

# 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	9
2.7	plain TeX compatibility	10
2.8	Attributes	12
2.8.1	Allocation	12
2.8.2	Interface	12
2.9	Catcode tables	12
2.9.1	Allocation	12
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>17</b>
3.1	Catcode checks for loading . . . . .	17
3.2	Catcode tables . . . . .	19
3.2.1	Predefined catcode tables . . . . .	19
3.2.2	Catcode table number stack . . . . .	20
3.2.3	Catcode table stack . . . . .	20
3.2.4	Catcode regime macros . . . . .	21
3.3	Attribute allocation . . . . .	21
3.4	Short test for plain T <sub>E</sub> X . . . . .	21
<b>4</b>	<b>Installation</b>	<b>22</b>
4.1	Download . . . . .	22
4.2	Bundle installation . . . . .	22
4.3	Package installation . . . . .	22
4.4	Refresh file name databases . . . . .	23
4.5	Some details for the interested . . . . .	23
<b>5</b>	<b>History</b>	<b>23</b>
[2007/12/12 v0.1]	. . . . .	23
[2009/04/10 v0.2]	. . . . .	23
[2009/12/02 v0.3]	. . . . .	23
[2010/03/09 v0.4]	. . . . .	23
<b>6</b>	<b>Index</b>	<b>24</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.

<code>\PushCatcodeTableNumStack</code> <code>\PopCatcodeTableNumStack</code>
---

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.

<code>\BeginCatcodeRegime {<i>&lt;catcodetable&gt;</i>}</code> <code>\EndCatcodeRegime</code>
--

`\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.

<code>\SetCatcodeRange {<i>&lt;from&gt;</i>} {<i>&lt;to&gt;</i>} {<i>&lt;catcode&gt;</i>}</code>
--

The catcodes of characters in range from *<from>* to inclusive *<to>* are set to *<catcode>*.

## 1.5 Lua module loading

Currently Lua $\text{\TeX}$  (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
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@luatex.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{luatex}{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@luatex.sty\endcsname
61 \ProvidesPackage{luatex}%
62 [2010/03/09 v0.4 LuaTeX basic definition package (H0)]

```

## 2.2 Catcodes

```

63 \begingroup
64 \catcode123 1 % {
65 \catcode125 2 % }
66 \def\x{\endgroup
67   \expandafter\edef\csname LuT@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\LuT@AtEnd{%
81     \LuT@AtEnd
82     \catcode#1 \the\catcode#1\relax
83   }%
84   \catcode#1 #2\relax
85 }
86 \TMP@EnsureCode{10}{12}% ^^J
87 \TMP@EnsureCode{34}{12}% "
88 \TMP@EnsureCode{36}{3}% $
89 \TMP@EnsureCode{39}{12}% '
90 \TMP@EnsureCode{40}{12}% (
91 \TMP@EnsureCode{41}{12}% )
92 \TMP@EnsureCode{42}{12}% *
93 \TMP@EnsureCode{43}{12}% +
94 \TMP@EnsureCode{44}{12}% ,
95 \TMP@EnsureCode{45}{12}% -
96 \TMP@EnsureCode{46}{12}% .
97 \TMP@EnsureCode{47}{12}% /
98 \TMP@EnsureCode{60}{12}% <

```

```

99 \TMP@EnsureCode{61}{12}% =
100 \TMP@EnsureCode{62}{12}% >
101 \TMP@EnsureCode{95}{12}% _ (other!)
102 \TMP@EnsureCode{96}{12}% '

```

## 2.3 Check for LuaTeX

Without LuaTeX there is no point in using this package.

```

103 \begingroup\expandafter\expandafter\expandafter\endgroup
104 \expandafter\ifx\csname RequirePackage\endcsname\relax
105   \input infwarerr.sty\relax
106   \input ifluatex.sty\relax
107 \else
108   \RequirePackage{infwarerr}[2007/09/09]%
109   \RequirePackage{ifluatex}[2009/04/10]%
110 \fi

111 \ifluatex
112 \else
113   \@PackageError{luatex}{%
114     This package may only be run using LuaTeX%
115   }\@ehc
116   \LuT@AtEnd
117   \expandafter\endinput
118 \fi

```

## 2.4 Provide LuaTeX primitives

```

119 \ifnum\luatexversion<36 %
120   \def\LuT@MakePrimitive#1{%
121     \expandafter\let\csname luatex#1\expandafter\endcsname
122     \csname #1\endcsname
123   }%
124 \else
125   \def\LuT@MakeLuatexPrimitive#1{%
126     \begingroup\expandafter\expandafter\expandafter\endgroup
127     \expandafter\ifx\csname luatex#1\endcsname\relax
128       \begingroup\expandafter\expandafter\expandafter\endgroup
129       \expandafter\ifx\csname #1\endcsname\relax
130         \else
131           \expandafter\let
132           \csname luatex#1\expandafter\endcsname
133           \csname #1\endcsname
134         \fi
135       \fi
136     \begingroup\expandafter\expandafter\expandafter\endgroup
137     \expandafter\ifx\csname luatex#1\endcsname\relax
138       \begingroup
139         \expandafter\let\csname luatex#1\endcsname\@undefined
140         \ifnum0%
141           \directlua{%
142             if tex.enableprimitives then %
143               tex.enableprimitives('luatex',{#1'})%
144               tex.print('1')%
145             end%
146           }%
147         \expandafter\ifx\csname luatex#1\endcsname\relax\else1\fi
148       =11 %
149       \global\expandafter\let
150       \csname luatex#1\expandafter\endcsname
151       \csname luatex#1\endcsname
152     \else
153       \@PackageError{luatex}{%

```



```

154         tex.enableprimitives failed for '#1'%
155     }\@ehc
156 \fi
157 \endgroup
158 \fi
159 }%
160 \def\LuT@MakePrimitive#1{%
161     \begingroup\expandafter\expandafter\expandafter\endgroup
162     \expandafter\ifx\csname#1\endcsname\relax
163         \begingroup
164             \expandafter\let\csname#1\endcsname\@undefined
165             \ifnum0%
166                 \directlua{%
167                     if tex.enableprimitives then %
168                         tex.enableprimitives('',{ '#1' })%
169                         tex.print('1')%
170                     end%
171                 }%
172             \expandafter\ifx\csname#1\endcsname\relax\else1\fi
173             =11 %
174             \global\expandafter\let
175             \csname#1\expandafter\endcsname
176             \csname#1\endcsname
177         \else
178             \@PackageError{luatex}{%
179                 tex.enableprimitives failed for '#1'%
180             }\@ehc
181         \fi
182     \endgroup
183 \fi
184 }%
185 \fi
186 \LuT@MakeLuatexPrimitive{attribute}
187 \LuT@MakeLuatexPrimitive{attributedef}
188 \LuT@MakeLuatexPrimitive{catcodetable}
189 \LuT@MakeLuatexPrimitive{initcatcodetable}
190 \LuT@MakeLuatexPrimitive{luaescapestring}
191 \LuT@MakeLuatexPrimitive{savecatcodetable}
192 \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  $\LaTeX$  and tests its presence independently of the format by looking for `\et@xins`.

```

193 \begingroup\expandafter\expandafter\expandafter\endgroup
194 \expandafter\ifx\csname RequirePackage\endcsname\relax
195 \else
196     \RequirePackage{etex}[1998/03/26]%
197 \fi

```

## 2.6 Adaption of $\epsilon$ -TeX's register allocation

$\epsilon$ -TeX has increased the number of TeX registers from  $2^8$  (256) to  $2^{15}$  (32768) for a register class. LuaTeX 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.

```

198 \begingroup\expandafter\expandafter\expandafter\endgroup
199 \expandafter\ifx\csname et@xins\endcsname\relax
200     \@PackageWarningNoLine{luatex}{%
201         Support for eTeX is not loaded (etex.src)%

```

```

202 }%
203 \else
204   \def\LuT@temp#1{%
205     \ifnum\count27#1=32768 %
206       \count27#1=65536 %
207     \fi
208   }%
209   \LuT@temp0%
210   \LuT@temp1%
211   \LuT@temp2%
212   \LuT@temp3%
213   \LuT@temp4%
214   \LuT@temp5%
215   \LuT@temp6%

```

$\varepsilon$ -TeX uses an array for the first 256 registers and then a tree structure. LuaTeX 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.

```

216 \let\newcount\globcount
217 \let\newdimen\globdimen
218 \let\newskip\globskip
219 \let\newbox\globbox
220 \fi

```

## 2.7 plain TeX compatibility

\@empty

```

221 \expandafter\ifx\csname @empty\endcsname\relax
222   \def\@empty{}%
223 \fi

```

\@gobble

```

224 \expandafter\ifx\csname @gobble\endcsname\relax
225   \long\def\@gobble#1{}%
226 \fi

```

\@firstofone

```

227 \expandafter\ifx\csname @firstofone\endcsname\relax
228   \long\def\@firstofone#1{#1}%
229 \fi

```

\@firstoftwo

```

230 \expandafter\ifx\csname @firstoftwo\endcsname\relax
231   \long\def\@firstoftwo#1#2{#1}%
232 \fi

```

\@car

```

233 \expandafter\ifx\csname @car\endcsname\relax
234   \def\@car#1#2\@nil{#1}%
235 \fi

```

\@cdr

```

236 \expandafter\ifx\csname @cdr\endcsname\relax
237   \def\@cdr#1#2\@nil{#2}%
238 \fi

```

\@ifstar

```

239 \expandafter\ifx\csname @ifstar\endcsname\relax
240   \def\@ifstar#1{%
241     \@ifnextchar*{\@firstoftwo{#1}}%
242   }%

```

```

\@ifnextchar
243 \long\def\@ifnextchar#1#2#3{%
244 \let\reserved@d=#1%
245 \def\reserved@a{#2}%
246 \def\reserved@b{#3}%
247 \futurelet\@let@token\@ifnch
248 }%

\@ifnch
249 \def\@ifnch{%
250 \ifx\@let@token\@sptoken
251 \let\reserved@c\@xifnch
252 \else
253 \ifx\@let@token\reserved@d
254 \let\reserved@c\reserved@a
255 \else
256 \let\reserved@c\reserved@b
257 \fi
258 \fi
259 \reserved@c
260 }%

\@sptoken
261 \let\LuT@temp\:%
262 \def\:{\let\@sptoken= }%
263 \: % explicit space

\@xifnch
264 \def\:\@xifnch}%
265 \expandafter\def\:{%
266 \futurelet\@let@token\@ifnch
267 }%
268 \let\:\LuT@temp
269 \fi

\@tempcnta
270 \expandafter\ifx\csname @tempcnta\endcsname\relax
271 \csname newcount\endcsname\@tempcnta
272 \fi

\@tempcntb
273 \expandafter\ifx\csname @tempcntb\endcsname\relax
274 \csname newcount\endcsname\@tempcntb
275 \fi

\LuT@newcommand
276 \begingroup\expandafter\expandafter\expandafter\endgroup
277 \expandafter\ifx\csname newcommand\endcsname\relax
278 \def\LuT@newcommand#1[#2]#3{%
279 \ifx#1\@undefined
280 \let#1\relax
281 \else
282 \ifx#1\relax
283 \else
284 \@PackageError{luatex}{%
285 \string#1 is already defined.\MessageBreak
286 Redefinition is skipped%
287 }\@ehc
288 \fi
289 \fi
290 \ifx#1\relax

```

```

291     \ifcase#2 %
292     \def#1{#3}%
293     \or
294     \def#1##1{#3}%
295     \or
296     \def#1##1##2{#3}%
297     \or
298     \def#1##1##2##3{#3}%
299     \or
300     \@INTERNAL@ERROR
301     \fi
302 \fi
303 }%
304 \else
305 \def\LuT@newcommand{\newcommand*}%
306 \fi

```

## 2.8 Attributes

### 2.8.1 Allocation

\LuT@AllocAttribute

```

307 \newcount\LuT@AllocAttribute
308 \LuT@AllocAttribute=\m@ne

```

\newattribute

```

309 \LuT@newcommand\newattribute[1]{%
310 \ifnum\LuT@AllocAttribute<65535 %
311 \global\advance\LuT@AllocAttribute\@ne
312 \allocationnumber\LuT@AllocAttribute
313 \global\luatexattributedef#1=\allocationnumber
314 \unsetattribute{#1}%
315 \wlog{\string#1=\string\attribute\the\allocationnumber}%
316 \else
317 \errmessage{No room for a new \string\attribute}%
318 \fi
319 }

```

### 2.8.2 Interface

\setattribute

```

320 \LuT@newcommand\setattribute[2]{%
321 #1=\numexpr#2\relax
322 }

```

\unsetattribute

```

323 \ifnum\luatexversion<37
324 \LuT@newcommand\LuT@UnsetAttributeValue[0]{}%
325 \let\LuT@UnsetAttributeValue\m@ne
326 \else
327 \LuT@newcommand\LuT@UnsetAttributeValue[0]{-2147483647}%
328 \fi
329 \LuT@newcommand\unsetattribute[1]{%
330 #1=\LuT@UnsetAttributeValue
331 }

```

## 2.9 Catcode tables

### 2.9.1 Allocation

\LuT@AllocCatcodeTable

```

332 \newcount\LuT@AllocCatcodeTable
333 \LuT@AllocCatcodeTable=\m@ne
334 \newcount\CatcodeTableStack
335 \CatcodeTableStack=\z@

\newcatcodetable

336 \LuT@newcommand\newcatcodetable[1]{%
337   \ifnum\LuT@AllocCatcodeTable<1114110 % 0x10FFFF is maximal \chardef
338     % or < 268435455 % 228 - 1
339   \global\advance\LuT@AllocCatcodeTable by\tw@
340   \allocationnumber=\LuT@AllocCatcodeTable
341   \global\chardef#1=\allocationnumber
342   \wlog{%
343     \string#1=\string\catcodetable\the\allocationnumber
344   }%
345   \else
346     \errmessage{No room for a new \string\catcodetable}%
347   \fi
348 }%
```

\IncCatcodeTableStack

```

349 \LuT@newcommand\IncCatcodeTableStack[0]{%
350   \ifnum\CatcodeTableStack<268435454 %
351   \global\advance\CatcodeTableStack by\tw@
352   \else
353     \@PackageError{luatex}{%
354       Catcode table stack overflow%
355     }\@ehd
356   \fi
357 }
```

\DecCatcodeTableStack

```

358 \LuT@newcommand\DecCatcodeTableStack[0]{%
359   \ifnum\CatcodeTableStack>\z@
360   \global\advance\CatcodeTableStack by-2 %
361   \else
362     \@PackageError{luatex}{%
363       Catcode table stack is empty%
364     }\@ehd
365   \fi
366 }
```

## 2.9.2 \SetCatcodeRange

\SetCatcodeRange

```

367 \LuT@newcommand\SetCatcodeRange[3]{%
368   \edef\LuT@temp{%
369     \noexpand\@tempcnta=\the\@tempcnta
370     \noexpand\@tempcntb=\the\@tempcntb
371     \noexpand\count@=\the\count@
372     \relax
373   }%
374   \@tempcnta=\numexpr#1\relax
375   \@tempcntb=\numexpr#2\relax
376   \count@=\numexpr#3\relax
377   \loop
378     \unless\ifnum\@tempcnta>\@tempcntb
379     \catcode\@tempcnta=\count@
380     \advance\@tempcnta by \@ne
381   \repeat
382   \LuT@temp
383 }
```

### 2.9.3 Predefined catcode tables

```

384 \newcatcodetable\CatcodeTableIniTeX
385 \newcatcodetable\CatcodeTableString
386 \newcatcodetable\CatcodeTableOther
387 \newcatcodetable\CatcodeTableLaTeX

388 \luatexinitcatcodetable\CatcodeTableIniTeX
389 \begingroup
390   \def\@makeother#1{\catcode#1=12\relax}%
391   \@firstofone{%
392     \luatexcatcodetable\CatcodeTableIniTeX
393     \begingroup
394       \SetCatcodeRange{0}{8}{15}%
395       \catcode9=10 % tab
396       \catcode11=15 %
397       \catcode12=13 % form feed
398       \SetCatcodeRange{14}{31}{15}%
399       \catcode35=6 % hash
400       \catcode36=3 % dollar
401       \catcode38=4 % ampersand
402       \catcode94=7 % circumflex
403       \catcode95=8 % underscore
404       \catcode123=1 % brace left
405       \catcode125=2 % brace right
406       \catcode126=13 % tilde
407       \catcode127=15 %
408       \luatexsavecatcodetable\CatcodeTableLaTeX
409     \endgroup
410     \@makeother{0}% nul
411     \@makeother{13}% carriage return
412     \@makeother{37}% percent
413     \@makeother{92}% backslash
414     \@makeother{127}%
415     \SetCatcodeRange{65}{90}{12}% A-Z
416     \SetCatcodeRange{97}{122}{12}% a-z
417     \luatexsavecatcodetable\CatcodeTableString
418     \@makeother{32}% space
419     \luatexsavecatcodetable\CatcodeTableOther
420   \endgroup
421 }%
```

### 2.9.4 Number stack

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

```
422 \def\LuT@NumStackEmpty{0}
```

`\LuT@NumStack`

```
423 \let\LuT@NumStack\LuT@NumStackEmpty
```

`\PushCatcodeTableNumStack`

```

424 \LuT@newcommand\PushCatcodeTableNumStack[0]{%
425   \xdef\LuT@NumStack{%
426     {\the\luatexcatcodetable}\LuT@NumStack
427   }%
428 }
```

`\PopCatcodeTableNumStack`

```

429 \LuT@newcommand\PopCatcodeTableNumStack[0]{%
430   \ifx\LuT@NumStack\LuT@NumStackEmpty
431     \@PackageWarning{luatex}{Empty catcode table number stack}%
432     \luatexcatcodetable\z@
433   \else
```

```

434 \luatexcatcodetable=\expandafter\@car\LuT@NumStack\@nil\relax
435 \xdef\LuT@NumStack{%
436 \expandafter\@cdr\LuT@NumStack\@nil
437 }%
438 \fi
439 }

```

### 2.9.5 Catcode regime macros

`\BeginCatcodeRegime`

```

440 \LuT@newcommand\BeginCatcodeRegime[1]{%
441 \PushCatcodeTableNumStack
442 \luatexcatcodetable=numexpr#1\relax
443 \IncCatcodeTableStack
444 \luatexsavecatcodetable\CatcodeTableStack
445 \luatexcatcodetable\CatcodeTableStack
446 }

```

`\EndCatcodeRegime`

```

447 \LuT@newcommand\EndCatcodeRegime[0]{%
448 \DecCatcodeTableStack
449 \PopCatcodeTableNumStack
450 }

```

### 2.10 Lua module loader

```

451 \begingroup\expandafter\expandafter\expandafter\endgroup
452 \expandafter\ifx\csname RequirePackage\endcsname\relax
453 \input luatex-loader.sty\relax
454 \else
455 \RequirePackage{luatex-loader}[2010/03/09]%
456 \fi

457 \LuT@AtEnd
458 </package>

459 <loader>

```

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

```

460 \begingroup
461 \catcode44 12 % ,
462 \catcode45 12 % -
463 \catcode46 12 % .
464 \catcode58 12 % :
465 \catcode64 11 % @
466 \catcode123 1 % {
467 \catcode125 2 % }
468 \expandafter\let\expandafter\x\csname ver@luatex-loader.sty\endcsname
469 \ifx\x\relax % plain-TeX, first loading
470 \else
471 \def\empty{%
472 \ifx\x\empty % LaTeX, first loading,
473 % variable is initialized, but \ProvidesPackage not yet seen
474 \else
475 \catcode35 6 % #
476 \expandafter\ifx\csname PackageInfo\endcsname\relax
477 \def\x#1#2{%
478 \immediate\write-1{Package #1 Info: #2.}%
479 }%
480 \else
481 \def\x#1#2{\PackageInfo{#1}{#2, stopped}}%
482 \fi
483 \x{luatex-loader}{The package is already loaded}%

```

```

484     \aftergroup\endinput
485     \fi
486     \fi
487 \endgroup
Package identification:
488 \begingroup
489   \catcode35 6 % #
490   \catcode40 12 % (
491   \catcode41 12 % )
492   \catcode44 12 % ,
493   \catcode45 12 % -
494   \catcode46 12 % .
495   \catcode47 12 % /
496   \catcode58 12 % :
497   \catcode64 11 % @
498   \catcode91 12 % [
499   \catcode93 12 % ]
500   \catcode123 1 % {
501   \catcode125 2 % }
502   \expandafter\ifx\csname ProvidesPackage\endcsname\relax
503     \def\x#1#2#3[#4]{\endgroup
504       \immediate\write-1{Package: #3 #4}%
505       \xdef#1{#4}%
506     }%
507   \else
508     \def\x#1#2[#3]{\endgroup
509       #2[#3]}%
510     \ifx#1@undefined
511       \xdef#1{#3}%
512     \fi
513     \ifx#1\relax
514       \xdef#1{#3}%
515     \fi
516   }%
517 \fi
518 \expandafter\x\csname ver@luatex-loader.sty\endcsname
519 \ProvidesPackage{luatex-loader}%
520 [2010/03/09 v0.4 Lua module loader (H0)]
521 \begingroup
522   \catcode10 12 % ^^J
523   \catcode34 12 % "
524   \catcode39 12 % '
525   \catcode40 12 % (
526   \catcode41 12 % )
527   \catcode44 12 % ,
528   \catcode46 12 % .
529   \catcode60 12 % <
530   \catcode61 12 % =
531   \catcode95 12 % _ (other!)
532   \catcode96 12 % `
533   \endlinechar=10 %
534   \ifnum\luatexversion<36 %
535     \directlua0%
536   \else %
537     \expandafter\directlua %
538   \fi %
539   {%
540     do
541       local script = "oberdiek.luatex.lua"
542       local file = kpse.find_file(script, "texmfscripts")
543       if file then
544         texio.write_nl("(" .. file .. ")")

```



```

545         dofile(file)
546     else
547         error("File '" .. script .. "' not found")
548     end
549 end
550 }%
551 \endgroup%
552 \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`.

```

553 <lua>
554 module("oberdiek.luatex", package.seeall)
555 function kpse_module_loader(module)
556     local script = module .. ".lua"
557     local file = kpse.find_file(script, "texmfscripts")
558     if file then
559         local loader, error = loadfile(file)
560         if loader then
561             texio.write_nl("(" .. file .. ")")
562             return loader
563         end
564         return "\n\t[oberdiek.luatex.kpse_module_loader] Loading error:\n\t"
565             .. error
566     end
567     return "\n\t[oberdiek.luatex.kpse_module_loader] Search failed"
568 end
569 table.insert(package.loaders, kpse_module_loader)
570 \lua>

```

## 3 Test

```

571 <test2>
572 \documentclass{article}
573 \def\LoadCommand{%
574     \RequirePackage{luatex}[2010/03/09]%
575 }
576 \test2>
577 <test3>
578 \documentclass{article}
579 \def\LoadCommand{%
580     \RequirePackage{luatex-loader}[2010/03/09]%
581 }
582 \test3>

```

### 3.1 Catcode checks for loading

```

583 <test1>
584 \catcode'\{=1 %
585 \catcode'\}=2 %

```

```

586 \catcode'\#=6 %
587 \catcode'\@=11 %
588 \expandafter\ifx\csname count@\endcsname\relax
589 \countdef\count@=255 %
590 \fi
591 \expandafter\ifx\csname @gobble\endcsname\relax
592 \long\def\@gobble#1{}%
593 \fi
594 \expandafter\ifx\csname @firstofone\endcsname\relax
595 \long\def\@firstofone#1{#1}%
596 \fi
597 \expandafter\ifx\csname loop\endcsname\relax
598 \expandafter\@firstofone
599 \else
600 \expandafter\@gobble
601 \fi
602 {%
603 \def\loop#1\repeat{%
604 \def\body{#1}%
605 \iterate
606 }%
607 \def\iterate{%
608 \body
609 \let\next\iterate
610 \else
611 \let\next\relax
612 \fi
613 \next
614 }%
615 \let\repeat=\fi
616 }%
617 \def\RestoreCatcodes{
618 \count@=0 %
619 \loop
620 \edef\RestoreCatcodes{%
621 \RestoreCatcodes
622 \catcode\the\count@=\the\catcode\count@\relax
623 }%
624 \ifnum\count@<255 %
625 \advance\count@ 1 %
626 \repeat
627
628 \def\RangeCatcodeInvalid#1#2{%
629 \count@=#1\relax
630 \loop
631 \catcode\count@=15 %
632 \ifnum\count@<#2\relax
633 \advance\count@ 1 %
634 \repeat
635 }
636 \expandafter\ifx\csname LoadCommand\endcsname\relax
637 \def\LoadCommand{\input luatex.sty\relax}%
638 \fi
639 \def\Test{%
640 \RangeCatcodeInvalid{0}{47}%
641 \RangeCatcodeInvalid{58}{64}%
642 \RangeCatcodeInvalid{91}{96}%
643 \RangeCatcodeInvalid{123}{255}%
644 \catcode'\@=12 %
645 \catcode'\=0 %
646 \catcode'\{=1 %
647 \catcode'\}=2 %

```

```

648 \catcode'\#=6 %
649 \catcode'\[=12 %
650 \catcode'\]=12 %
651 \catcode'\%=14 %
652 \catcode'\ =10 %
653 \catcode13=5 %
654 \LoadCommand
655 \RestoreCatcodes
656 }
657 \Test
658 \csname @@end\endcsname
659 \end
660 </test1>

```

## 3.2 Catcode tables

### 3.2.1 Predefined catcode tables

```

661 <*test4>
662 \NeedsTeXFormat{LaTeX2e}

```

Remember L<sup>A</sup>T<sub>E</sub>X's initial catcodes in count registers starting at \TestLaTeX.

```

663 \count0=0 %
664 \chardef\TestLaTeX=1000 %
665 \chardef\TestMax=300 %
666 \loop
667   \count\numexpr\TestLaTeX+\count0\relax=\catcode\count0 %
668   \ifnum\count0<\TestMax
669     \advance\count0 by 1 %
670   \repeat
671 \documentclass{minimal}
672 \usepackage{luatex}[2010/03/09]
673 \usepackage{qstest}
674 \IncludeTests{*}
675 \LogTests{log}{*}{*}
676 \makeatletter
677 \def\Check#1{%
678   \Expect*{\the\count@=\the\catcode\count@}%
679   *{\the\count@=#1}%
680 }
681 \newcount\scratch
682 \def\Test#1#2{%
683   \begin{qstest}{CatcodeTable#1}{CatcodeTable#1}%
684     \luatexcatcodetable\csname CatcodeTable#1\endcsname
685     \count@=\z@
686     \loop
687       \scratch=#2\relax
688       \Expect*{\the\count@=\the\catcode\count@}%
689       *{\the\count@=\the\scratch}%
690       \ifnum\count@<\TestMax
691         \advance\count@\@ne
692       \repeat
693   \end{qstest}%
694 }
695 \Test{LaTeX}{\the\count\numexpr\TestLaTeX+\count@}
696 \Test{String}{\ifnum\count@=32 10\else 12\fi}
697 \Test{Other}{12}
698 \luatexinitcatcodetable99 %
699 \Test{IniTeX}{%
700   0\relax
701   \begingroup
702     \luatexcatcodetable99 %
703     \global\scratch=\the\catcode\count@
704   \endgroup

```

705 }

### 3.2.2 Catcode table number stack

```
706 \begin{qstest}{CatcodeTableNumStack}{CatcodeTableNumStack}
707   \def\TestStack#1{%
708     \Expect*{\LuT@NumStack}{#1}%
709   }%
710   \TestStack{0}%
711   \PushCatcodeTableNumStack
712   \TestStack{{0}0}%
713   \@firstofone{%
714     \begin{group}
715       \luatexinitcatcodetable12 %
716       \luatexcatcodetable12 %
717       \PushCatcodeTableNumStack
718       \TestStack{{12}{0}0}%
719       \PopCatcodeTableNumStack
720       \TestStack{{0}0}%
721       \PopCatcodeTableNumStack
722       \TestStack{0}%
723       \def\TestWarning{Missing empty stack warning}%
724       \def\@PackageWarning#1#2{\def\TestWarning{empty stack}}%
725       \PopCatcodeTableNumStack
726       \TestStack{0}%
727       \Expect*{\TestWarning}{empty stack}%
728     \end{group}
729   }%
730 \end{qstest}
```

### 3.2.3 Catcode table stack

```
731 \begin{qstest}{CatcodeTableStack}{CatcodeTableStack}
732   \def\TestStack#1{%
733     \Expect*{\the\CatcodeTableStack}{#1}%
734   }%
735   \TestStack{0}%
736   \IncCatcodeTableStack
737   \TestStack{2}%
738   \IncCatcodeTableStack
739   \TestStack{4}%
740   \begin{group}
741     \IncCatcodeTableStack
742     \TestStack{6}%
743   \end{group}
744   \TestStack{6}%
745   \begin{group}
746     \DecCatcodeTableStack
747     \TestStack{4}%
748   \end{group}
749   \TestStack{4}%
750   \DecCatcodeTableStack
751   \TestStack{2}%
752   \DecCatcodeTableStack
753   \TestStack{0}%
754   \begin{group}
755     \def\TestError{Missing error}%
756     \def\@PackageError#1#2#3{%
757       \def\TestError{Empty stack}%
758     }%
759     \DecCatcodeTableStack
760     \TestStack{0}%
761     \Expect*{\TestError}{Empty stack}%
762   \end{group}
763 \end{qstest}
```

### 3.2.4 Catcode regime macros

```

764 \begin{qstest}{CatcodeRegime}{CatcodeRegime}
765   \def\TestStacks#1#2#3{%
766     \Expect*{\the\luatexcatcodetable}{#1}%
767     \Expect*{\the\CatcodeTableStack}{#2}%
768     \Expect*{\LuT@NumStack}{#3}%
769   }%
770   \TestStacks{0}{0}{0}%
771   \catcode'\|=7 %
772   \BeginCatcodeRegime\CatcodeTableLaTeX
773     \TestStacks{2}{2}{00}%
774     \Expect*{\the\catcode'\|}{12}%
775   \EndCatcodeRegime
776   \TestStacks{0}{0}{0}%
777   \Expect*{\the\catcode'\|}{7}%
778 \end{qstest}

```

### 3.3 Attribute allocation

```

779 \begin{qstest}{Attributes}{Attributes}
780   \newattribute\TestAttr
781   \Expect*{\meaning\TestAttr}%
782     *{\string\attribute\number\allocationnumber}%
783   \Expect*{\the\allocationnumber}{0}%
784   \begingroup
785     \newattribute\TestAttr
786     \Expect*{\the\allocationnumber}{1}%
787   \endgroup
788   \Expect*{\the\allocationnumber}{0}%
789   \Expect*{\meaning\TestAttr}*{\string\attribute1}%
790   \Expect*{\the\TestAttr}*{\number\LuT@UnsetAttributeValue}%
791   \def\Test#1{%
792     \setattribute\TestAttr{#1}%
793     \Expect*{\the\TestAttr}{#1}%
794   }%
795   \Test{0}%
796   \Test{1}%
797   \Test{-1}%
798   \Test{123}%
799   \unsetattribute\TestAttr
800   \Expect*{\the\TestAttr}*{\number\LuT@UnsetAttributeValue}%
801   \begingroup
802     \Expect*{\the\TestAttr}*{\number\LuT@UnsetAttributeValue}%
803     \Test{1234}%
804   \endgroup
805   \Expect*{\the\TestAttr}*{\number\LuT@UnsetAttributeValue}%
806 \end{qstest}

807 \@@end
808 </test4>

```

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

```

809 <*test5>
810 \input luatex.sty\relax
811 \newattribute\TestAttr
812 \setattribute\TestAttr{10}
813 \unsetattribute\TestAttr
814 \newcatcodetable\TestCTa
815 \begingroup
816   \SetCatcodeRange{'A'}{'Z'}{12}%
817 \endgroup
818 \BeginCatcodeRegime\CatcodeTableLaTeX
819 \EndCatcodeRegime

```

```
820 \end
821 </test5>
```

## 4 Installation

### 4.1 Download

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

[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`.

---

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

## 4.4 Refresh file name databases

If your  $\text{\TeX}$  distribution (te $\text{\TeX}$ , mik $\text{\TeX}$ , ...) relies on file name databases, you must refresh these. For example, te $\text{\TeX}$  users run `texhash` or `mktexlsr`.

## 4.5 Some details for the interested

**Attached source.** The PDF documentation on CTAN also includes the `.dtx` source file. It can be extracted by AcrobatReader 6 or higher. Another option is `pdftk`, e.g. unpack the file into the current directory:

```
pdftk 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 Lua $\text{\TeX}$  0.36.

[2009/12/02 v0.3]

- Unsetting of attributes updated for Lua $\text{\TeX}$  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; numbers in roman refer to the code lines where the entry is used.

Symbols	C
<code>\#</code> ..... 586, 648	<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, 379, 390, 395, 396, 397, 399, 400, 401, 402, 403, 404, 405, 406, 407, 461, 462, 463, 464, 465, 466, 467, 475, 489, 490, 491, 492, 493, 494, 495, 496, 497, 498, 499, 500, 501, 522, 523, 524, 525, 526, 527, 528, 529, 530, 531, 532, 584, 585, 586, 587, 622, 631, 644, 645, 646, 647, 648, 649, 650, 651, 652, 653, 667, 678, 688, 703, 771, 774, 777
<code>\%</code> ..... 651	<code>\catcodetable</code> ..... 343, 346
<code>\:</code> ..... 261, 262, 263, 264, 265, 268	<code>\CatcodeTableIniTeX</code> .. 4, 384, 388, 392
<code>\@</code> ..... 587, 644	<code>\CatcodeTableLaTeX</code> . 387, 408, 772, 818
<code>\@@end</code> ..... 807	<code>\CatcodeTableOther</code> ..... 386, 419
<code>\@INTERNAL@ERROR</code> ..... 300	<code>\CatcodeTableStack</code> 4, 334, 335, 350, 351, 359, 360, 444, 445, 733, 767
<code>\@PackageError</code> ..... . 113, 153, 178, 284, 353, 362, 756	<code>\CatcodeTableString</code> ..... 385, 417
<code>\@PackageWarning</code> ..... 431, 724	<code>\chardef</code> ..... 337, 341, 664, 665
<code>\@PackageWarningNoLine</code> ..... 200	<code>\Check</code> ..... 677
<code>\@car</code> ..... 233, 434	<code>\count</code> 205, 206, 663, 667, 668, 669, 695
<code>\@cdr</code> ..... 236, 436	<code>\count@</code> ..... 371, 376, 379, 589, 618, 622, 624, 625, 629, 631, 632, 633, 678, 679, 685, 688, 689, 690, 691, 695, 696, 703
<code>\@ehc</code> ..... 115, 155, 180, 287	<code>\countdef</code> ..... 589
<code>\@ehd</code> ..... 355, 364	<code>\csname</code> ..... 10, 18, 44, 60, 67, 104, 121, 122, 127, 129, 132, 133, 137, 139, 147, 150, 151, 162, 164, 172, 175, 176, 194, 199, 221, 224, 227, 230, 233, 236, 239, 270, 271, 273, 274, 277, 452, 468, 476, 502, 518, 588, 591, 594, 597, 636, 658, 684
<code>\@empty</code> ..... 221	
<code>\@firstofone</code> .. 227, 391, 595, 598, 713	
<code>\@firstoftwo</code> ..... 230, 241	
<code>\@gobble</code> ..... 224, 592, 600	
<code>\@ifnch</code> ..... 247, 249, 266	
<code>\@ifnextchar</code> ..... 241, 243	
<code>\@ifstar</code> ..... 239	
<code>\@let@token</code> ..... 247, 250, 253, 266	
<code>\@makeoother</code> ..... . 390, 410, 411, 412, 413, 414, 418	
<code>\@ne</code> ..... 311, 380, 691	
<code>\@nil</code> ..... 234, 237, 434, 436	
<code>\@sptoken</code> ..... 250, 261	
<code>\@tempcnta</code> 270, 369, 374, 378, 379, 380	
<code>\@tempcntb</code> ..... 273, 370, 375, 378	
<code>\@undefined</code> .... 52, 139, 164, 279, 510	
<code>\@xifnch</code> ..... 251, 264	
<code>\[</code> ..... 649	
<code>\[</code> ..... 645	
<code>\{</code> ..... 584, 646	
<code>\}</code> ..... 585, 647	
<code>\]</code> ..... 650	
<code>\ </code> ..... 771, 774, 777	
<code>\_</code> ..... 652	
A	D
<code>\advance</code> ..... 311, 339, 351, 360, 380, 625, 633, 669, 691	<code>\DecCatcodeTableStack</code> ..... ..... 358, 448, 746, 750, 752, 759
<code>\aftergroup</code> ..... 26, 484	<code>\directlua</code> ..... 141, 166, 535, 537
<code>\allocationnumber</code> . 312, 313, 315, 340, 341, 343, 782, 783, 786, 788	<code>\documentclass</code> ..... 572, 578, 671
<code>\attribute</code> ..... 315, 317, 782, 789	
B	E
<code>\begin</code> ..... 683, 706, 731, 764, 779	<code>\empty</code> ..... 13, 14, 471, 472
<code>\BeginCatcodeRegime</code> .. 5, 440, 772, 818	<code>\end</code> .. 659, 693, 730, 763, 778, 806, 820
<code>\body</code> ..... 604, 608	<code>\EndCatcodeRegime</code> .... 447, 775, 819
	<code>\endcsname</code> ..... 10, 18, 44, 60, 67, 104, 121, 122, 127, 129, 132, 133, 137, 139, 147, 150, 151, 162, 164, 172, 175, 176, 194, 199, 221, 224, 227, 230, 233, 236, 239, 270, 271, 273, 274,



277, 452, 468, 476, 502, 518, 588, 591, 594, 597, 636, 658, 684	\LuT@temp . 204, 209, 210, 211, 212, 213, 214, 215, 261, 268, 368, 382
\endinput . . . . . 26, 117, 484	\LuT@UnsetAttributeValue . . . 324, 325, 327, 330, 790, 800, 802, 805
\endlinechar . . . . . 533	
\errmessage . . . . . 317, 346	
\Expect . . . . . 678, 688, 708, 727, 733, 761, 766, 767, 768, 774, 777, 781, 783, 786, 788, 789, 790, 793, 800, 802, 805	
	<b>M</b>
	\m@ne . . . . . 308, 325, 333
	\makeatletter . . . . . 676
	\meaning . . . . . 781, 789
	\MessageBreak . . . . . 285
<b>F</b>	<b>N</b>
\futurelet . . . . . 247, 266	\n . . . . . 564, 567
	\NeedsTeXFormat . . . . . 662
<b>G</b>	\newattribute . . . 3, 309, 780, 785, 811
\globbox . . . . . 219	\newbox . . . . . 219
\globcount . . . . . 216	\newcatcodetable . . . . . . . . 4, 336, 384, 385, 386, 387, 814
\globdimen . . . . . 217	\newcommand . . . . . 305
\globskip . . . . . 218	\newcount . . . . 216, 307, 332, 334, 681
	\newdimen . . . . . 217
<b>I</b>	\newskip . . . . . 218
\ifcase . . . . . 291	\next . . . . . 609, 611, 613
\ifluatex . . . . . 111	\number . . . . . 782, 790, 800, 802, 805
\ifnum . . . . . 119, 140, 165, 205, 310, 323, 337, 350, 359, 378, 534, 624, 632, 668, 690, 696	\numexpr 321, 374, 375, 376, 442, 667, 695
\ifx . . . . . 11, 14, 18, 44, 52, 55, 104, 127, 129, 137, 147, 162, 172, 194, 199, 221, 224, 227, 230, 233, 236, 239, 250, 253, 270, 273, 277, 279, 282, 290, 430, 452, 469, 472, 476, 502, 510, 513, 588, 591, 594, 597, 636	
\immediate . . . . . 20, 46, 478, 504	
\IncCatcodeTableStack . . . . . . . . . . 349, 443, 736, 738, 741	<b>P</b>
\IncludeTests . . . . . 674	\PackageInfo . . . . . 23, 481
\input . . . . . 105, 106, 453, 637, 810	\PopCatcodeTableNumStack . . . . . . . . . . 429, 449, 719, 721, 725
\iterate . . . . . 605, 607, 609	\ProvidesPackage . . . . 15, 61, 473, 519
	\PushCatcodeTableNumStack . . . . . . . . . . 5, 424, 441, 711, 717
<b>L</b>	
\LoadCommand . . . . . 573, 579, 637, 654	<b>R</b>
\LogTests . . . . . 675	\RangeCatcodeInvalid . . . . . . . . . . 628, 640, 641, 642, 643
\loop . . . . . 377, 603, 619, 630, 666, 686	\repeat 381, 603, 615, 626, 634, 670, 692
\luatexattributedef . . . . . 313	\RequirePackage . . . . . . . . . . 108, 109, 196, 455, 574, 580
\luatexcatcodetable 392, 426, 432, 434, 442, 445, 684, 702, 716, 766	\reserved@a . . . . . 245, 254
\luatexinitcatcodetable 388, 698, 715	\reserved@b . . . . . 246, 256
\luatexsavecatcodetable . . . . . . . . . . 408, 417, 419, 444	\reserved@c . . . . . 251, 254, 256, 259
\luatexversion . . . . . 119, 323, 534	\reserved@d . . . . . 244, 253
\LuT@AllocAttribute 307, 310, 311, 312	\RestoreCatcodes . . 617, 620, 621, 655
\LuT@AllocCatcodeTable . . . . . . . . . . 332, 337, 339, 340	
\LuT@AtEnd . . . . . 80, 81, 116, 457	<b>S</b>
\LuT@MakeLuatexPrimitive . . . . . . . . . . 125, 186, 187, 188, 189, 190, 191	\scratch . . . . . 681, 687, 689, 703
\LuT@MakePrimitive . . . . 120, 160, 192	\setattribute . . . . . 3, 320, 792, 812
\LuT@newcommand . . . . . 276, 309, 320, 324, 327, 329, 336, 349, 358, 367, 424, 429, 440, 447	\SetCatcodeRange . . . . . . . . 5, 367, 394, 398, 415, 416, 816
\LuT@NumStack . . . . . 423, 425, 426, 430, 434, 435, 436, 708, 768	
\LuT@NumStackEmpty . . . . 422, 423, 430	<b>T</b>
	\t . . . . . 564, 567
	\Test . . 639, 657, 682, 695, 696, 697, 699, 791, 795, 796, 797, 798, 803
	\TestAttr . . . . . 780, 781, 785, 789, 790, 792, 793, 799, 800, 802, 805, 811, 812, 813
	\TestCTa . . . . . 814
	\TestError . . . . . 755, 757, 761
	\TestLaTeX . . . . . 664, 667, 695
	\TestMax . . . . . 665, 668, 690

<code>\TestStack</code>	707, 710, 712, 718, 720, 722, 726, 732, 735, 737, 739, 742, 744, 747, 749, 751, 753, 760		
<code>\TestStacks</code>	765, 770, 773, 776		
<code>\TestWarning</code>	723, 724, 727		
<code>\the</code>	68, 69, 70, 71, 82, 315, 343, 369, 370, 371, 426, 622, 678, 679, 688, 689, 695, 703, 733, 766, 767, 774, 777, 783, 786, 788, 790, 793, 800, 802, 805		
<code>\TMP@EnsureCode</code>	79, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102		
<code>\tw@</code>	339, 351		
		<b>U</b>	
		<code>\unless</code>	378
		<code>\unsetattribute</code>	3, 314, 323, 799, 813
		<code>\usepackage</code>	672, 673
		<b>W</b>	
		<code>\wlog</code>	315, 342
		<code>\write</code>	20, 46, 478, 504
		<b>X</b>	
		<code>\x</code>	10, 11, 14, 19, 23, 25, 45, 50, 60, 66, 74, 468, 469, 472, 477, 481, 483, 503, 508, 518
		<b>Z</b>	
		<code>\z@</code>	335, 359, 432, 685