446	<code>\setattribute</code> . . . . .	3, 336, 842, 862
<code>\LuT@temp</code> .	220, 225, 226, 227, 228, 229, 230, 231, 277, 284, 384, 398	<code>\SetCatcodeRange</code> . . . . .	5, 383, 410, 414, 431, 432, 866
		<code>\space</code> . . . . .	667, 668, 676
		<b>T</b>	
		<code>\t</code> . . . . .	589, 592
		<code>\Test</code> . .	680, 703, 728, 744, 746, 747, 749, 841, 845, 846, 847, 848, 853
		<code>\TestAttr</code> . . . . .	830, 831, 835, 839, 840, 842, 843, 849, 850, 852, 855, 861, 862, 863
		<code>\TestCTa</code> . . . . .	864

<code>\TestError</code> .....	805, 807, 811	<code>\tw@</code> .....	355, 367
<code>\TestLaTeX</code> .....	710, 713, 744		
<code>\TestMax</code> .....	711, 714, 736, 743	<b>U</b>	
<code>\TestStack</code> .....	757, 760, 762, 768, 770, 772, 776, 782, 785, 787, 789, 792, 794, 797, 799, 801, 803, 810	<code>\unless</code> .....	394
<code>\TestStacks</code> .....	815, 820, 823, 826	<code>\unsetattribute</code> .	3, 330, <u>339</u> , 849, 863
<code>\TestWarning</code> .....	773, 774, 777	<code>\usepackage</code> .....	718, 719
<code>\the</code> .....	77, 78, 79, 80, 81, 82, 83, 84, 97, 331, 359, 385, 386, 387, 442, 647, 667, 668, 724, 725, 734, 735, 744, 753, 783, 816, 817, 824, 827, 833, 836, 838, 840, 843, 850, 852, 855	<b>W</b>	
<code>\TMP@EnsureCode</code> .....	.. 94, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118	<code>\wlog</code> .....	331, 358
		<code>\write</code> .....	23, 52, 497, 526
		<b>X</b>	
		<code>\x</code> .....	14, 15, 18, 22, 26, 28, 51, 56, 66, 75, 87, 488, 489, 492, 496, 500, 502, 525, 530, 540
		<b>Z</b>	
		<code>\z@</code> .....	351, 375, 448, 731