

# The `luatex` package

Heiko Oberdiek  
<oberdiek@uni-freiburg.de>

2009/04/10 v0.2

## 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	<code>L<sub>A</sub>T<sub>E</sub>X</code>	2
1.1.2	plain- <code>T<sub>E</sub>X</code>	2
1.2	Register allocation	3
1.2.1	Register with 16 bit	3
1.2.2	Insertions	3
1.3	Lua states	3
1.4	Attributes	3
1.5	Catcode tables	4
1.5.1	Interface proposal	4
1.6	Lua module loading	5
1.6.1	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 <code>LUA<sub>T</sub>EX</code>	8
2.4	Inherit support for $\epsilon$ - <code>T<sub>E</sub>X</code>	8
2.5	Adaption of $\epsilon$ - <code>T<sub>E</sub>X</code> 's register allocation	8
2.6	plain- <code>T<sub>E</sub>X</code> compatibility	9
2.7	Lua states	11
2.8	Attributes	11
2.8.1	Allocation	11
2.8.2	Interface	11
2.9	Catcode tables	12
2.9.1	Allocation	12
2.9.2	<code>\SetCatcodeRange</code>	12
2.9.3	Predefined catcode tables	13
2.9.4	Number stack	13
2.9.5	Catcode regime macros	14
2.10	Lua module loader	14
2.11	Lua script	16

<b>3</b>	<b>Test</b>	<b>16</b>
3.1	Catcode checks for loading . . . . .	17
3.2	Catcode tables . . . . .	18
3.2.1	Predefined catcode tables . . . . .	18
3.2.2	Catcode table number stack . . . . .	19
3.2.3	Catcode table stack . . . . .	19
3.2.4	Catcode regime macros . . . . .	20
3.3	Attribute allocation . . . . .	20
3.4	Lua states . . . . .	20
3.5	Short test for plain- $\text{\TeX}$ . . . . .	21
<b>4</b>	<b>Installation</b>	<b>21</b>
4.1	Download . . . . .	21
4.2	Bundle installation . . . . .	21
4.3	Package installation . . . . .	21
4.4	Refresh file name databases . . . . .	22
4.5	Some details for the interested . . . . .	22
<b>5</b>	<b>History</b>	<b>22</b>
	[2007/12/12 v0.1] . . . . .	22
	[2009/04/10 v0.2] . . . . .	23
<b>6</b>	<b>Index</b>	<b>23</b>

# 1 Documentation

## 1.1 Introduction

$\text{\TeX}$  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- $\text{\TeX}$  or  $\text{\LaTeX}$  implement an allocation scheme for registers. A package reserves with `\newcount` an unused register number for its own exclusive use.

Nowadays  $\text{\TeX}$  is not alone anymore:  $\varepsilon\text{-}\text{\TeX}$ , pdf $\text{\TeX}$  and other compilers for  $\text{\TeX}$  are developed that extend and add new features and resources.

Now  $\text{\LaTeX}$  has reached beta state. It inherits most of pdf $\text{\TeX}$ 's features including  $\varepsilon\text{-}\text{\TeX}$ . Also it implements new concepts such as attributes or catcode tables.

### 1.1.1 $\text{\LaTeX}$

$\text{\LaTeX} 2_{\varepsilon}$  is frozen and therefore refuses to even notice the new  $\text{\TeX}$  variants. Not even the old  $\varepsilon\text{-}\text{\TeX}$  is supported by its kernel. At least there is a third party package `etex` that manages the new  $\varepsilon\text{-}\text{\TeX}$  resources.

This package tries to do the same for  $\text{\LaTeX}$  and starts to support at least a few of the new features.

### 1.1.2 plain- $\text{\TeX}$

$\text{\LaTeX}$  has inherited its resource handling from plain- $\text{\TeX}$ . The interface is basically the same: `\newcount`, ... Therefore this package tries to follow this tradition by providing compatibility to plain- $\text{\TeX}$ . It can be loaded with plain- $\text{\TeX}$  and defines at least some of the features that this packages provides for  $\text{\LaTeX}$ .

## 1.2 Register allocation

### 1.2.1 Register with 16 bit

Because L<sup>A</sup>T<sub>E</sub>X is a super set of  $\varepsilon$ -T<sub>E</sub>X 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 L<sup>A</sup>T<sub>E</sub>X stores the registers of a class in the same Lua table unlike  $\varepsilon$ -T<sub>E</sub>X, where registers below 256 are stored in an array and higher numbers are put in a tree structure.

## 1.3 Lua states

`\newluastate {⟨cmd⟩}`

Macro `\newluastate` reserves a new Lua state and stores the number in `\cmd`.

## 1.4 Attributes

Nodes can have custom attributes in L<sup>A</sup>T<sub>E</sub>X. 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 T<sub>E</sub>X's range of non-negative integers up to  $2^{31}$  are available.

`\newattribute {⟨cmd⟩}`

Macro `\newattribute` defines command `⟨cmd⟩` using `\attributedef` using an 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 T<sub>E</sub>X integer numbers).

`\unsetAttribute {⟨cmd⟩}`

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

## 1.5 Catcode tables

L<sup>A</sup>T<sub>E</sub>X 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: L<sup>A</sup>T<sub>E</sub>X'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.)

### 1.5.1 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 { $\langle catcodetable \rangle$ }`  
`\EndCatcodeRegime`

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

```
\PushCatcodeTableNumStack
\catcodetable $\langle catcodetable \rangle$  \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 { $\langle from \rangle$ } { $\langle to \rangle$ } { $\langle catcode \rangle$ }`

The catcodes of characters in range from  $\langle from \rangle$  to inclusive  $\langle to \rangle$  are set to  $\langle catcode \rangle$ .

## 1.6 Lua module loading

Currently  $\text{\LaTeX}$  (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  $\langle module \rangle$  by

```
kpse.find_file(" $\langle module \rangle$ .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.

### 1.6.1 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-TeX 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 \expandafter\let\expandafter\x\csname ver@luatex.sty\endcsname
9 \ifcase 0%
10 \ifx\x\relax % plain
11 \else
12 \ifx\x\empty % LaTeX
13 \else
14 1%
15 \fi
16 \fi
17 \else
18 \catcode35 6 % #
19 \catcode123 1 % {
20 \catcode125 2 % }
21 \expandafter\ifx\csname PackageInfo\endcsname\relax
22 \def\x#1#2{%
23 \immediate\write-1{Package #1 Info: #2.}%
24 }%
25 \else
26 \def\x#1#2{\PackageInfo{#1}{#2, stopped}}%
27 \fi
28 \x{luatex}{The package is already loaded}%
29 \endgroup
30 \expandafter\endinput
31 \fi
32 \endgroup
```

Package identification:

```
33 \begingroup
34 \catcode35 6 % #
```

```

35 \catcode40 12 % (
36 \catcode41 12 % )
37 \catcode44 12 % ,
38 \catcode45 12 % -
39 \catcode46 12 % .
40 \catcode47 12 % /
41 \catcode58 12 % :
42 \catcode64 11 % @
43 \catcode123 1 % {
44 \catcode125 2 % }
45 \expandafter\ifx\csname ProvidesPackage\endcsname\relax
46   \def\x#1#2#3[#4]{\endgroup
47     \immediate\write-1{Package: #3 #4}%
48     \xdef#1{#4}%
49   }%
50 \else
51   \def\x#1#2[#3]{\endgroup
52     #2[#{#3}]%
53     \ifx#1@undefined
54       \xdef#1{#3}%
55     \fi
56     \ifx#1\relax
57       \xdef#1{#3}%
58     \fi
59   }%
60 \fi
61 \expandafter\x\csname ver@luatex.sty\endcsname
62 \ProvidesPackage{luatex}%
63 [2009/04/10 v0.2 LuaTeX basic definition package (H0)]

```

## 2.2 Catcodes

```

64 \begingroup
65 \catcode123 1 % {
66 \catcode125 2 % }
67 \def\x{\endgroup
68   \expandafter\edef\csname LuT@AtEnd\endcsname{%
69     \catcode35 \the\catcode35\relax
70     \catcode64 \the\catcode64\relax
71     \catcode123 \the\catcode123\relax
72     \catcode125 \the\catcode125\relax
73   }%
74 }%
75 \x
76 \catcode35 6 % #
77 \catcode64 11 % @
78 \catcode123 1 % {
79 \catcode125 2 % }
80 \def\TMP@EnsureCode#1#2{%
81   \edef\LuT@AtEnd{%
82     \LuT@AtEnd
83     \catcode#1 \the\catcode#1\relax
84   }%
85   \catcode#1 #2\relax
86 }
87 \TMP@EnsureCode{10}{12}% ^^J
88 \TMP@EnsureCode{34}{12}% "
89 \TMP@EnsureCode{36}{3}% $
90 \TMP@EnsureCode{39}{12}% '
91 \TMP@EnsureCode{40}{12}% (
92 \TMP@EnsureCode{41}{12}% )
93 \TMP@EnsureCode{42}{12}% *

```

```

94 \TMP@EnsureCode{43}{12}% +
95 \TMP@EnsureCode{44}{12}% ,
96 \TMP@EnsureCode{45}{12}% -
97 \TMP@EnsureCode{46}{12}% .
98 \TMP@EnsureCode{47}{12}% /
99 \TMP@EnsureCode{60}{12}% <
100 \TMP@EnsureCode{61}{12}% =
101 \TMP@EnsureCode{62}{12}% >
102 \TMP@EnsureCode{95}{12}% _ (other!)
103 \TMP@EnsureCode{96}{12}% ‘

```

## 2.3 Check for LuaTeX

Without L<sup>A</sup>T<sub>E</sub>X there is no point in using this package.

```

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

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

```

## 2.4 Inherit support for $\epsilon$ -T<sub>E</sub>X

Package `etex` is not compatible for plain-T<sub>E</sub>X. 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`.

```

120 \begingroup\expandafter\expandafter\expandafter\endgroup
121 \expandafter\ifx\csname RequirePackage\endcsname\relax
122 \else
123   \RequirePackage{etex}[1998/03/26]%
124 \fi

```

## 2.5 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. L<sup>A</sup>T<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.

```

125 \begingroup\expandafter\expandafter\expandafter\endgroup
126 \expandafter\ifx\csname et@xins\endcsname\relax
127   \@PackageWarningNoLine{luatex}{%
128     Support for eTeX is not loaded (etex.src)%
129   }%
130 \else
131   \def\LuT@temp#1{%
132     \ifnum\count27#1=32768 %
133       \count27#1=65536 %
134     \fi
135   }%
136   \LuT@temp0%

```



```

137 \LuT@temp1%
138 \LuT@temp2%
139 \LuT@temp3%
140 \LuT@temp4%
141 \LuT@temp5%
142 \LuT@temp6%

```

$\varepsilon$ -TeX uses an array for the first 256 registers and then a tree structure. L<sup>A</sup>T<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.

```

143 \let\newcount\globcount
144 \let\newdimen\globdimen
145 \let\newskip\globskip
146 \let\newbox\globbox
147 \fi

```

## 2.6 plain-TeX compatibility

`\@empty`

```

148 \expandafter\ifx\csname @empty\endcsname\relax
149 \def\@empty{}%
150 \fi

```

`\@gobble`

```

151 \expandafter\ifx\csname @gobble\endcsname\relax
152 \long\def\@gobble#1{}%
153 \fi

```

`\@firstofone`

```

154 \expandafter\ifx\csname @firstofone\endcsname\relax
155 \long\def\@firstofone#1{#1}%
156 \fi

```

`\@firstoftwo`

```

157 \expandafter\ifx\csname @firstoftwo\endcsname\relax
158 \long\def\@firstoftwo#1#2{#1}%
159 \fi

```

`\@car`

```

160 \expandafter\ifx\csname @car\endcsname\relax
161 \def\@car#1#2\@nil{#1}%
162 \fi

```

`\@cdr`

```

163 \expandafter\ifx\csname @cdr\endcsname\relax
164 \def\@cdr#1#2\@nil{#2}%
165 \fi

```

`\@ifstar`

```

166 \expandafter\ifx\csname @ifstar\endcsname\relax
167 \def\@ifstar#1{%
168 \ifnextchar*\@firstoftwo{#1}}%
169 }%

```

`\@ifnextchar`

```

170 \long\def\@ifnextchar#1#2#3{%
171 \let\reserved@=#1%
172 \def\reserved@a{#2}%
173 \def\reserved@b{#3}%
174 \futurelet\@let@token\@ifnch
175 }%

```

```

\@ifnch
176 \def\@ifnch{%
177 \ifx\@let@token\@sptoken
178 \let\reserved@c\@xifnch
179 \else
180 \ifx\@let@token\reserved@d
181 \let\reserved@c\reserved@a
182 \else
183 \let\reserved@c\reserved@b
184 \fi
185 \fi
186 \reserved@c
187 }%

\@sptoken
188 \let\LuT@temp\:%
189 \def\:{\let\@sptoken= }%
190 \: % explicit space

\@xifnch
191 \def\:{\@xifnch}%
192 \expandafter\def\: {%
193 \futurelet\@let@token\@ifnch
194 }%
195 \let\:\LuT@temp
196 \fi

\@tempcnta
197 \expandafter\ifx\csname @tempcnta\endcsname\relax
198 \csname newcount\endcsname\@tempcnta
199 \fi

\@tempcntb
200 \expandafter\ifx\csname @tempcntb\endcsname\relax
201 \csname newcount\endcsname\@tempcntb
202 \fi

\LuT@newcommand
203 \begingroup\expandafter\expandafter\expandafter\endgroup
204 \expandafter\ifx\csname newcommand\endcsname\relax
205 \def\LuT@newcommand#1[#2]#3{%
206 \ifx#1\@undefined
207 \let#1\relax
208 \else
209 \ifx#1\relax
210 \else
211 \@PackageError{luatex}{%
212 \string#1 is already defined.\MessageBreak
213 Redefinition is skipped%
214 }\@ehc
215 \fi
216 \fi
217 \ifx#1\relax
218 \ifcase#2 %
219 \def#1{#3}%
220 \or
221 \def#1##1{#3}%
222 \or
223 \def#1##1##2{#3}%
224 \or
225 \def#1##1##2##3{#3}%

```

```

226      \or
227      \@INTERNAL@ERROR
228      \fi
229      \fi
230  }%
231 \else
232   \def\LuT@newcommand{\newcommand*}%
233 \fi

```

## 2.7 Lua states

\LuT@AllocLuaState

```

234 \newcount\LuT@AllocLuaState
235 \LuT@AllocLuaState=\z@

```

\newluastate

```

236 \LuT@newcommand\newluastate[1]{%
237   \ifnum\LuT@AllocLuaState<65535 %
238     \global\advance\LuT@AllocLuaState\@ne
239     \allocationnumber\LuT@AllocLuaState
240     \global\chardef#1=\allocationnumber
241     \wlog{\string#1=\string\luastate\the\allocationnumber}%
242   \else
243     \errmessage{No room for a new \string\luastate}%
244   \fi
245 }

```

## 2.8 Attributes

### 2.8.1 Allocation

\LuT@AllocAttribute

```

246 \newcount\LuT@AllocAttribute
247 \LuT@AllocAttribute=\m@ne

```

\newattribute

```

248 \LuT@newcommand\newattribute[1]{%
249   \ifnum\LuT@AllocAttribute<65535 %
250     \global\advance\LuT@AllocAttribute\@ne
251     \allocationnumber\LuT@AllocAttribute
252     \global\attributedef#1=\allocationnumber
253     \unsetattribute{#1}%
254     \wlog{\string#1=\string\attribute\the\allocationnumber}%
255   \else
256     \errmessage{No room for a new \string\attribute}%
257   \fi
258 }

```

### 2.8.2 Interface

\setattribute

```

259 \LuT@newcommand\setattribute[2]{%
260   #1=\numexpr#2\relax
261 }

```

\unsetattribute

```

262 \LuT@newcommand\unsetattribute[1]{%
263   #1=\m@ne
264 }

```

## 2.9 Catcode tables

### 2.9.1 Allocation

`\LuT@AllocCatcodeTable`

```
265 \newcount\LuT@AllocCatcodeTable
266 \LuT@AllocCatcodeTable=\m@ne
267 \newcount\CatcodeTableStack
268 \CatcodeTableStack=\z@
```

`\newcatcodetable`

```
269 \LuT@newcommand\newcatcodetable[1]{%
270   \ifnum\LuT@AllocCatcodeTable<1114110 % 0x10FFF is maximal \chardef
271     % or < 268435455 % 228 - 1
272   \global\advance\LuT@AllocCatcodeTable by\tw@
273   \allocationnumber=\LuT@AllocCatcodeTable
274   \global\chardef#1=\allocationnumber
275   \wlog{%
276     \string#1=\string\catcodetable\the\allocationnumber
277   }%
278   \else
279     \errmessage{No room for a new \string\catcodetable}%
280   \fi
281 }%
```

`\IncCatcodeTableStack`

```
282 \LuT@newcommand\IncCatcodeTableStack[0]{%
283   \ifnum\CatcodeTableStack<268435454 %
284     \global\advance\CatcodeTableStack by\tw@
285   \else
286     \@PackageError{luatex}{%
287       Catcode table stack overflow%
288     }\@ehd
289   \fi
290 }
```

`\DecCatcodeTableStack`

```
291 \LuT@newcommand\DecCatcodeTableStack[0]{%
292   \ifnum\CatcodeTableStack>\z@
293     \global\advance\CatcodeTableStack by-2 %
294   \else
295     \@PackageError{luatex}{%
296       Catcode table stack is empty%
297     }\@ehd
298   \fi
299 }
```

### 2.9.2 \SetCatcodeRange

`\SetCatcodeRange`

```
300 \LuT@newcommand\SetCatcodeRange[3]{%
301   \edef\LuT@temp{%
302     \noexpand\@tempcnta=\the\@tempcnta
303     \noexpand\@tempcntb=\the\@tempcntb
304     \noexpand\count@=\the\count@
305     \relax
306   }%
307   \@tempcnta=\numexpr#1\relax
308   \@tempcntb=\numexpr#2\relax
309   \count@=\numexpr#3\relax
310   \loop
311     \unless\ifnum\@tempcnta>\@tempcntb
```

```

312     \catcode\@tempcnta=\count@
313     \advance\@tempcnta by \@ne
314     \repeat
315     \LuT@temp
316 }

```

### 2.9.3 Predefined catcode tables

```

317 \newcatcodetable\CatcodeTableIniTeX
318 \newcatcodetable\CatcodeTableString
319 \newcatcodetable\CatcodeTableOther
320 \newcatcodetable\CatcodeTableLaTeX

321 \initcatcodetable\CatcodeTableIniTeX
322 \begingroup
323   \def\@makeoother#1{\catcode#1=12\relax}%
324   \@firstofone{%
325     \catcodetable\CatcodeTableIniTeX
326     \begingroup
327       \SetCatcodeRange{0}{8}{15}%
328       \catcode9=10 % tab
329       \catcode11=15 %
330       \catcode12=13 % form feed
331       \SetCatcodeRange{14}{31}{15}%
332       \catcode35=6 % hash
333       \catcode36=3 % dollar
334       \catcode38=4 % ampersand
335       \catcode94=7 % circumflex
336       \catcode95=8 % underscore
337       \catcode123=1 % brace left
338       \catcode125=2 % brace right
339       \catcode126=13 % tilde
340       \catcode127=15 %
341       \savecatcodetable\CatcodeTableLaTeX
342     \endgroup
343     \@makeoother{0}% nul
344     \@makeoother{13}% carriage return
345     \@makeoother{37}% percent
346     \@makeoother{92}% backslash
347     \@makeoother{127}%
348     \SetCatcodeRange{65}{90}{12}% A-Z
349     \SetCatcodeRange{97}{122}{12}% a-z
350     \savecatcodetable\CatcodeTableString
351     \@makeoother{32}% space
352     \savecatcodetable\CatcodeTableOther
353   \endgroup
354 }%

```

### 2.9.4 Number stack

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

```

355 \def\LuT@NumStackEmpty{0}

```

`\LuT@NumStack`

```

356 \let\LuT@NumStack\LuT@NumStackEmpty

```

`\PushCatcodeTableNumStack`

```

357 \LuT@newcommand\PushCatcodeTableNumStack[0]{%
358   \xdef\LuT@NumStack{%
359     {\the\catcodetable}\LuT@NumStack
360   }%
361 }

```

\PopCatcodeTableNumStack

```

362 \LuT@newcommand\PopCatcodeTableNumStack[0]{%
363   \ifx\LuT@NumStack\LuT@NumStackEmpty
364     \@PackageWarning{luatex}{Empty catcode table number stack}%
365     \catcodetable\z@
366   \else
367     \catcodetable=\expandafter\@car\LuT@NumStack\@nil\relax
368     \xdef\LuT@NumStack{%
369       \expandafter\@cdr\LuT@NumStack\@nil
370     }%
371   \fi
372 }
```

## 2.9.5 Catcode regime macros

\BeginCatcodeRegime

```

373 \LuT@newcommand\BeginCatcodeRegime[1]{%
374   \PushCatcodeTableNumStack
375   \catcodetable=\numexpr#1\relax
376   \IncCatcodeTableStack
377   \savecatcodetable\CatcodeTableStack
378   \catcodetable\CatcodeTableStack
379 }
```

\EndCatcodeRegime

```

380 \LuT@newcommand\EndCatcodeRegime[0]{%
381   \DecCatcodeTableStack
382   \PopCatcodeTableNumStack
383 }
```

## 2.10 Lua module loader

```

384 \begingroup\expandafter\expandafter\expandafter\endgroup
385 \expandafter\ifx\csname RequirePackage\endcsname\relax
386   \input luatex-loader.sty\relax
387 \else
388   \RequirePackage{luatex-loader}[2009/04/10]%
389 \fi

390 \LuT@AtEnd
391 </package>

392 <*loader>
```

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

```

393 \begingroup
394   \catcode44 12 % ,
395   \catcode45 12 % -
396   \catcode46 12 % .
397   \catcode58 12 % :
398   \catcode64 11 % @
399   \expandafter\let\expandafter\x\csname ver@luatex-loader.sty\endcsname
400   \ifcase 0%
401     \ifx\x\relax % plain
402     \else
403       \ifx\x\empty % LaTeX
404       \else
405         1%
406       \fi
407     \fi
408   \else
409     \catcode35 6 % #
410     \catcode123 1 % {
```

```

411 \catcode125 2 % }
412 \expandafter\ifx\csname PackageInfo\endcsname\relax
413 \def\x#1#2{%
414 \immediate\write-1{Package #1 Info: #2.}%
415 }%
416 \else
417 \def\x#1#2{\PackageInfo{#1}{#2, stopped}}%
418 \fi
419 \x{luatex-loader}{The package is already loaded}%
420 \endgroup
421 \expandafter\endinput
422 \fi
423 \endgroup
Package identification:
424 \begingroup
425 \catcode35 6 % #
426 \catcode40 12 % (
427 \catcode41 12 % )
428 \catcode44 12 % ,
429 \catcode45 12 % -
430 \catcode46 12 % .
431 \catcode47 12 % /
432 \catcode58 12 % :
433 \catcode64 11 % @
434 \catcode123 1 % {
435 \catcode125 2 % }
436 \expandafter\ifx\csname ProvidesPackage\endcsname\relax
437 \def\x#1#2#3[#4]{\endgroup
438 \immediate\write-1{Package: #3 #4}%
439 \xdef#1{#4}%
440 }%
441 \else
442 \def\x#1#2[#3]{\endgroup
443 #2[#{#3}]%
444 \ifx#1\@undefined
445 \xdef#1{#3}%
446 \fi
447 \ifx#1\relax
448 \xdef#1{#3}%
449 \fi
450 }%
451 \fi
452 \expandafter\x\csname ver@luatex-loader.sty\endcsname
453 \ProvidesPackage{luatex-loader}%
454 [2009/04/10 v0.2 Lua module loader (H0)]
455 \begingroup
456 \catcode10 12 % ^^J
457 \catcode34 12 % "
458 \catcode39 12 % '
459 \catcode40 12 % (
460 \catcode41 12 % )
461 \catcode44 12 % ,
462 \catcode46 12 % .
463 \catcode60 12 % <
464 \catcode61 12 % =
465 \catcode95 12 % _ (other!)
466 \catcode96 12 % `
467 \endlinechar=10 %
468 \ifnum\luatexversion<36 %
469 \directlua0%
470 \else %
471 \expandafter\directlua %

```

```

472 \fi %
473 {%
474     do
475         local script = "oberdiek.luatex.lua"
476         local file = kpse.find_file(script, "texmfscripts")
477         if file then
478             texio.write_nl("(" .. file .. ")")
479             dofile(file)
480         else
481             error("File '" .. script .. "' not found")
482         end
483     end
484 }%
485 \endgroup%
486 </loader>

```

## 2.11 Lua script

Currently L<sup>A</sup>T<sub>E</sub>X 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`.

```

487 <lua>
488 module("oberdiek.luatex", package.seeall)
489 function kpse_module_loader(module)
490     local script = module .. ".lua"
491     local file = kpse.find_file(script, "texmfscripts")
492     if file then
493         local loader, error = loadfile(file)
494         if loader then
495             texio.write_nl("(" .. file .. ")")
496             return loader
497         end
498         return "\n\t[oberdiek.luatex.kpse_module_loader] Loading error:\n\t"
499             .. error
500     end
501     return "\n\t[oberdiek.luatex.kpse_module_loader] Search failed"
502 end
503 table.insert(package.loaders, kpse_module_loader)
504 </lua>

```

## 3 Test

```

505 <test2>
506 \documentclass{article}
507 \def\LoadCommand{%
508     \RequirePackage{luatex}[2009/04/10]%
509 }
510 </test2>
511 <test3>
512 \documentclass{article}
513 \def\LoadCommand{%
514     \RequirePackage{luatex-loader}[2009/04/10]%
515 }

```



516 </test3>

### 3.1 Catcode checks for loading

517 <test1>

```
518 \catcode'\{=1 %
519 \catcode'\}=2 %
520 \catcode'\#=6 %
521 \catcode'\@=11 %
522 \expandafter\ifx\csname count@\endcsname\relax
523   \countdef\count@=255 %
524 \fi
525 \expandafter\ifx\csname @gobble\endcsname\relax
526   \long\def\@gobble#1{}%
527 \fi
528 \expandafter\ifx\csname @firstofone\endcsname\relax
529   \long\def\@firstofone#1{#1}%
530 \fi
531 \expandafter\ifx\csname loop\endcsname\relax
532   \expandafter\@firstofone
533 \else
534   \expandafter\@gobble
535 \fi
536 {%
537   \def\loop#1\repeat{%
538     \def\body{#1}%
539     \iterate
540   }%
541   \def\iterate{%
542     \body
543     \let\next\iterate
544   \else
545     \let\next\relax
546   \fi
547   \next
548 }%
549 \let\repeat=\fi
550 }%
551 \def\RestoreCatcodes{
552 \count@=0 %
553 \loop
554   \edef\RestoreCatcodes{%
555     \RestoreCatcodes
556     \catcode\the\count@=\the\catcode\count@\relax
557   }%
558 \ifnum\count@<255 %
559   \advance\count@ 1 %
560 \repeat
561
562 \def\RangeCatcodeInvalid#1#2{%
563   \count@=#1\relax
564   \loop
565     \catcode\count@=15 %
566     \ifnum\count@<#2\relax
567       \advance\count@ 1 %
568     \repeat
569 }
570 \expandafter\ifx\csname LoadCommand\endcsname\relax
571   \def\LoadCommand{\input luatex.sty\relax}%
572 \fi
573 \def\Test{%
574   \RangeCatcodeInvalid{0}{47}%
575   \RangeCatcodeInvalid{58}{64}%
```

```

576 \RangeCatcodeInvalid{91}{96}%
577 \RangeCatcodeInvalid{123}{255}%
578 \catcode'\@=12 %
579 \catcode'\=0 %
580 \catcode'\{=1 %
581 \catcode'\}=2 %
582 \catcode'\#=6 %
583 \catcode'\[=12 %
584 \catcode'\]=12 %
585 \catcode'\%=14 %
586 \catcode'\_ =10 %
587 \catcode13=5 %
588 \LoadCommand
589 \RestoreCatcodes
590 }
591 \Test
592 \csname @@end\endcsname
593 \end
594 </test1>

```

## 3.2 Catcode tables

### 3.2.1 Predefined catcode tables

```

595 <*test4>
596 \NeedsTeXFormat{LaTeX2e}

Remember LATEX's initial catcodes in count registers starting at \TestLaTeX.
597 \count0=0 %
598 \chardef\TestLaTeX=1000 %
599 \chardef\TestMax=300 %
600 \loop
601   \count\numexpr\TestLaTeX+\count0\relax=\catcode\count0 %
602   \ifnum\count0<\TestMax
603     \advance\count0 by 1 %
604   \repeat
605 \documentclass{minimal}
606 \usepackage{luatex}[2009/04/10]
607 \usepackage{qstest}
608 \IncludeTests{*}
609 \LogTests{log}{*}{*}
610 \makeatletter
611 \def\Check#1{%
612   \Expect*{\the\count@=\the\catcode\count@}%
613   *{\the\count@=#1}%
614 }
615 \newcount\scratch
616 \def\Test#1#2{%
617   \begin{qstest}{CatcodeTable#1}{CatcodeTable#1}%
618     \catcodetable\csname CatcodeTable#1\endcsname
619     \count@=\z@
620     \loop
621       \scratch=#2\relax
622       \Expect*{\the\count@=\the\catcode\count@}%
623       *{\the\count@=\the\scratch}%
624     \ifnum\count@<\TestMax
625       \advance\count@\@ne
626     \repeat
627   \end{qstest}%
628 }
629 \Test{LaTeX}{\the\count\numexpr\TestLaTeX+\count@}
630 \Test{String}{\ifnum\count@=32 10\else 12\fi}
631 \Test{Other}{12}
632 \initcatcodetable99 %

```

```

633 \Test{IniTeX}{%
634   0\relax
635   \begingroup
636     \catcodetable99 %
637     \global\scratch=\the\catcode\count@
638   \endgroup
639 }

```

### 3.2.2 Catcode table number stack

```

640 \begin{qstest}{CatcodeTableNumStack}{CatcodeTableNumStack}
641   \def\TestStack#1{%
642     \Expect*{\LuT@NumStack}{#1}%
643   }%
644   \TestStack{0}%
645   \PushCatcodeTableNumStack
646   \TestStack{{0}0}%
647   \@firstofone{%
648     \begingroup
649       \initcatcodetable12 %
650       \catcodetable12 %
651       \PushCatcodeTableNumStack
652       \TestStack{{12}{0}0}%
653       \PopCatcodeTableNumStack
654       \TestStack{{0}0}%
655       \PopCatcodeTableNumStack
656       \TestStack{0}%
657       \def\TestWarning{Missing empty stack warning}%
658       \def\@PackageWarning#1#2{\def\TestWarning{empty stack}}%
659       \PopCatcodeTableNumStack
660       \TestStack{0}%
661       \Expect*{\TestWarning}{empty stack}%
662     \endgroup
663   }%
664 \end{qstest}

```

### 3.2.3 Catcode table stack

```

665 \begin{qstest}{CatcodeTableStack}{CatcodeTableStack}
666   \def\TestStack#1{%
667     \Expect*{\the\CatcodeTableStack}{#1}%
668   }%
669   \TestStack{0}%
670   \IncCatcodeTableStack
671   \TestStack{2}%
672   \IncCatcodeTableStack
673   \TestStack{4}%
674   \begingroup
675     \IncCatcodeTableStack
676     \TestStack{6}%
677   \endgroup
678   \TestStack{6}%
679   \begingroup
680     \DecCatcodeTableStack
681     \TestStack{4}%
682   \endgroup
683   \TestStack{4}%
684   \DecCatcodeTableStack
685   \TestStack{2}%
686   \DecCatcodeTableStack
687   \TestStack{0}%
688   \begingroup
689     \def\TestError{Missing error}%
690     \def\@PackageError#1#2#3{%
691       \def\TestError{Empty stack}%

```

```

692 }%
693 \DecCatcodeTableStack
694 \TestStack{0}%
695 \Expect*{\TestError}{Empty stack}%
696 \endgroup
697 \end{qstest}

```

### 3.2.4 Catcode regime macros

```

698 \begin{qstest}{CatcodeRegime}{CatcodeRegime}
699 \def\TestStacks#1#2#3{%
700 \Expect*{\the\catcodetable}{#1}%
701 \Expect*{\the\CatcodeTableStack}{#2}%
702 \Expect*{\LuT@NumStack}{#3}%
703 }%
704 \TestStacks{0}{0}{0}%
705 \catcode'\|=7 %
706 \BeginCatcodeRegime\CatcodeTableLaTeX
707 \TestStacks{2}{2}{{0}0}%
708 \Expect*{\the\catcode'\|}{12}%
709 \EndCatcodeRegime
710 \TestStacks{0}{0}{0}%
711 \Expect*{\the\catcode'\|}{7}%
712 \end{qstest}

```

## 3.3 Attribute allocation

```

713 \begin{qstest}{Attributes}{Attributes}
714 \newattribute\TestAttr
715 \Expect*{\meaning\TestAttr}%
716 *{\string\attribute\number\allocationnumber}%
717 \Expect*{\the\allocationnumber}{0}%
718 \begingroup
719 \newattribute\TestAttr
720 \Expect*{\the\allocationnumber}{1}%
721 \endgroup
722 \Expect*{\the\allocationnumber}{0}%
723 \Expect*{\meaning\TestAttr}*{\string\attribute1}%
724 \Expect*{\the\TestAttr}{-1}%
725 \def\Test#1{%
726 \setattribute\TestAttr{#1}%
727 \Expect*{\the\TestAttr}{#1}%
728 }%
729 \Test{0}%
730 \Test{1}%
731 \Test{-1}%
732 \Test{123}%
733 \unsetattribute\TestAttr
734 \Expect*{\the\TestAttr}{-1}%
735 \begingroup
736 \Expect*{\the\TestAttr}{-1}%
737 \Test{1234}%
738 \endgroup
739 \Expect*{\the\TestAttr}{-1}%
740 \end{qstest}

```

## 3.4 Lua states

```

741 \begin{qstest}{LuaState}{LuaState}
742 \newluastate\TestLuaState
743 \Expect*{\number\TestLuaState}{1}%
744 \newluastate\TestLuaState
745 \Expect*{\number\TestLuaState}{2}%
746 \end{qstest}
747 \@@end

```

748  $\langle$ /test4 $\rangle$

### 3.5 Short test for plain- $\text{\TeX}$

```
749  $\langle$ *test5 $\rangle$ 
750 \input luatex.sty\relax
751 \newluastate\TestLuaState
752 \newattribute\TestAttr
753 \setattribute\TestAttr{10}
754 \unsetattribute\TestAttr
755 \newcatcodetable\TestCTa
756 \begingroup
757   \SetCatcodeRange{'A'}{'Z'}{12}%
758 \endgroup
759 \BeginCatcodeRegime\CatcodeTableLaTeX
760 \EndCatcodeRegime
761 \end
762  $\langle$ /test5 $\rangle$ 
```

## 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  $\text{\TeX}$  Files” ([CTAN:tds/tds.pdf](#)). Directories with `texmf` in their name are usually organized this way.

### 4.2 Bundle installation

**Unpacking.** Unpack the `oberdiek.tds.zip` in the TDS tree (also known as `texmf` tree) of your choice. Example (linux):

```
unzip oberdiek.tds.zip -d ~/texmf
```

**Script installation.** Check the directory `TDS:scripts/oberdiek/` for scripts that need further installation steps. Package `attachfile2` comes with the Perl script `pdfatfi.pl` that should be installed in such a way that it can be called as `pdfatfi`. Example (linux):

```
chmod +x scripts/oberdiek/pdfatfi.pl
cp scripts/oberdiek/pdfatfi.pl /usr/local/bin/
```

### 4.3 Package installation

**Unpacking.** The `.dtx` file is a self-extracting docstrip archive. The files are extracted by running the `.dtx` through plain- $\text{\TeX}$ :

```
tex luatex.dtx
```

---

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

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

```

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

```

If you have a `docstrip.cfg` that configures and enables `docstrip`'s TDS installing feature, then some files can already be in the right place, see the documentation of `docstrip`.

## 4.4 Refresh file name databases

If your  $\TeX$  distribution (`te $\TeX$` , `mik $\TeX$` , ...) relies on file name databases, you must refresh these. For example, `te $\TeX$`  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  $\LaTeX$ .** The `.dtx` chooses its action depending on the format:

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

**$\LaTeX$ :** Generate the documentation.

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

```
latex \let\install=y\input{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 $\LaTeX$` :

```

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.

- Requires package `ifluatex` in version 2.0 to ensure `\luatexversion`.
- Updates the call of `\directlua`, the syntax has changed in L<sup>A</sup>T<sub>E</sub>X 0.36.

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.

23

<code>\end</code>	593, 627, 664, 697, 712, 740, 746, 761		
<code>\EndCatcodeRegime</code>	.... 380, 709, 760		
<code>\endscname</code>	8, 21, 45, 61, 68, 105, 121, 126, 148, 151, 154, 157, 160, 163, 166, 197, 198, 200, 201, 204, 385, 399, 412, 436, 452, 522, 525, 528, 531, 570, 592, 618		
<code>\endinput</code>	.... 30, 118, 421		
<code>\endlinechar</code>	.... 467		
<code>\errmessage</code>	.... 243, 256, 279		
<code>\Expect</code>	.... 612, 622, 642, 661, 667, 695, 700, 701, 702, 708, 711, 715, 717, 720, 722, 723, 724, 727, 734, 736, 739, 743, 745		
<b>F</b>			
<code>\futurelet</code>	.... 174, 193		
<b>G</b>			
<code>\globbox</code>	.... 146		
<code>\globcount</code>	.... 143		
<code>\globdimen</code>	.... 144		
<code>\globskip</code>	.... 145		
<b>I</b>			
<code>\ifcase</code>	.... 9, 218, 400		
<code>\ifluatex</code>	.... 112		
<code>\ifnum</code>	. 132, 237, 249, 270, 283, 292, 311, 468, 558, 566, 602, 624, 630		
<code>\ifx</code>	.... 10, 12, 21, 45, 53, 56, 105, 121, 126, 148, 151, 154, 157, 160, 163, 166, 177, 180, 197, 200, 204, 206, 209, 217, 363, 385, 401, 403, 412, 436, 444, 447, 522, 525, 528, 531, 570		
<code>\immediate</code>	.... 23, 47, 414, 438		
<code>\IncCatcodeTableStack</code>	.... 282, 376, 670, 672, 675		
<code>\IncludeTests</code>	.... 608		
<code>\initcatcodetable</code>	.... 321, 632, 649		
<code>\input</code>	.... 106, 107, 386, 571, 750		
<code>\iterate</code>	.... 539, 541, 543		
<b>L</b>			
<code>\LoadCommand</code>	.... 507, 513, 571, 588		
<code>\LogTests</code>	.... 609		
<code>\loop</code>	.... 310, 537, 553, 564, 600, 620		
<code>\luastate</code>	.... 241, 243		
<code>\luatexversion</code>	.... 468		
<code>\LuT@AllocAttribute</code>	246, 249, 250, 251		
<code>\LuT@AllocCatcodeTable</code>	.... 265, 270, 272, 273		
<code>\LuT@AllocLuaState</code>	. 234, 237, 238, 239		
<code>\LuT@AtEnd</code>	.... 81, 82, 117, 390		
<code>\LuT@newcommand</code>	.... 203, 236, 248, 259, 262, 269, 282, 291, 300, 357, 362, 373, 380		
<code>\LuT@NumStack</code>	.... 356, 358, 359, 363, 367, 368, 369, 642, 702		
<code>\LuT@NumStackEmpty</code>	.... 355, 356, 363		
<code>\LuT@temp</code>	. 131, 136, 137, 138, 139, 140, 141, 142, 188, 195, 301, 315		
<b>M</b>			
<code>\m@ne</code>	.... 247, 263, 266		
<code>\makeatletter</code>	.... 610		
<code>\meaning</code>	.... 715, 723		
<code>\MessageBreak</code>	.... 212		
<b>N</b>			
<code>\n</code>	.... 498, 501		
<code>\NeedsTeXFormat</code>	.... 596		
<code>\newattribute</code>	... 3, 248, 714, 719, 752		
<code>\newbox</code>	.... 146		
<code>\newcatcodetable</code>	.... 4, 269, 317, 318, 319, 320, 755		
<code>\newcommand</code>	.... 232		
<code>\newcount</code>	. 143, 234, 246, 265, 267, 615		
<code>\newdimen</code>	.... 144		
<code>\newluastate</code>	.... 3, 236, 742, 744, 751		
<code>\newskip</code>	.... 145		
<code>\next</code>	.... 543, 545, 547		
<code>\number</code>	.... 716, 743, 745		
<code>\numexpr</code>	260, 307, 308, 309, 375, 601, 629		
<b>P</b>			
<code>\PackageInfo</code>	.... 26, 417		
<code>\PopCatcodeTableNumStack</code>	.... 362, 382, 653, 655, 659		
<code>\ProvidesPackage</code>	.... 62, 453		
<code>\PushCatcodeTableNumStack</code>	.... 5, 357, 374, 645, 651		
<b>R</b>			
<code>\RangeCatcodeInvalid</code>	.... 562, 574, 575, 576, 577		
<code>\repeat</code>	314, 537, 549, 560, 568, 604, 626		
<code>\RequirePackage</code>	.... 109, 110, 123, 388, 508, 514		
<code>\reserved@a</code>	.... 172, 181		
<code>\reserved@b</code>	.... 173, 183		
<code>\reserved@c</code>	.... 178, 181, 183, 186		
<code>\reserved@d</code>	.... 171, 180		
<code>\RestoreCatcodes</code>	. 551, 554, 555, 589		
<b>S</b>			
<code>\savecatcodetable</code>	. 341, 350, 352, 377		
<code>\scratch</code>	.... 615, 621, 623, 637		
<code>\setattribute</code>	.... 3, 259, 726, 753		
<code>\SetCatcodeRange</code>	.... 5, 300, 327, 331, 348, 349, 757		
<b>T</b>			
<code>\t</code>	.... 498, 501		
<code>\Test</code>	. 573, 591, 616, 629, 630, 631, 633, 725, 729, 730, 731, 732, 737		
<code>\TestAttr</code>	.... 714, 715, 719, 723, 724, 726, 727, 733, 734, 736, 739, 752, 753, 754		
<code>\TestCTa</code>	.... 755		
<code>\TestError</code>	.... 689, 691, 695		
<code>\TestLaTeX</code>	.... 598, 601, 629		
<code>\TestLuaState</code>	. 742, 743, 744, 745, 751		
<code>\TestMax</code>	.... 599, 602, 624		
<code>\TestStack</code>	641, 644, 646, 652, 654, 656, 660, 666, 669, 671, 673, 676, 678, 681, 683, 685, 687, 694		



<code>\TestStacks</code> . . . . .	699, 704, 707, 710	<code>\unsetattribute</code> . .	4, 253, <u>262</u> , 733, 754
<code>\TestWarning</code> . . . . .	657, 658, 661	<code>\usepackage</code> . . . . .	606, 607
<code>\the</code> . . . . .	69, 70, 71, 72, 83, 241, 254, 276, 302, 303, 304, 359, 556, 612, 613, 622, 623, 629, 637, 667, 700, 701, 708, 711, 717, 720, 722, 724, 727, 734, 736, 739		
<code>\TMP@EnsureCode</code> . . . . .	80, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103		
<code>\tw@</code> . . . . .	272, 284		
		<b>W</b>	
		<code>\wlog</code> . . . . .	241, 254, 275
		<code>\write</code> . . . . .	23, 47, 414, 438
		<b>X</b>	
		<code>\x</code> . . . . .	8, 10, 12, 22, 26, 28, 46, 51, 61, 67, 75, 399, 401, 403, 413, 417, 419, 437, 442, 452
		<b>U</b>	
<code>\unless</code> . . . . .	311	<b>Z</b>	
		<code>\z@</code> . . . . .	235, 268, 292, 365, 619