

# 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  $\epsilon$ -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  $\epsilon$ -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 L<sup>A</sup>T<sub>E</sub>X 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 L<sup>A</sup>T<sub>E</sub>X (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 \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 [2009/04/10 v0.2 LuaTeX basic definition package (HO)]

```

## 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 Inherit support for $\varepsilon$ -TeX

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

```

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

```

## 2.5 Adaption of $\varepsilon$ -TeX’s register allocation

$\varepsilon$ -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.

```

124 \begingroup\expandafter\expandafter\expandafter\endgroup
125 \expandafter\ifx\csname et@xins\endcsname\relax
126   \@PackageWarningNoLine{luatex}{%
127     Support for eTeX is not loaded (etex.src)%
128   }%
129 \else
130   \def\LuT@temp#1{%
131     \ifnum\count27#1=32768 %
132       \count27#1=65536 %
133     \fi
134   }%
135   \LuT@temp0%
136   \LuT@temp1%

```



```

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

```

$\varepsilon$ -T<sub>E</sub>X 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.

```

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

```

## 2.6 plain-T<sub>E</sub>X compatibility

`\@empty`

```

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

```

`\@gobble`

```

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

```

`\@firstofone`

```

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

```

`\@firstoftwo`

```

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

```

`\@car`

```

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

```

`\@cdr`

```

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

```

`\@ifstar`

```

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

```

`\@ifnextchar`

```

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

```

```

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

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

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

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

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

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

```

```

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

```

## 2.7 Lua states

\LuT@AllocLuaState

```

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

```

\newluastate

```

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

```

## 2.8 Attributes

### 2.8.1 Allocation

\LuT@AllocAttribute

```

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

```

\newattribute

```

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

```

### 2.8.2 Interface

\setattribute

```

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

```

\unsetattribute

```

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

```

## 2.9 Catcode tables

### 2.9.1 Allocation

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

\newcatcodetable
268 \LuT@newcommand\newcatcodetable[1]{%
269   \ifnum\LuT@AllocCatcodeTable<1114110 % 0x10FFF is maximal \chardef
270     % or < 268435455 %  $2^{28} - 1$ 
271   \global\advance\LuT@AllocCatcodeTable by\tw@
272   \allocationnumber=\LuT@AllocCatcodeTable
273   \global\chardef#1=\allocationnumber
274   \wlog{%
275     \string#1=\string\catcodetable\the\allocationnumber
276   }%
277   \else
278     \errmessage{No room for a new \string\catcodetable}%
279   \fi
280 }%

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

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

### 2.9.2 \SetCatcodeRange

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

```

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

```

### 2.9.3 Predefined catcode tables

```

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

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

```

### 2.9.4 Number stack

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

```

354 \def\LuT@NumStackEmpty{0}

```

`\LuT@NumStack`

```

355 \let\LuT@NumStack\LuT@NumStackEmpty

```

`\PushCatcodeTableNumStack`

```

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

```

\PopCatcodeTableNumStack

```

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

```

## 2.9.5 Catcode regime macros

\BeginCatcodeRegime

```

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

```

\EndCatcodeRegime

```

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

```

## 2.10 Lua module loader

```

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

389 \LuT@AtEnd
390 </package>

391 <*loader>

```

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

```

392 \begingroup
393   \catcode44 12 % ,
394   \catcode45 12 % -
395   \catcode46 12 % .
396   \catcode58 12 % :
397   \catcode64 11 % @
398   \catcode123 1 % {
399   \catcode125 2 % }
400   \expandafter\let\expandafter\x\csname ver@luatex-loader.sty\endcsname
401   \ifx\x\relax % plain-TeX, first loading
402   \else
403     \def\empty{%
404       \ifx\x\empty % LaTeX, first loading,
405       % variable is initialized, but \ProvidesPackage not yet seen
406     \else
407       \catcode35 6 % #
408       \expandafter\ifx\csname PackageInfo\endcsname\relax
409         \def\x#1#2{%

```

```

410         \immediate\write-1{Package #1 Info: #2.}%
411     }%
412     \else
413         \def\x#1#2{\PackageInfo{#1}{#2, stopped}}%
414     \fi
415     \x{luatex-loader}{The package is already loaded}%
416     \aftergroup\endinput
417 \fi
418 \fi
419 \endgroup
Package identification:
420 \begingroup
421     \catcode35 6 % #
422     \catcode40 12 % (
423     \catcode41 12 % )
424     \catcode44 12 % ,
425     \catcode45 12 % -
426     \catcode46 12 % .
427     \catcode47 12 % /
428     \catcode58 12 % :
429     \catcode64 11 % @
430     \catcode91 12 % [
431     \catcode93 12 % ]
432     \catcode123 1 % {
433     \catcode125 2 % }
434     \expandafter\ifx\csname ProvidesPackage\endcsname\relax
435         \def\x#1#2#3[#4]{\endgroup
436             \immediate\write-1{Package: #3 #4}%
437             \xdef#1{#4}%
438         }%
439     \else
440         \def\x#1#2[#3]{\endgroup
441             #2[#{#3}]%
442             \ifx#1\@undefined
443                 \xdef#1{#3}%
444             \fi
445             \ifx#1\relax
446                 \xdef#1{#3}%
447             \fi
448         }%
449     \fi
450 \expandafter\x\csname ver@luatex-loader.sty\endcsname
451 \ProvidesPackage{luatex-loader}%
452 [2009/04/10 v0.2 Lua module loader (H0)]
453 \begingroup
454     \catcode10 12 % ^^J
455     \catcode34 12 % "
456     \catcode39 12 % '
457     \catcode40 12 % (
458     \catcode41 12 % )
459     \catcode44 12 % ,
460     \catcode46 12 % .
461     \catcode60 12 % <
462     \catcode61 12 % =
463     \catcode95 12 % _ (other!)
464     \catcode96 12 % `
465     \endlinechar=10 %
466     \ifnum\luatexversion<36 %
467         \directlua{}
468     \else %
469         \expandafter\directlua %
470     \fi %

```

```

471 {%
472     do
473         local script = "oberdiek.luatex.lua"
474         local file = kpse.find_file(script, "texmfscripts")
475         if file then
476             texio.write_nl("(" .. file .. ")")
477             dofile(file)
478         else
479             error("File '" .. script .. "' not found")
480         end
481     end
482 }%
483 \endgroup%
484 \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`.

```

485 (*lua)

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

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

```

## 3 Test

```

503 (*test2)
504 \documentclass{article}
505 \def\LoadCommand{%
506     \RequirePackage{luatex}[2009/04/10]%
507 }
508 \test2

509 (*test3)
510 \documentclass{article}
511 \def\LoadCommand{%
512     \RequirePackage{luatex-loader}[2009/04/10]%
513 }
514 \test3

```



### 3.1 Catcode checks for loading

```
515 (*test1)
516 \catcode'\{=1 %
517 \catcode'\}=2 %
518 \catcode'\#=6 %
519 \catcode'\@=11 %
520 \expandafter\ifx\csname count@\endcsname\relax
521 \countdef\count@=255 %
522 \fi
523 \expandafter\ifx\csname @gobble\endcsname\relax
524 \long\def\@gobble#1{}%
525 \fi
526 \expandafter\ifx\csname @firstofone\endcsname\relax
527 \long\def\@firstofone#1{#1}%
528 \fi
529 \expandafter\ifx\csname loop\endcsname\relax
530 \expandafter\@firstofone
531 \else
532 \expandafter\@gobble
533 \fi
534 {%
535 \def\loop#1\repeat{%
536 \def\body{#1}%
537 \iterate
538 }%
539 \def\iterate{%
540 \body
541 \let\next\iterate
542 \else
543 \let\next\relax
544 \fi
545 \next
546 }%
547 \let\repeat=\fi
548 }%
549 \def\RestoreCatcodes{}
550 \count@=0 %
551 \loop
552 \edef\RestoreCatcodes{%
553 \RestoreCatcodes
554 \catcode\the\count@=\the\catcode\count@\relax
555 }%
556 \ifnum\count@<255 %
557 \advance\count@ 1 %
558 \repeat
559
560 \def\RangeCatcodeInvalid#1#2{%
561 \count@=#1\relax
562 \loop
563 \catcode\count@=15 %
564 \ifnum\count@<#2\relax
565 \advance\count@ 1 %
566 \repeat
567 }
568 \expandafter\ifx\csname LoadCommand\endcsname\relax
569 \def\LoadCommand{\input luatex.sty\relax}%
570 \fi
571 \def\Test{%
572 \RangeCatcodeInvalid{0}{47}%
573 \RangeCatcodeInvalid{58}{64}%
574 \RangeCatcodeInvalid{91}{96}%
575 \RangeCatcodeInvalid{123}{255}%
```

```

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

```

## 3.2 Catcode tables

### 3.2.1 Predefined catcode tables

```

593 <*test4>
594 \NeedsTeXFormat{LaTeX2e}

```

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

```

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

```

```

633 \begingroup
634   \catcodetable99 %
635   \global\scratch=\the\catcode\count@
636 \endgroup
637 }

```

### 3.2.2 Catcode table number stack

```

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

```

### 3.2.3 Catcode table stack

```

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

```

```

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

```

### 3.2.4 Catcode regime macros

```

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

```

## 3.3 Attribute allocation

```

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

```

## 3.4 Lua states

```

739 \begin{qstest}{LuaState}{LuaState}
740 \newluaState\TestLuaState
741 \Expect*{\number\TestLuaState}{1}%
742 \newluaState\TestLuaState
743 \Expect*{\number\TestLuaState}{2}%
744 \end{qstest}

745 @@end
746 /test4

```

### 3.5 Short test for plain-TeX

```
747 <*test5>
748 \input luatex.sty\relax
749 \newluastate\TestLuaState
750 \newattribute\TestAttr
751 \setattribute\TestAttr{10}
752 \unsetattribute\TestAttr
753 \newcatcodetable\TestCTa
754 \begingroup
755   \SetCatcodeRange{'A'}{'Z'}{12}%
756 \endgroup
757 \BeginCatcodeRegime\CatcodeTableLaTeX
758 \EndCatcodeRegime
759 \end
760 </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 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-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 $\TeX$ , mi $\TeX$ , ...) 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.

[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 L<sup>A</sup>T<sub>E</sub>X 0.36.

## 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	B
<code>\#</code> ..... 518, 580	<code>\begin</code> .... 615, 638, 663, 696, 711, 739
<code>\%</code> ..... 583	<code>\BeginCatcodeRegime</code> .. 5, <u>372</u> , 704, 757
<code>\:</code> ..... 187, 188, 189, 190, 191, 194	<code>\body</code> ..... 536, 540
<code>\@</code> ..... 519, 576	
<code>\@@end</code> ..... 745	
<code>\@INTERNAL@ERROR</code> ..... 226	
<code>\@PackageError</code> 113, 210, 285, 294, 688	
<code>\@PackageWarning</code> ..... 363, 656	
<code>\@PackageWarningNoLine</code> ..... 126	
<code>\@car</code> ..... <u>159</u> , 366	
<code>\@cdr</code> ..... <u>162</u> , 368	
<code>\@ehc</code> ..... 115, 213	
<code>\@ehd</code> ..... 287, 296	
<code>\@empty</code> ..... <u>147</u>	
<code>\@firstofone</code> .. <u>153</u> , 323, 527, 530, 645	
<code>\@firstoftwo</code> ..... <u>156</u> , 167	
<code>\@gobble</code> ..... <u>150</u> , 524, 532	
<code>\@ifnch</code> ..... 173, <u>175</u> , 192	
<code>\@ifnextchar</code> ..... 167, <u>169</u>	
<code>\@ifstar</code> ..... <u>165</u>	
<code>\@let@token</code> ..... 173, 176, 179, 192	
<code>\@makeoother</code> ..... ..... 322, 342, 343, 344, 345, 346, 350	
<code>\@one</code> ..... 237, 249, 312, 623	
<code>\@nil</code> ..... 160, 163, 366, 368	
<code>\@sptoken</code> ..... 176, <u>187</u>	
<code>\@tempcnta</code> <u>196</u> , 301, 306, 310, 311, 312	
<code>\@tempcntb</code> ..... <u>199</u> , 302, 307, 310	
<code>\@undefined</code> ..... 52, 205, 442	
<code>\@xifnch</code> ..... 177, <u>190</u>	
<code>\[</code> ..... 581	
<code>\]</code> ..... 577	
<code>\{</code> ..... 516, 578	
<code>\}</code> ..... 517, 579	
<code>\]</code> ..... 582	
<code>\ </code> ..... 703, 706, 709	
<code>\_</code> ..... 584	
A	C
<code>\advance</code> ..... 237, 249, 271, 283, 292, 312, 557, 565, 601, 623	<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, 311, 322, 327, 328, 329, 331, 332, 333, 334, 335, 336, 337, 338, 339, 393, 394, 395, 396, 397, 398, 399, 407, 421, 422, 423, 424, 425, 426, 427, 428, 429, 430, 431, 432, 433, 454, 455, 456, 457, 458, 459, 460, 461, 462, 463, 464, 516, 517, 518, 519, 554, 563, 576, 577, 578, 579, 580, 581, 582, 583, 584, 585, 599, 610, 620, 635, 703, 706, 709
<code>\aftergroup</code> ..... 26, 416	<code>\catcodetable</code> 275, 278, 324, 358, 364, 366, 374, 377, 616, 634, 648, 698
<code>\allocationnumber</code> ..... ..... 238, 239, 240, 250, 251, 253, 272, 273, 275, 714, 715, 718, 720	<code>\CatcodeTableIniTeX</code> .. 4, 316, 320, 324
<code>\attribute</code> ..... 253, 255, 714, 721	<code>\CatcodeTableLaTeX</code> . 319, 340, 704, 757
<code>\attributedef</code> ..... 251	<code>\CatcodeTableOther</code> ..... 318, 351
	<code>\CatcodeTableStack</code> 4, 266, 267, 282, 283, 291, 292, 376, 377, 665, 699
	<code>\CatcodeTableString</code> ..... 317, 349
	<code>\chardef</code> ..... 239, 269, 273, 596, 597
	<code>\Check</code> ..... 609
	<code>\count</code> 131, 132, 595, 599, 600, 601, 627
	<code>\count@</code> ..... 303, 308, 311, 521, 550, 554, 556, 557, 561, 563, 564, 565, 610, 611, 617, 620, 621, 622, 623, 627, 628, 635
	<code>\countdef</code> ..... 521
	<code>\csname</code> 10, 18, 44, 60, 67, 104, 120, 125, 147, 150, 153, 156, 159, 162, 165, 196, 197, 199, 200, 203, 384, 400, 408, 434, 450, 520, 523, 526, 529, 568, 590, 616
	D
	<code>\DecCatcodeTableStack</code> ..... ..... <u>290</u> , 380, 678, 682, 684, 691
	<code>\directlua</code> ..... 467, 469
	<code>\documentclass</code> ..... 504, 510, 603
	E
	<code>\empty</code> ..... 13, 14, 403, 404

<code>\end</code>	591, 625, 662, 695, 710, 738, 744, 759
<code>\EndCatcodeRegime</code>	379, 707, 758
<code>\endcsname</code>	10, 18, 44, 60, 67, 104, 120, 125, 147, 150, 153, 156, 159, 162, 165, 196, 197, 199, 200, 203, 384, 400, 408, 434, 450, 520, 523, 526, 529, 568, 590, 616
<code>\endinput</code>	26, 117, 416
<code>\endlinechar</code>	465
<code>\errmessage</code>	242, 255, 278
<code>\Expect</code>	610, 620, 640, 659, 665, 693, 698, 699, 700, 706, 709, 713, 715, 718, 720, 721, 722, 725, 732, 734, 737, 741, 743
<b>F</b>	
<code>\futurelet</code>	173, 192
<b>G</b>	
<code>\gloabox</code>	145
<code>\globcount</code>	142
<code>\globdimen</code>	143
<code>\globskip</code>	144
<b>I</b>	
<code>\ifcase</code>	217
<code>\ifluatex</code>	111
<code>\ifnum</code>	131, 236, 248, 269, 282, 291, 310, 466, 556, 564, 600, 622, 628
<code>\ifx</code>	11, 14, 18, 44, 52, 55, 104, 120, 125, 147, 150, 153, 156, 159, 162, 165, 176, 179, 196, 199, 203, 205, 208, 216, 362, 384, 401, 404, 408, 434, 442, 445, 520, 523, 526, 529, 568
<code>\immediate</code>	20, 46, 410, 436
<code>\IncCatcodeTableStack</code>	281, 375, 668, 670, 673
<code>\IncludeTests</code>	606
<code>\initcatcodetable</code>	320, 630, 647
<code>\input</code>	105, 106, 385, 569, 748
<code>\iterate</code>	537, 539, 541
<b>L</b>	
<code>\LoadCommand</code>	505, 511, 569, 586
<code>\LogTests</code>	607
<code>\loop</code>	309, 535, 551, 562, 598, 618
<code>\luastate</code>	240, 242
<code>\luatexversion</code>	466
<code>\LuT@AllocAttribute</code>	245, 248, 249, 250
<code>\LuT@AllocCatcodeTable</code>	264, 269, 271, 272
<code>\LuT@AllocLuaState</code>	233, 236, 237, 238
<code>\LuT@AtEnd</code>	80, 81, 116, 389
<code>\LuT@newcommand</code>	202, 235, 247, 258, 261, 268, 281, 290, 299, 356, 361, 372, 379
<code>\LuT@NumStack</code>	355, 357, 358, 362, 366, 367, 368, 640, 700
<code>\LuT@NumStackEmpty</code>	354, 355, 362
<code>\LuT@temp</code>	130, 135, 136, 137, 138, 139, 140, 141, 187, 194, 300, 314
<b>M</b>	
<code>\m@ne</code>	246, 262, 265
<code>\makeatletter</code>	608
<code>\meaning</code>	713, 721
<code>\MessageBreak</code>	211
<b>N</b>	
<code>\n</code>	496, 499
<code>\NeedsTeXFormat</code>	594
<code>\newattribute</code>	3, 247, 712, 717, 750
<code>\newbox</code>	145
<code>\newcatcodetable</code>	4, 268, 316, 317, 318, 319, 753
<code>\newcommand</code>	231
<code>\newcount</code>	142, 233, 245, 264, 266, 613
<code>\newdimen</code>	143
<code>\newluastate</code>	3, 235, 740, 742, 749
<code>\newskip</code>	144
<code>\next</code>	541, 543, 545
<code>\number</code>	714, 741, 743
<code>\numexpr</code>	259, 306, 307, 308, 374, 599, 627
<b>P</b>	
<code>\PackageInfo</code>	23, 413
<code>\PopCatcodeTableNumStack</code>	361, 381, 651, 653, 657
<code>\ProvidesPackage</code>	15, 61, 405, 451
<code>\PushCatcodeTableNumStack</code>	5, 356, 373, 643, 649
<b>R</b>	
<code>\RangeCatcodeInvalid</code>	560, 572, 573, 574, 575
<code>\repeat</code>	313, 535, 547, 558, 566, 602, 624
<code>\RequirePackage</code>	108, 109, 122, 387, 506, 512
<code>\reserved@a</code>	171, 180
<code>\reserved@b</code>	172, 182
<code>\reserved@c</code>	177, 180, 182, 185
<code>\reserved@d</code>	170, 179
<code>\RestoreCatcodes</code>	549, 552, 553, 587
<b>S</b>	
<code>\savecatcodetable</code>	340, 349, 351, 376
<code>\scratch</code>	613, 619, 621, 635
<code>\setattribute</code>	3, 258, 724, 751
<code>\SetCatcodeRange</code>	5, 299, 326, 330, 347, 348, 755
<b>T</b>	
<code>\t</code>	496, 499
<code>\Test</code>	571, 589, 614, 627, 628, 629, 631, 723, 727, 728, 729, 730, 735
<code>\TestAttr</code>	712, 713, 717, 721, 722, 724, 725, 731, 732, 734, 737, 750, 751, 752
<code>\TestCTa</code>	753
<code>\TestError</code>	687, 689, 693
<code>\TestLaTeX</code>	596, 599, 627
<code>\TestLuaState</code>	740, 741, 742, 743, 749
<code>\TestMax</code>	597, 600, 622
<code>\TestStack</code>	639, 642, 644, 650, 652, 654, 658, 664, 667, 669, 671, 674, 676, 679, 681, 683, 685, 692



<code>\TestStacks</code> . . . . .	697, 702, 705, 708	<code>\unsetattribute</code> . .	4, 252, <u>261</u> , 731, 752
<code>\TestWarning</code> . . . . .	655, 656, 659	<code>\usepackage</code> . . . . .	604, 605
<code>\the</code> . . .	68, 69, 70, 71, 82, 240, 253, 275, 301, 302, 303, 358, 554, 610, 611, 620, 621, 627, 635, 665, 698, 699, 706, 709, 715, 718, 720, 722, 725, 732, 734, 737		
<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> . . . . .	271, 283		
		<b>W</b>	
		<code>\wlog</code> . . . . .	240, 253, 274
		<code>\write</code> . . . . .	20, 46, 410, 436
		<b>X</b>	
		<code>\x</code> . . . . .	10, 11, 14, 19, 23, 25, 45, 50, 60, 66, 74, 400, 401, 404, 409, 413, 415, 435, 440, 450
		<b>U</b>	
<code>\unless</code> . . . . .	310	<b>Z</b>	
		<code>\z@</code> . . . . .	234, 267, 291, 364, 617