

# The `luatexbase-cctb` package

Heiko Oberdiek (primary author of `luatex`)  
Élie Roux, Manuel Pégourié-Gonnard, Philipp Gesang\*

<https://github.com/lualatex/luatexbase>  
[luatex-dev@tug.org](mailto:luatex-dev@tug.org)

v0.6 2013-05-11

## Abstract

In addition to the registers existing in  $\mathrm{T}_{\mathrm{E}}\mathrm{X}$  and  $\varepsilon\text{-}\mathrm{T}_{\mathrm{E}}\mathrm{X}$ ,  $\mathrm{LuaT}_{\mathrm{E}}\mathrm{X}$  introduces a new concept: catcode tables. This package takes care of catcode table allocation just like Plain  $\mathrm{T}_{\mathrm{E}}\mathrm{X}$  and  $\mathrm{LaT}_{\mathrm{E}}\mathrm{X}$  do for other registers. It also provides a few handy macros from common use cases.

## Contents

<b>1</b>	<b>Documentation</b>	<b>1</b>
1.1	Allocation and setting	1
1.2	Access from Lua	2
1.3	High-level stack interface	3
1.4	Low-level stack interface	3
<b>2</b>	<b>Implementation</b>	<b>4</b>
2.1	$\mathrm{T}_{\mathrm{E}}\mathrm{X}$ package	4
2.1.1	Preliminaries	4
2.1.2	Load supporting Lua module	5
2.1.3	Primitives needed	5
2.1.4	Allocation and temporary tables	5
2.1.5	Setting catcode tables	6
2.1.6	Predefined tables	7
2.1.7	Catcode table number stack and regimes	8
2.2	Lua module	9
<b>3</b>	<b>Test files</b>	<b>10</b>

## 1 Documentation

### 1.1 Allocation and setting

The main macro defined here is `\newluatexcatcodetable`. It behaves the same as `\newbox`. Additionally, the newly allocated catcode table is initialised to the catcodes of  $\mathrm{IniT}_{\mathrm{E}}\mathrm{X}$ . In order to help you define the catcode tables (once they are allocated), two helper macros are available.

---

\*See “History” in [luatexbase.pdf](#) for details.

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

Set all characters code in the range  $\langle from \rangle$ – $\langle to \rangle$  to the given catcode  $\langle value \rangle$ .

`\setluatexcatcodetable{⟨table⟩}{⟨catcode statements⟩}`

Set a previously allocated  $\langle table \rangle$  to the catcodes given by executing  $\langle catcode statements \rangle$ . Note that  $\langle table \rangle$  must not be the current active catcode table. You may, however, load another catcode table in your  $\langle catcode statements \rangle$  (if you don't, then the current catcodes will be used as a basis).

For your convenience, a few catcode tables are predefined:

- `\CatcodeTableIniTeX`: IniTeX catcodes.
- `\CatcodeTableString`: the catcode regime used by `\string` and `\meaning`: everything has catcode 12, except space (U+0020) that has catcode 10.
- `\CatcodeTableOther`: everything (included space) has catcode 12.
- `\CatcodeTableLaTeX`: basic L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> catcodes.
- `\CatcodeTableLaTeXAtLetter`: same as above, but @ is a letter.
- `\CatcodeTableExpl`: catcodes used by L<sup>A</sup>T<sub>E</sub>X 3 with `\ExplSyntaxOn`. Be aware that this does not provide the exact same environment as `\ExplSyntaxOn`: most noticeably, some booleans are not set, and `\endlinechar` is not adjusted (it should be 32).

## 1.2 Access from Lua

Various Lua functions, such as `tex.print` accept a catcode table number as argument. In order to use in Lua the catcode tables allocated in T<sub>E</sub>X, the package writer needs to know their number. Since `\chardef` is used for the definition of the control sequence, this is rather easy to do. However, for extra ease of use, the numbers are also directly accessible from Lua as the value of the table `luatexbase.catcodetables`, whose keys is the name of the control sequence (without any leading backslash). For example, after

`\newluatexcatcodetable\mycatcodes`

the variable `luatexbase.catcodetables.mycatcodes` will hold the number of the catcode table `\mycatcodes`. You will usually want to do

`local mycatcodes = luatexbase.catcodetables.mycatcodes`

at the beginning of your Lua module (assuming it is loaded after the catcode table has been allocated) and then use `tex.print(mycatcodes, ...)` in the rest of your Lua code.

Also, nicknames are available for the predefined catcode tables:

- `CatcodeTableIniTeX = ini`,
- `CatcodeTableString = string`,
- `CatcodeTableOther = other`,
- `CatcodeTableLaTeX = latex`,
- `CatcodeTableLaTeXAtLetter = latex-atletter = latex-package`,
- `CatcodeTableExpl = expl = expl3`,

### 1.3 High-level stack interface

In many situations, you want to ensure a proper set of catcodes for some time, then go back to the previous catcodes without using a group. The obvious use case is a package, which may be loaded while unexpected catcodes are active. A pair of macros is provided to deal precisely with this situation.

```
\BeginCatcodeRegime{<catcode table>}  
<code>  
\EndCatcodeRegime
```

`\BeginCatcodeRegime` remembers the current catcode table, then initializes a new one with a copy of the `<catcode table>` given (so that this table is not changed by anything in `<code>`), and makes it the active table. `\EndCatcodeRegime` restores the previous catcode table.

For example, a L<sup>A</sup>T<sub>E</sub>X package might do:

```
\BeginCatcodeRegime\CatcodeTableLaTeXAtLetter  
% package code  
\EndCatcodeRegime
```

Every catcode change made in the package code will be lost avec `\EndCatcodeRegime`; if you want them to survive, please use `\AtEndOfPackage` or `\AtBeginDocument`.

By the way, you may add catcode statements in the argument of `\BeginCatcodeRegime` after the name of a catcode table. So the first line of the previous example is equivalent to

```
\BeginCatcodeRegime{\CatcodeTableLaTeX \makeatletter}
```

### 1.4 Low-level stack interface

The previous macros use internally two stacks : the first one holds the numbers of the active catcodes tables, so remembering the current table means pushing its number on the stack, and restoring the previously active table means popping it off the stack. This is done with:

```
\PushCatcodeTableNumStack  
\PopCatcodeTableNumStack
```

Note that `\PushCatcodeTableNumStack` doesn't change current catcodes but `\PopCatcodeTableNumStack` does.

The second stack consists of temporary tables, mainly used to hold copies of existing tables. A temporary table are allocated with `\IncCatcodeTableStack`, accessed with `\CatcodeTableStack` and the freed with `\DecCatcodeTableStack`, none of which changes the current table.

As an illustration, `\BeginCatcodeRegime` does

```
\PushCatcodeTableNumStack  
\IncCatcodeTableStack  
\setluatexcatcodetable\CatcodeTableStack{\luatexcatcodetable#1}%  
\luatexcatcodetable\CatcodeTableStack
```

and `\EndCatcodeRegime` is defined as

```
\DecCatcodeTableStack  
\PopCatcodeTableNumStack
```

If you choose to use this low-level interface, it is your responsibility to ensure proper balancing of push-pop and inc-dec, as well as making sure that a temporary table is never used after it has been freed.

## 2 Implementation

### 2.1 T<sub>E</sub>X package

1 (\*texpackage)

#### 2.1.1 Preliminaries

Catcode defenses and reload protection.

```
2 \begingroup\catcode61\catcode48\catcode32=10\relax% = and space
3 \catcode123 1 % {
4 \catcode125 2 % }
5 \catcode 35 6 % #
6 \toks0\expandafter{\expandafter\endlinechar\the\endlinechar}%
7 \edef\x{\endlinechar13}%
8 \def\y#1 #2 {%
9 \toks0\expandafter{\the\toks0 \catcode#1 \the\catcode#1}%
10 \edef\x{\x \catcode#1 #2}}%
11 \y 13 5 % carriage return
12 \y 61 12 % =
13 \y 32 10 % space
14 \y 123 1 % {
15 \y 125 2 % }
16 \y 35 6 % #
17 \y 64 11 % @ (letter)
18 \y 10 12 % new line ^^J
19 \y 39 12 % '
20 \y 40 12 % (
21 \y 41 12 % )
22 \y 44 12 % ,
23 \y 45 12 % -
24 \y 46 12 % .
25 \y 47 12 % /
26 \y 58 12 % :
27 \y 60 12 % <
28 \y 62 12 % >
29 \y 91 12 % [
30 \y 93 12 % ]
31 \y 94 7 % ^
32 \y 95 8 % _
33 \y 96 12 % `
34 \toks0\expandafter{\the\toks0 \relax\noexpand\endinput}%
35 \edef\y#1{\noexpand\expandafter\endgroup%
36 \noexpand\ifx#1\relax \edef#1{\the\toks0}\x\relax%
37 \noexpand\else \noexpand\expandafter\noexpand\endinput%
38 \noexpand\fi}%
39 \expandafter\y\cename luatexbase@cctb@sty@endinput\endcename%
```

Package declaration.

```
40 \begingroup
41 \expandafter\ifx\cename ProvidesPackage\endcename\relax
42 \def\x#1[#2]{\immediate\write16{Package: #1 #2}}
43 \else
44 \let\x\ProvidesPackage
```

```

45 \fi
46 \expandafter\endgroup
47 \x{luatexbase-cctb}[2013/05/11 v0.6 Catcodetable allocation for LuaTeX]

    Make sure LuaTeX is used.

48 \begingroup\expandafter\expandafter\expandafter\endgroup
49 \expandafter\ifx\csname RequirePackage\endcsname\relax
50 \input ifluatex.sty
51 \else
52 \RequirePackage{ifluatex}
53 \fi
54 \ifluatex\else
55 \begingroup
56 \expandafter\ifx\csname PackageError\endcsname\relax
57 \def\x#1#2#3{\begingroup \newlinechar10
58 \errhelp{#3}\errmessage{Package #1 error: #2}\endgroup}
59 \else
60 \let\x\PackageError
61 \fi
62 \expandafter\endgroup
63 \x{luatexbase-cctb}{LuaTeX is required for this package. Aborting.}{%
64 This package can only be used with the LuaTeX engine^^J%
65 (command 'lualatex' or 'luatex').^^J%
66 Package loading has been stopped to prevent additional errors.}
67 \expandafter\luatexbase@cctb@sty@endinput%
68 \fi

```

### 2.1.2 Load supporting Lua module

First load `luatexbase-modutils` (hence `luatexbase-loader` and `luatexbase-compat`), then the supporting Lua module.

```

69 \begingroup\expandafter\expandafter\expandafter\endgroup
70 \expandafter\ifx\csname RequirePackage\endcsname\relax
71 \input luatexbase-modutils.sty
72 \else
73 \RequirePackage{luatexbase-modutils}
74 \fi
75 \luatexbase@directlua{require('luatexbase.cctb')}

```

### 2.1.3 Primitives needed

`luatexbase-compat` is already loaded; just make sure the primitives we need are available.

```

76 \luatexbase@ensure@primitive{luaescapestring}
77 \luatexbase@ensure@primitive{catcodetable}
78 \luatexbase@ensure@primitive{initcatcodetable}
79 \luatexbase@ensure@primitive{savecatcodetable}

```

### 2.1.4 Allocation and temporary tables

Counters for the allocated and temporary tables.

```

80 \newcount\lltxb@catcodetable@alloc
81 \lltxb@catcodetable@alloc\m@ne

```

```

82 \newcount\CatcodeTableStack
83 \CatcodeTableStack\z@

```

The allocation macro. Allocate only odd numbers to allow using even numbers for temporary tables.

```

84 \def\newluatexcatcodetable#1{%
85   \ifnum\llt\b@catcodetable@alloc<65535\relax
86     \global\advance\llt\b@catcodetable@alloc\tw@
87     \allocationnumber\llt\b@catcodetable@alloc
88     \global\chardef#1\allocationnumber
89     \luatexinitcatcodetable\allocationnumber
90     \begingroup\escapechar\m@ne
91     \luatexbase@directlua{luatexbase.catcodetabledef_from_tex(
92       '\luatexluaescapestring{\string#1}', '\number\allocationnumber')}%
93     \endgroup
94     \wlog{\string#1=\string\luatexcatcodetable\the\allocationnumber}%
95   \else
96     \errmessage{No room for a new \string\luatexcatcodetable}%
97   \fi}

```

Allocate a new temporary table, using even numbers.

```

98 \def\IncCatcodeTableStack{%
99   \ifnum\CatcodeTableStack<65534\relax
100     \global\advance\CatcodeTableStack\tw@
101   \else
102     \errmessage{CatcodeTableStack overflow}%
103   \fi}

```

Free a temporary table.

```

104 \def\DecCatcodeTableStack{%
105   \ifnum\CatcodeTableStack>\z@
106     \global\advance\CatcodeTableStack-2\relax
107   \else
108     \errmessage{CatcodeTableStack underflow}%
109   \fi}

```

### 2.1.5 Setting catcode tables

A small patch to get two new counters in Plain too.

```

110 \expandafter\ifx\csname @tempcnta\endcsname\relax
111   \csname newcount\endcsname\@tempcnta
112 \fi
113 \expandafter\ifx\csname @tempcntb\endcsname\relax
114   \csname newcount\endcsname\@tempcntb
115 \fi

```

Set the catcodes for a range of characters.

```

116 \def\SetCatcodeRange#1#2#3{%
117   \edef\luaSCR@temp{%
118     \noexpand\@tempcnta=\the\@tempcnta
119     \noexpand\@tempcntb=\the\@tempcntb
120     \noexpand\count@=\the\count@
121     \relax}%

```

```

122 \@tempcnta=#1\relax
123 \@tempcntb=#2\relax
124 \count@=#3\relax
125 \loop\unless\ifnum\@tempcnta>\@tempcntb
126   \catcode\@tempcnta=\count@
127   \advance\@tempcnta\@ne
128 \repeat
129 \luaSCR@temp}

```

Make \SetCatcodeRange available with its old name.

```

130 \def\lltxb@synonym#1#2{%
131   \def#1{%
132     \begingroup
133     \ifdefined\PackageWarning
134       \let\x\PackageWarning
135     \else
136       \def\x####1####2{%
137         \begingroup \endlinechar10
138         \message{Package ####1 warning: ####2}%
139       \endgroup}
140   \fi
141   \expandafter\endgroup
142   \x{luatex}{\string#1 is deprecated^^J%
143     and will be removed in the next version of the package.^^J%
144     Please use \string#2 instead.}
145   \let#1#2%
146   #1}}
147 \lltxb@synonym \setcatcoderange \SetCatcodeRange

```

Set a catcode table.

```

148 \def\setluatexcatcodetable#1#2{%
149   \begingroup
150   #2%
151   \luatexsavecatcodetable#1%
152   \endgroup}

```

### 2.1.6 Predefined tables

The IniTeX catcode table needs no extra initialisation.

```

153 \newluatexcatcodetable\CatcodeTableIniTeX

```

The String and Other catcode tables.

```

154 \newluatexcatcodetable\CatcodeTableString
155 \setluatexcatcodetable\CatcodeTableString{%
156   \luatexcatcodetable\CatcodeTableIniTeX
157   \catcode0 12 % nul
158   \catcode13 12 % carriage return
159   \catcode37 12 % percent
160   \SetCatcodeRange{65}{90}{12}% A-Z
161   \SetCatcodeRange{97}{122}{12}% a-z
162   \catcode92 12 % backslash
163   \catcode127 12 }

```

```

164 \newluatexcatcodetable\CatcodeTableOther
165 \setluatexcatcodetable\CatcodeTableOther{%
166   \luatexcatcodetable\CatcodeTableString
167   \catcode32 12 }

```

Tables for L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>.

```

168 \newluatexcatcodetable\CatcodeTableLaTeX
169 \setluatexcatcodetable\CatcodeTableLaTeX{%
170   \luatexcatcodetable\CatcodeTableIniTeX
171   \SetCatcodeRange{0}{31}{15}%
172   \catcode9 10 % tab
173   \catcode10 12 % newline
174   \catcode12 13 % form feed
175   \catcode13 5 % carriage return
176   \catcode35 6 % hash
177   \catcode36 3 % dollar
178   \catcode38 4 % ampersand
179   \catcode94 7 % circumflex
180   \catcode95 8 % underscore
181   \catcode123 1 % brace left
182   \catcode125 2 % brace right
183   \catcode126 13 % tilde
184   \catcode127 15 }

185 \newluatexcatcodetable\CatcodeTableLaTeXAtLetter
186 \setluatexcatcodetable\CatcodeTableLaTeXAtLetter{%
187   \luatexcatcodetable\CatcodeTableLaTeX
188   \catcode64=11 }

```

A table for expl3.

```

189 \newluatexcatcodetable\CatcodeTableExpl
190 \setluatexcatcodetable\CatcodeTableExpl{%
191   \luatexcatcodetable\CatcodeTableLaTeX
192   \catcode126 10 % tilde is a space char
193   \catcode32 9 % space is ignored
194   \catcode9 9 % tab also ignored
195   \catcode95 11 % underscore letter
196   \catcode58 11 % colon letter
197 }

```

Now do the shortcuts.

```

198 \luatexbase@directlua{luatexbase.catcodetable_do_shortcuts()}

```

### 2.1.7 Catcode table number stack and regimes

The stack is implemented as a list of (brace-enclosed) T<sub>E</sub>X arguments. Initially, it contains only 0, the number of the active table when LuaT<sub>E</sub>X starts.

```

199 \def\lltxb@cctb@numstack{0}

```

Pushing a number on the stack actually means adding it to the beginning of the list (ie, unshifting it).

```

200 \def\PushCatcodeTableNumStack{%
201   \xdef\lltxb@cctb@numstack{{\the\luatexcatcodetable}\lltxb@cctb@numstack}}

```



Popping a number off the stack means shifting it.

```

202 \def\PopCatcodeTableNumStack{%
203   \expandafter\llt\b@cctb@numstack@pop\llt\b@cctb@numstack\@nil}
204 \def\llt\b@cctb@numstack@pop#1#2\@nil{%
205   \ifx\empty#2\empty
206     \begingroup
207     \ifdefined\PackageWarning
208       \let\x\PackageWarning
209     \else
210       \def\x####1####2{%
211         \begingroup \endlinechar10
212         \message{Package ####1 warning: ####2}%
213       \endgroup}%
214     \fi
215   \expandafter\endgroup
216   \x{luatexbase-cctb}{Popping from an empty CatcodeTableNumStack}%
217   \luatexcatcodetable\z@
218 \else
219   \gdef\llt\b@cctb@numstack{#2}%
220   \luatexcatcodetable#1\relax
221 \fi}

```

Now the catcode regime pseudo-environment as presented in the documentation.

```

222 \def\BeginCatcodeRegime#1{%
223   \PushCatcodeTableNumStack
224   \IncCatcodeTableStack
225   \setluatexcatcodetable\CatcodeTableStack{\luatexcatcodetable#1}%
226   \luatexcatcodetable\CatcodeTableStack}
227 \def\EndCatcodeRegime{%
228   \DecCatcodeTableStack
229   \PopCatcodeTableNumStack}

```

That's all, folks!

```

230 \luatexbase@cctb@sty@endinput%
231 \</texpackage>

```

## 2.2 Lua module

```

232 (*luamodule)
233 luatexbase          = luatexbase or { }
234 local luatexbase    = luatexbase
235 luatexbase.provides_module({
236   name      = "luatexbase-cctb",
237   version   = 0.6,
238   date      = "2013/05/11",
239   description = "Catcodetable allocation for LuaTeX",
240   author     = "Heiko Oberdiek, Elie Roux and Manuel Pegourie-Gonnard",
241   copyright  = "Heiko Oberdiek, Elie Roux and Manuel Pegourie-Gonnard",
242   license    = "CC0",
243 })

```

The number associated to a CS name is remembered in the `catcodetables` table.

```

244 luatexbase.catcodetables = luatexbase.catcodetables or { }

```

```

245 local catcodetables          = luatexbase.catcodetables
246 local function catcodetabledef_from_tex(name, number)
247     catcodetables[name] = tonumber(number)
248 end
249 luatexbase.catcodetabledef_from_tex = catcodetabledef_from_tex

    The next function creates some shortcuts for better readability in lua code. This makes
    luatexbase.catcodetables.latex equivalent to luatexbase.catcodetables.CatcodeTableLaTeX.
250 local function catcodetable_do_shortcuts()
251     local cat = catcodetables
252     cat['latex']          = cat.CatcodeTableLaTeX
253     cat['latex-package']  = cat.CatcodeTableLaTeXAtLetter
254     cat['latex-atletter'] = cat.CatcodeTableLaTeXAtLetter
255     cat['ini']            = cat.CatcodeTableIniTeX
256     cat['expl3']          = cat.CatcodeTableExpl
257     cat['expl']           = cat.CatcodeTableExpl
258     cat['string']         = cat.CatcodeTableString
259     cat['other']          = cat.CatcodeTableOther
260 end
261 luatexbase.catcodetable_do_shortcuts = catcodetable_do_shortcuts
262 </luamodule>

```

### 3 Test files

The tests done are very basic: we just make sure that the package loads correctly and the macros don't generate any error, under both LaTeX en Plain TeX.

```

263 <testplain,testlatex>\catcode00 15
264 <testplain>\input luatexbase-cctb.sty
265 <testlatex>\RequirePackage{luatexbase-cctb}
266 <*testplain,testlatex>
267 \begingroup \catcode64 11 \global\let\lua\luatexbase@directlua \endgroup

```

Also check that the catcodetable's number is remembered well, independently of the current value of `\escapechar`.

```

268 \newluatexcatcodetable\testcctb
269 \lua{assert(luatexbase.catcodetables.testcctb)}
270 \begingroup
271 \escapechar64
272 \newluatexcatcodetable\anothercctb
273 \endgroup
274 \lua{assert(luatexbase.catcodetables.anothercctb)}

```

Now, play a little bit with predefined tables. Be careful to change catcodes only inside a group to avoid chaos.

```

275 \def\test#1#2#3{%
276     \begingroup
277     \ifcsname CatcodeTable#1\endcsname \else \INTERR \fi
278     \luatexcatcodetable\csname CatcodeTable#1\endcsname
279     \ifnum\catcode#2=#3 \else \ERROR \fi
280     \endgroup}
281 \test {IniTeX}      {00} {09}
282 \test {IniTeX}      {92} {00}
283 \test {IniTeX}      {64} {12}

```

```

284 \test {IniTeX}          {65} {11}
285 \test {String}         {92} {12}
286 \test {String}         {65} {12}
287 \test {String}         {32} {10}
288 \test {Other}          {92} {12}
289 \test {Other}          {65} {12}
290 \test {Other}          {32} {12}
291 \test {LaTeXAtLetter} {64} {11}
292 \test {LaTeX}          {64} {12}
293 \test {Expl}           {32} {09}
294 \testlatex\documentclass{minimal}
295 \lua{%
296   tex.sprint('\string\setbox0=\string\hbox{')
297   tex.sprint(luatexbase.catcodetables.string, "\string\undef # _^&")
298   tex.sprint('}')
299 }
300 \testplain, testlatex
301 \testplain\bye
302 \testlatex\stop

```